

# 一个 VLSI 三层布线通孔最少化的启发式算法<sup>1</sup>

马 琪 严晓浪

(杭州电子工业学院 CAD 研究所 杭州 310037)

**摘 要** 该文在三层布线的线段-相交图模型基础上,提出了一个启发式算法来解决 VLSI 三层布线通孔最少化问题,该算法通过“总体优化”和“局部优化”两个阶段对三层布线进行通孔优化。算法考虑了实际约束的处理方法,并进行大量的布线实例验证。

**关键词** VLSI, 通孔最少化, 启发式, 三层布线

**中图分类号** TN305.97

## 1 引 言

通常 VLSI 三层布线采用的是 HVH 或 VHV 分层原则,这种横竖分层对于提高布通率很有效,但引入了大量通孔,通孔降低了芯片的可靠性和成品率,增加了制造成本,还增大了芯片面积,对电路性能也会产生一定影响。因此,减少芯片中通孔的数目便成了十分具有现实意义的问题。通孔最少化问题可分为两类——有约束通孔最少化 (Constrained Via Minimization, CVM) 问题和无约束通孔最少化 (Unconstrained Via Minimization, UVM) 问题。而有约束通孔最少化,即对于一个已经完成的详细布线,如何在不改变所有布线线段的平面位置的条件下,求得一个合理的线段分层,使所需的通孔数目最少。

文献 [1] 最先描述了三层 CVM 问题,并用平面图顶点三着色问题证明了 3CVM 的 NP 完全性,还提出了一种局部优化 (local sense) 的算法,这种算法逐一检查每个通孔,通过对与该通孔相连的线段布线层再分配,来消除该通孔。文献 [2-6] 对 3CVM 问题作了进一步的研究,其中文献 [2, 3, 6] 提出了总体优化 (global sense) 的 3CVM 算法,是将已布线段当无层处理,用一个图模型来表示这种无层布线,再将每一线段分配到更好的层位上,使通孔尽可能多地被消去,结果普遍比文献 [1] 要好。

我们提出的启发式 3CVM 算法,是以线段-相交图为图模型,将总体优化和局部优化两类算法相结合,分两步对三层详细布线进行通孔优化处理。算法考虑了对实际物理约束和不同布线风格的处理方法。实验证明,这是一个快速有效的算法。

## 2 3CVM 的图模型

### 2.1 线段-相交图 (Segment-Crossing Graph, SCG)

图 1(a) 所示的是 HVH 的三层布线的简单例子,图中小圆点表示通孔,这种风格的布线有大量通孔。图 1(b) 给出的是同个布线经通孔优化后的分层方法,与图 1(a) 相比,各线段的平面几何位置没有任何改变,但由于没有局限于 HVH(VHV) 分层原则,整个布线只需一个通孔。

为描述 3CVM 问题,我们先作如下定义:

**定义 1** 所有布线线段的平面位置已确定,但其分层未定的一个“未分层”或曰“不完整”的详细布线称为过渡布线 (Transient routing),如图 1(c) 即为图 1(a), 1(b) 的过渡布线。

<sup>1</sup> 1999-09-06 收到, 2000-02-13 定稿

**定义 2** 在过渡布线中，不与其它任何连线相交的一段最大连线称候选通孔 (Viacandidate)，它至少可以容纳一个通孔。图 1(c) 中小圆点  $V$  是最为常见的候选通孔，在原来的布线分层中它是事实上的通孔。图 2(a) 中的小圆点  $V$  也是一种候选通孔，在原来的布线分层中它不是通孔，但可以容纳一个通孔 (即可以成为通孔)。文献 [2, 3, 6] 没有将后者看作候选通孔，使这类候选通孔不能参加优化，所得结果的优化程度要受到影响。

**定义 3** 在过渡布线中，连接两个候选通孔 (或管脚端) 的连线称为线段 (Wire segment)，图 1(c) 中用  $S$  表示。

**定义 4** 在过渡布线中，两条属于不同线网的线段有一个及一个以上的相交点，则称这两条线段相交 (Crossing)，在实际布线中，相交的线段不能布在同一层上。一种称为 knock-knee 连接的布线模式，如图 2(b) 所示，线网  $a$  和线网  $b$  布在不同层并相交在所谓的 knock-knee 点，该点在图中用小方框标注。若我们把  $a$  和  $b$  看成就是弯曲线段，则线段  $a$  和  $b$  相交于 knock-knee 点。另一种布线风格即所谓的交叠布线，这种布线允许不同层上的走线小部分交叠 (overlap)，如图 2(c) 所示，交叠部分用小矩形标注，我们把  $c$  看成有多个弯曲的线段，则线段  $a$  和  $c$ ， $b$  和  $c$  相交于交叠部分。

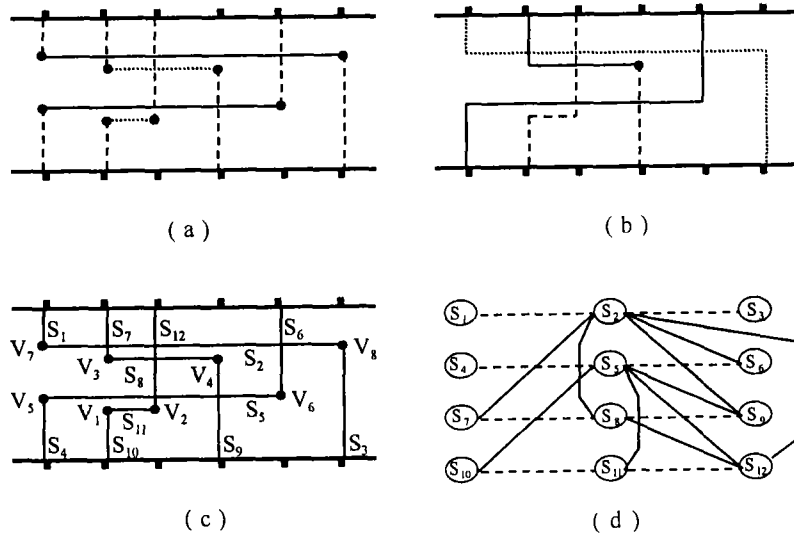


图 1 一个三层布线的实例

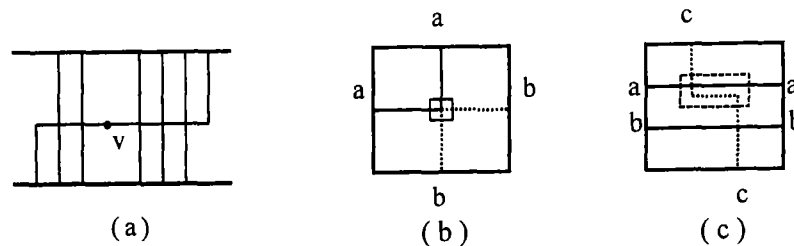


图 2 候选通孔和线段相交

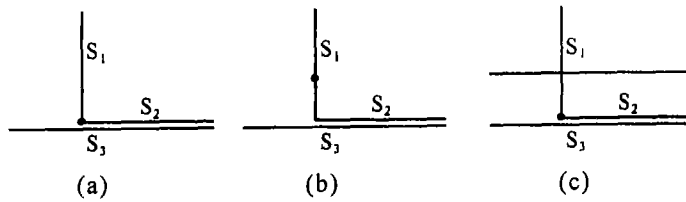


图3 三层布线中的一种约束情况

在文献 [2, 3] 中, 提出了 3CVM 的一种图模型 SCG。

**定义 5** 给定一个三层过渡布线, 定义图  $G_s = (V_s, E_v \cup E_c)$  为它的 SCG, 如图 1(d) 所示。其中顶点集  $V_s$  中每个顶点代表一条线段。

边集分成两个子集, 称通孔边集 (via-edge set)  $E_v$  与相交边集 (cross-edge set)  $E_c$ 。  $E_v$  中的边 (通孔边) 表示该边两个连接顶点所对应的线段由候选通孔相连接, 在图 1(d) 中用虚线表示;  $E_c$  中的边 (相交边) 表示该边两个连接顶点所对应的线段是相互相交的, 在图 1(d) 中用实线表示。

一个三层布线可用 SCG 图表示, 而我们可以将三层布线分层问题 3CVM 描述为 SCG 图特定的 3 着色问题。设  $L = \{1, 2, 3\}$  为颜色集, 其中一种颜色对应一个布线层, 三层布线的分层问题就相对于在该 3 层布线的线段-相交图上用三种颜色对每个顶点进行着色。当一通孔边连接的两个顶点着同一种颜色时, 这个候选通孔就不会成为通孔; 而一相交边连接的两个顶点不能着同一种颜色。因此, 线段的分层问题就等价于对线段-相交图的每个顶点着上合适颜色, 使得在满足  $E_c$  边连接的两个顶点着不同颜色的前提下, 使  $E_v$  边连接的两顶点尽可能着相同的颜色。

在 SCG 图中, 对顶点  $u \in V_s$ , 与它邻接的顶点以两种不同类型的边与它相连接, 为区分这两种顶点, 我们作如下定义:

顶点  $u$  的相容邻接顶点集, 记作  $A(u)$ , 指与顶点  $u$  以通孔边相连的所有顶点的集合。

顶点  $u$  的相斥邻接顶点集, 记作  $C(u)$ , 指与顶点  $u$  以相交边相连的所有顶点的集合。

## 2.2 物理约束的处理

由于物理约束, 某些管脚只能出现在三层金属中的某几 (一或二) 层, 与它相连的线段也被限制布在这几层; 另外, 某些线网必须布在某些特定的布线层, 该线网的所有线段也被限制布在这些布线层。在 SCG 图中, 与该线段相对应的顶点则必须着以与特定布线层相应的颜色, 我们把此类顶点  $u$  称为限制着色顶点 (restrictedly-colored vertex),  $u$  所应着的颜色的集合称为  $u$  的限制色 (restricted color) 集  $q(u)$ 。

图 3 所示的是三层布线中的另一种约束情况。在图 3(a) 的过渡布线中, 线段  $S_3$  穿过了连接线段  $S_1$  和  $S_2$  的候选通孔, 那么在实际布线中, 不能将线段  $S_1$  布在第一 (三) 层、线段  $S_2$  布在第三 (一) 层, 同时将线段  $S_3$  布在第二层, 即如果线段  $S_1$  和线段  $S_2$  要布在不同的布线层 (用通孔连接), 则两布线层必须相邻。如果在实际布线中, 我们限制线段  $S_3$  只能布在第一、三层, 就可以很好地解决这个问题。在对应的 SCG 图中, 线段  $S_3$  成为限制着色顶点, 其  $q(u) = \{1, 3\}$ 。在图 1 中,  $S_2$  和  $S_5$  就是这样的限制着色顶点。

图 3(a) 的这类限制着色顶点可以通过候选通孔逃逸来避免。见图 2(b), 如果线段  $S_1$  在与候选通孔相邻的部分 (图中是上部) 可以容纳通孔, 可以将候选通孔移至该部分, 线段  $S_1$  相应

缩短, 而线段  $S_2$  变弯曲. 图 3(c) 所示的是候选通孔不能逃逸的情况. 图 1 中的  $S_2$  和  $S_5$  都不能避免.

在 SCG 中, 对限制着色顶点  $u$ , 将个数等于  $|L - q(u)|$  的虚顶点以相交边与该限制着色顶点相连, 每个虚顶点分别着以颜色集合  $L - q(u)$  的一种限制色, 表示  $u$  不能着以其限制色集以外的颜色, 我们记  $u$  的虚顶点集为  $Q(u)$ . 虚顶点不代表任何线段, 所以不参加着色, 它只表示一种约束.

### 3 基于 SCG 的启发式 3CVM 算法

#### 3.1 与算法有关的基本概念

为叙述算法准确方便起见, 再引入几个基本定义:

赋予顶点  $u$  的颜色称为顶点  $u$  的已着色 (active color), 记作  $a(u)$ , 已赋颜色的顶点称作已着色点 (active vertex), 未赋色的顶点称为未着色点 (inactive vertex).  $A(u)$  中所有已着色点的已着色的集合叫做顶点  $u$  的推荐颜色集, 记作  $r(u)$ .  $C(u)$  中所有已着色点的已着色的集合  $b(u)$  叫做顶点  $u$  的约束颜色集,  $b(u)$  与  $L - q(u)$  (若  $u$  不是限制着色顶点, 则  $L - q(u) = \phi$ ) 的并集叫做顶点  $u$  的约束颜色全集, 记作  $c(u)$ , 显然, 不能将  $c(u)$  中的颜色赋予顶点  $u$ , 即  $a(u) \notin c(u)$ . 定义  $e(u) = r(u) - (r(u) \cap c(u))$ , 如果  $a(u) \in e(u)$ , 即  $a(u) \in r(u)$  且  $a(u) \notin c(u)$ , 则与  $u$  所对应的线段相连的某些候选通孔可以消除 (即候选通孔没有成为事实上的通孔).

对于顶点  $u$ , 若  $c(u)$  的模 (颜色数目)  $|c(u)| = p$ , 则称  $u$  为  $p$  约束点 ( $p$ -Constrained vertex), 简称  $pC$  点.

图 4 中,  $r(u) = \{1, 2, 3\}$ ,  $c(u) = \{2\}$ ,  $e(u) = \{1, 3\}$ , 顶点  $u$  是一个 1C 点.

#### 3.2 总体优化

基于上述定义, 我们给出如下总体优化算法:

ALGORITHM GLOBAL-MINIMIZATION

- (1) Construct an SCG  $G_s$  from the given detailed routing;
- (2) Let all vertices in  $G_s$  be inactive vertices;
- (3) REPEAT
- (4) Select an inactive vertex  $u$  with maximum priority  $P(u)$  as Eq.(1);
- (5) IF vertex  $u$  is a 3C vertex, THEN  
CALL Backtracking( $u$ );
- (6) IF vertex  $u$  is a 2C vertex; THEN  
Assign  $u$  the only color left;
- (7) IF vertex  $u$  is a 1C or 0C vertex; THEN  
Assign  $u$  a color  $a(u)$  based on Eqs.(2), (3) and (4);
- (8) Renew  $r(j)$  for all  $j \in A(u)$  and  $b(i)$  for all  $i \in C(u)$  and  $q(i)$   
 $r(j) = r(j) \cup a(u)$ ,  $b(i) = b(i) \cup a(u)$ ;  
Renew  $c(i) = b(i) \cup (L - q(i))$  for all  $i \in C(u)$ ;  
Renew  $P(u)$  for all vertices  $u$  in  $G_s$ ,
- (9) UNTIL all vertices in  $G_s$  are active vertices.

算法的结果非常依赖于着色的顺序, Step(4) 按顶点  $u$  的优先权函数  $P(u)$  从大到小进行着色, 其中

$$P(u) = |c(u)| \times \max_{v \in V_s} |C(v)| + |C(u)| \tag{1}$$

顶点优先权的高低依次为 3C, 2C, 1C 和 0C 点, 当顶点同为 pC 点时, 相斥邻接顶点个数  $|C(u)|$  多的顶点 (与较多线段相交的线段) 优先权高。

Step(7) 决定对 1C 或 0C 顶点  $u$  着以何种颜色以最大限度消除候选通孔。

若  $e(u) \neq \phi$ , 设  $|r_\tau(u)|$  为  $A(u)$  中颜色为  $\tau$  的顶点个数, 则使通孔减少数最多的颜色应为

$$a(u) = \tau, \quad \tau \in e(u) \text{ 并且 } |r_\tau(u)| \text{ 最大} \tag{2}$$

而  $|r_\tau(u)|$  也就是能减少的通孔数。在图 4 中,  $e(u) = \{1, 3\}$ ,  $|r_1(u)| = 1$ ,  $|r_3(u)| = 2$ , 所以  $a(u) = 3$ 。

有时存在多个颜色  $\tau$  的  $|r_\tau(u)|$  同为最大的情况, 如在图 5 中,  $e(u) = \{1, 2\}$ ,  $|r_1(u)| = |r_2(u)| = 1$ , 为处理这种特例, 设  $A'(u)$  为  $A(u)$  中所有未着色点的集合,  $d_\tau(A') =$  顶点  $j$  的数目,  $j \in A'$  且  $\tau \in c(j)$ , 则顶点  $u$  应着色为

$$a(u) = \tau, \quad \tau \text{ 满足 (2) 式并且 } d_\tau(A') \text{ 值最小} \tag{3}$$

如果再出现多个颜色  $\tau$  的  $d_\tau(A')$  同为最小的情况, 则从中任选一个颜色  $\tau$ 。

在图 5 中,  $A'(u) = \{x, y\}$ ,  $d_1(A') = 2$ ,  $d_2(A') = 1$ , 应选  $a(u) = 2$ , 候选通孔  $(x, u)$  可消除, 若  $a(u) = 1$ , 则候选通孔  $(x, u)$  和  $(y, u)$  都不能消除。

在  $e(u) = \phi$  情况下, 此时选

$$a(u) = \tau, \quad \tau \in (L - c(u)) \text{ 并且 } d_\tau(A') \text{ 值最小} \tag{4}$$

同理, 如果出现多个颜色  $\tau$  的  $d_\tau(A')$  同为最小的情况, 则从中任选一个颜色  $\tau$ 。

Step(5) 处理  $u$  为 3C 顶点的情况。此时  $L - c(u) = \phi$ , 需通过以下回溯过程 (Backtracking) 对某些已着色点重新着色来对  $u$  进行赋色。

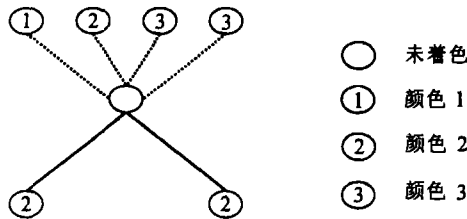


图 4 顶点的  $e(u)$  和  $r_\tau(u)$

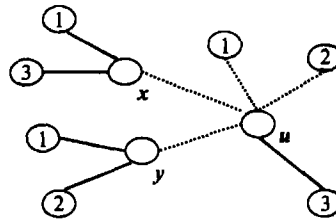


图 5 多个颜色的  $|r_\tau(u)|$  同为最大的情况

Backtracking( $u$ )

/\* vertex  $u$  is a 3C vertex \*/

(1) Select a color  $a(u)$  and assign it to  $u$  based on Eqs.(5) and (6) is minimum;

(2) FOR all  $i$  in  $C(u)$  DO

- (3) IF  $(a(i) = a(u))$  THEN /\* vertex  $i$  is set to be active \*/
- (4) BEGIN
- (5) Renew  $r(j)$  for all  $j \in A(i)$  and  $b(j)$  for all  $j \in C(i)$ ,  $r(j) = r(j) - a(i)$ ,  $b(j) = b(j) - a(i)$ ;  
Renew  $c(j) = b(j) \cup (L - q(j))$  for all  $j \in C(u)$ ;  
Renew  $P(V)$  for all  $V$  in  $G_s$ ;
- (6) Set vertex  $i$  to be inactive;
- (7) END
- (8) END

回溯过程必须选择  $b(u)$  中的一种颜色  $\tau$  赋于  $u$ ，所以  $C(u)$  中已着色为  $\tau$  的那个 (或那些) 已着色点必须转变成未着色点。 $C(u)$  中的已着色点依次为  $0C$ ， $1C$ ， $2C$  顶点，如果选择  $2C$  点所着的颜色赋给  $u$ ，则该  $2C$  点会变成  $3C$  点，在以后对这个未着色  $3C$  点进行处理时，又必须做一次回溯。而选择其它顶点就可避免重做回溯。

回溯过程用两个参数  $h_\tau(u)$  和  $f_\tau(u)$  决定选择  $b(u)$  中的哪种颜色赋给  $3C$  顶点  $u$ ，其中

$$h_\tau(u) = \max |c(i)|, \quad \text{对所有顶点 } i \in c(u) \text{ 并且 } \tau = a(u) \tag{5}$$

回溯过程首先选择  $h_\tau(u)$  为最小的颜色  $\tau$  赋于  $u$ ，当有多个颜色  $\tau$  的  $h_\tau(u)$  同为最小时，则看当这几种颜色赋于  $u$  后，哪一种颜色使最少的已着色点变为未着色点，即  $f_\tau(u)$  最小。

$$f_\tau(u) = \text{满足 } a(i) = \tau \text{ 且 } i \in C(u) \text{ 的顶点 } i \text{ 的数目} \tag{6}$$

图 6 是一个三层布线的例子， $h_1(u) = 2$ ， $h_2(u) = 1$ ， $h_3(u) = 1$ ，应选颜色 2 或 3，由于  $h_2(u) = h_3(u)$ ，而  $f_2(u) = 2$ ， $f_3(u) = 1$ ，所以最后  $a(u) = 3$ 。

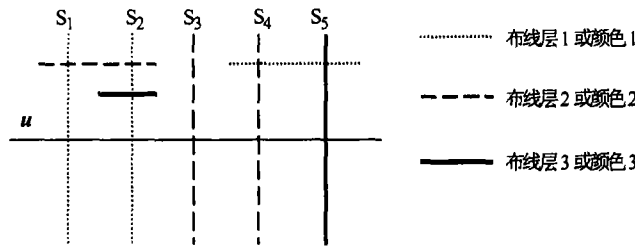


图 6 回溯过程中的着色选择

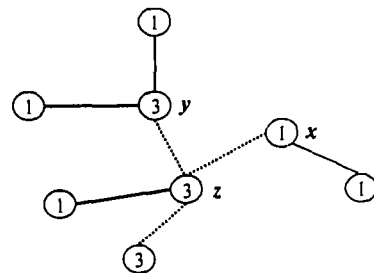


图 7 局部优化的逐点检查

### 3.3 局部优化

局部优化算法逐一检查每个顶点，通过对与该顶点相连的顶点重新着色来消除候选通孔。这个过程可用图 7 来表述。图 7 中，由于总体优化的着色顺序依次为  $x$ ， $y$ ， $z$ ，连接  $x$  和  $z$  的候选通孔没有被消除，如果将颜色 3 赋给顶点  $x$ ，则候选通孔  $(x, z)$  可消除。

从例子可以看出，要使与  $x$  相连的候选通孔能被消除，必须  $e(x) \neq \phi$  且  $a(x) \notin e(x)$ ，我们把这样的顶点  $x$  称为可变色点。局部优化实际是一个多次迭代改善过程，每一个迭代过程依次检查并处理所有可变色点，直到达到平衡态或不存在可变色点，迭代结束。所谓平衡态是指在连续几次迭代中可变色点的数目  $\delta$  不超过一个小的常数  $\Delta$ ，如  $\Delta = 4$ 。

## ALGORITHM LOCAL-MINIMIZATION

- (1) REPEAT
- (2)  $\delta = 0$
- (3) FOR each vertex  $i$  DO
- (4) IF  $e(i) \neq \phi$  and  $a(i) \notin e(i)$  THEN
- (5) BEGIN
- (6) Randomly select a color  $m$  from  $e(i)$ ;
- (7)  $a(i) = m$ ;
- (8)  $\delta = \delta + 1$ ;
- (9) Recalculate  $r(j)$  for all  $j \in A(i)$ ,  
Recalculate  $c(j)$  for all  $j \in C(i)$ ;
- (10) END
- (11) UNTIL ( $\delta = 0$  or  $\delta < \Delta$ )

## 4 实验结果

本算法用 C 语言编程, 在 Sun UltraSparc 10 工作站上实现。我们对算法进行了大量的实例验证。表 1 列出了实例运行结果。

表 1 三层布线的通孔最少化实验结果

实例	线网数	原通孔数	优化后的通孔数 (无管脚分层约束)	优化后的通孔数 (有管脚分层约束)*
Small Channel 本文图 2	4	8	1	6
Small Channel 文献 [5] 图 6	10	23	0	16
Example 1 文献 [5] 图 9	21	57	43	57
Example 3a 文献 [5] 图 10	30	68	50	68

\* 有管脚分层约束指所有管脚都被限布在第二层。

## 参 考 文 献

- [1] K. C. Chang, D. H-C. Du, Layer assignment problem for 3-layer routing, IEEE Trans. on Computer, 1988, C-37(5), 625-629.
- [2] K. E. Chang, H. F. Jyu, W. S. Feng, Constrained via minimization for three-layer routing, IEEE Trans. on Computer-Aided Design, 1989, CAD-2(6), 346-354.
- [3] S. C. Fang, K. E. Chang, W. S. Feng, S. J. Chen, Constrained via minimization with practical consideration for multi-layer VLSI/PCB routing problem, in Proc. 28th ACM/IEEE Design Automation Conf., 1991, 4, 5, 60-65.
- [4] K. Ann, S. Sahni, Constrained via minimization, IEEE Trans. on CAD of IC and Sys., 1993, 12(2), 273-282.
- [5] J. S. Cherng, S. J. Chen, C. C. Tsai, J. M. Ho, An efficient approach for via minimization in multi-layer VLSI/PCB routing, in Proc. IEEE Custom Integrated Circuits Conf., 1995, 473-476.
- [6] K. Takahashi, T. Watanabe, A heuristic algorithm to solve constrained via minimization for three-layer routing problem, in Proc. ISCAS, 1998, 486-490.

- [7] Y. K. Chen, M. L. Liu, Three-layer channel routing, IEEE Trans. on Computer-Aided Design, 1984, CAD-3(2), 156-163.

## A HEURISTIC ALGORITHM TO THREE-LAYER CONSTRAINED VIA MINIMIZATION

Ma Qi    Yan Xiaolang

(CAD Research Center, Hangzhou Institute of Electronic Engineering, Hangzhou 310037, China)

**Abstract** In this paper, a heuristic approach for constrained via minimization for VLSI three-layer routing on the basis of segment-crossing graph model is presented. Two types of algorithm—global sense and local sense are combined into the new algorithm. In addition, many physical constraints are taken into consideration. According to the results, this algorithm is verified to be very efficient and encouraging.

**Key words** VLSI, Constrained via minimization, Heuristic, Three-layer routing

马 琪: 男, 1968 年生, 博士, 主要从事集成电路物理设计、逻辑设计自动化方面的研究.

严晓浪: 男, 1947 年生, 教授, 博士生导师, 硕士, 杭州电子工业学院 CAD 研究所所长, 主要从事电子设计自动化方面的教学与科研.