

Keumog Ahn
University of Minnesota

and

Sartaj Sahni
University of Florida

Abstract

We show that the general two layer constrained via minimization problem and the three layer constrained via minimization problem for HVH topologies are NP-hard. A backtracking and a heuristic algorithm for the three layer HVH constrained via minimization problem are proposed. The backtracking algorithm can also be used for three layer non HVH problems. Experimental results indicate that our heuristic generally outperforms that of [CHAN89].

Keywords and Phrases

Constrained via minimization, NP-hard, HVH, 2 layer, 3 layer, heuristic

* This research was supported, in part, by the National Science Foundation under grants DCR-84-20935 and MIP-86-17374.

1. Introduction

The input to the *general constrained via minimization* (CVM) problem is a two dimensional routing region with terminals arranged in rows and columns inside the region together with a layout of the nets. This layout specifies wire paths together with layer assignment for each segment of each wire path. The initial layout is feasible in that it contains no crossing or overlap (Figure 1) of wire segments of different nets. The objective is to reassign the wire segments to layers so as to minimize the number of vias. The reassignment is required to preserve the original topology (i.e., wire paths but not layer assignment) and should be such that there is no crossing or overlap of wire segments of different nets on any layer. Figure 2(a) gives a possible input to the CVM problem. Vias are shown as small boxes. There are 4 vias in the layout of Figure 2(a). Broken lines indicate wires in layer one and solid lines indicate wires in the second layer. Figure 2(b) gives another layout that preserves the layout topology and uses 1 via. The instance of Figure 2 is a 2 layer instance as the input layout uses 2 layers.

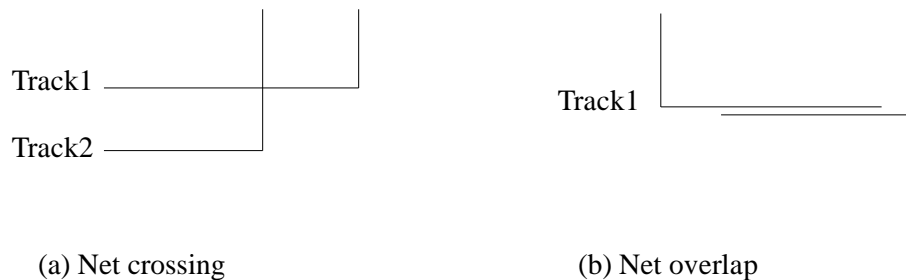


Figure 1 Crossing and overlap

In a grid based CVM problem the input layout is confined to follow horizontal and vertical grid lines that are superimposed on the routing region. Figure 3(a) shows a 2 layer grid based CVM input that uses 5 vias and Figure 3(b) shows a layout obtained from that of Figure 3(a) by reassigning layers but preserving the input topology. The layout of Figure 3(b) uses 1 via. Solid lines represent wires on one layer and broken ones represent wires on the other layer.

The HVH CVM problem is a 3 layer grid based CVM problem in which the input topologies are restricted such that layers 1 and 3 contain only horizontal wire segments and layer 2 contains only vertical wire segments. However, this restriction does not apply to the output. In reassigning layers, the topology has to be preserved but both horizontal and vertical segments can be assigned to any layer.

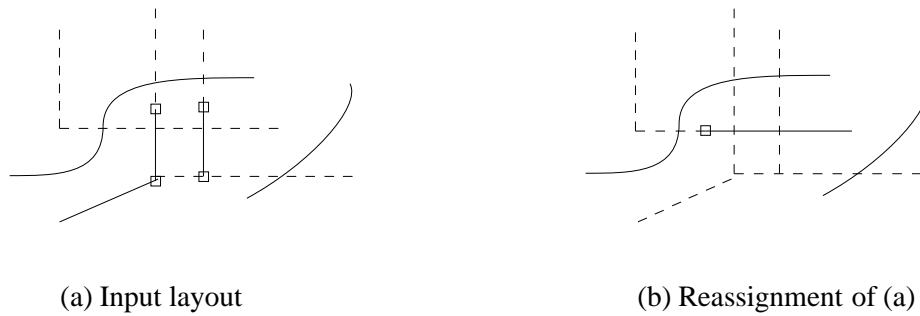


Figure 2 An instance of the CVM problem



Figure 3 An instance of a grid based 2CVM problem

Note that a via that goes from layer 1 to layer 3 passes layer 2 and makes electrical contact with any wire segment that uses this grid position on any of the three layers. An example of an HVH CVM instance is given in Figure 4(a) and a possible layer reassignment that preserves net topology is provided in Figure 4(b). Broken lines represent wires on layer 2 while wires on layers 1 and 3 are represented by solid lines. Layer 3 (1) wires are labeled 3 (1). The layout of Figure 4(a) uses 7 vias while that of Figure 4(b) uses no via.

Another variant of the CVM problem is the channel CVM (CCVM) problem. In this the net terminals are constrained to lie on two sides of a routing channel. Figure 5(a) gives an instance of the grid based 2CCVM problem and Figure 5(b) gives a layer reassignment that reduces the number of vias

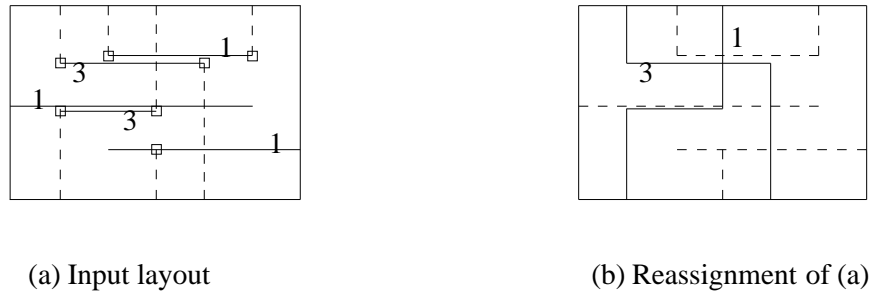


Figure 4 An instance of HVH CVM problem

from 6 to 2.



Figure 5 An instance of 2CCVM problem

The general 2 layer CVM (2CVM) problem was shown to be NP-hard by Naclerio, Masuda, and Nakajima in [NACL89]. Their proof also applies to the 2 layer CVM problem with any of the following restrictions:

- 1) The input layout is grid based
- 2) Vias are restricted to lie at wire junctions (i.e., points where two or more wire segments of the same net meet) that were in the original layout
- 3) The maximum wire junction degree is limited to any number ≥ 6 .

Heuristics for the grid based 2CVM problem have been proposed by several researchers (see [CHAN87] and [XION88] for example). The three layer CVM problem was first shown to be NP-hard by Chung and Du [CHAN88]. They did this by reducing the planar graph three colorability problem to the 3 layer CVM problem. Chang, Jyu, and Feng [CHAN89] provide an alternate proof using the three satisfiability problem. Actually the 2 layer proof of [NACL89] extends to the three layer case. For this, we need merely replace their two layer sublayout (Figure 6(a)) by the 3 layer sublayout of Figure 6(b). As a result, the 3CVM problem remains NP-hard under the added restrictions cited earlier.

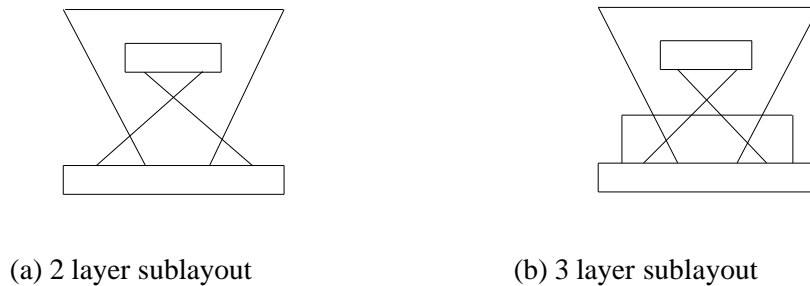


Figure 6 Constructs for the NP-hardness proof of [NACL89]

Both Chang and Du [CHAN88] and Chang, Jyu, and Feng [CHAN89] develop heuristics for the 3CVM problem. Experimental studies performed by [CHAN89] indicate that their heuristic is superior to that of [CHAN88]. Heuristics for the gridless 2CVM problem are proposed in [NACL87]. A related problem is the unconstrained via minimization (UVM) problem in which the wire layout is not constrained by a specified topology. Heuristics for 2UVM can be found in [HASH71], [HSU83], and [MARE84].

In this paper, we first show that the 2 layer channel CVM (2CCVM) problem is NP-hard (Section 2). Next, in Section 3, we show that the HVH CVM problem is NP-hard. In Section 4, we make some observations about the HVH CVM problem. These are used in Sections 5 and 6, respectively, to arrive at a heuristic and a backtracking algorithm for the HVH CVM problem. The backtracking algorithm may be applied, without modification, to three layer non HVH CVM instances also. Experimental results are provided in Section 7.

Throughout this paper, we assume that vias are restricted to junctions in the specified input layout. These points are called via candidates. The via candidates for the instance of Figure 1(a) are shown by

boxes in Figure 7. Observe that since a wire can change layers only by the use of a via, entire wire segments that contain no via candidate must be placed in the same layer.

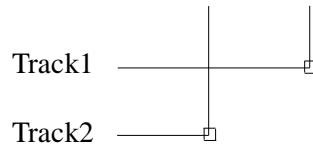


Figure 7 Via candidates

2. Complexity of 2CCVM

In this section we show that the two layer CVM problem with the added restriction that the pins lie on two sides of a channel (2CCVM) is NP-hard. Our proof shows how any instance of 2CVM may be transformed, in polynomial time, into an instance of the 2CCVM problem such that from an optimal solution to the 2CCVM instance one can obtain, in polynomial time, an optimal solution to the original 2CVM instance.

Consider any instance I of 2CVM. The top and bottom boundaries of this may be viewed as the two sides of a channel (Figure 1(a)). To transform this instance into an instance of 2CCVM, we need to eliminate all pin positions that are not on these two sides of the channel. Each such pin position, p , falls into one of the categories: (PA) there is a path from p to one of the two sides of the channel and this path does not intersect any of the nets of I ; and (PB) there is no such path. In the example of Figure 8(a), all nets are two pin nets and all pin positions other than A fall into category PA. The layer assignments are not shown in this figure. Pin positions in category PA can be moved to one of the two sides of the channel by extending the net from this position to a position on the channel side without intersecting any existing net. The extensions are shown by broken lines in Figure 8(b). Clearly, such an extension which, in effect, moves pin positions to the sides of the channel does not affect the number of vias in the optimal solution. Let I' be the instance of 2CCVM obtained by extending nets in this way so that all remaining pin positions that are not on one of the two channel sides are in category PB.

To extend the type PB pins to the sides of the channel, we need a more complex construction. Let p be such a remaining pin position. Every path from p to one of the two channel sides crosses at least

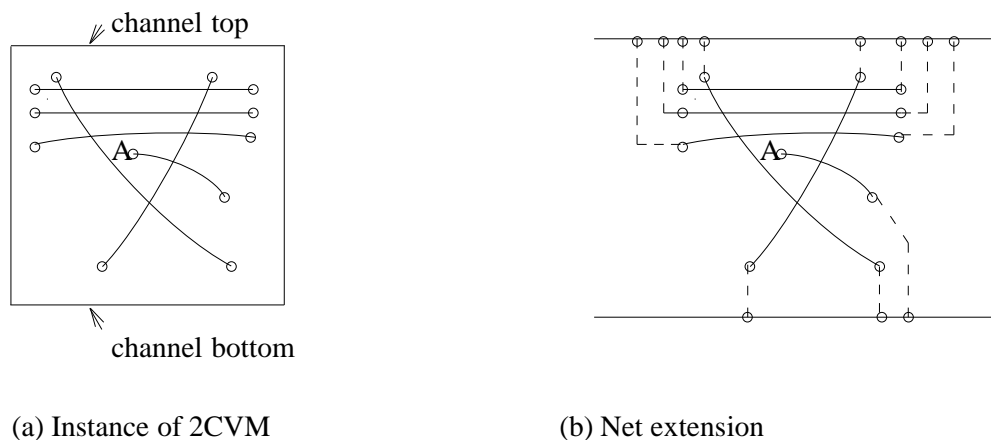


Figure 8 An example of pins and their extensions

one net. Suppose that r net segments are crossed. Figure 9(a) gives an example with $r = 3$. At each point where this path crosses a net, we shall use a crossover box. Two types, S and T, of crossover boxes are used. Type S is used at the first crossover and type T at all remaining ones. A type S crossover box has one input terminal and 8 output terminals. Type T boxes have 8 input and 8 output terminals. The pin position p is connected to the input of S, the outputs of an S box and a T box are connected to the inputs of the next T box (if any) on the path. If there is no such next box then the eight outputs are extended to the channel side as in Figure 9(b). So, one pin position p results in 8 pin positions on the channel side. The connection of the outputs of one box to the inputs of the next is done, in order, left to right. I.e., the i 'th output of a box is connected to the i 'th input of the next box, $1 \leq i \leq 8$.

The configurations for the S and T boxes are shown in Figures 10 and 11, respectively. The broken lines labeled A and B denote the net being crossed by the path from the pin p to one of the channel sides. By appropriately scaling the construction, S and T crossover boxes may be placed so as not to intersect any of the existing nets other than the ones crossed by the aforementioned path from p . Note that the eight output terminals of an S box and the eight output and eight input terminals of a T box represent exactly four distinct nets.

Let I'' be the 2CCVM instance obtained from I' by extending all pin positions in category PB using S and T crossover boxes as described above. It should be clear that I'' can be obtained from the original 2CVM instance I in polynomial time and that the size of I'' is a polynomial function of the size of I . We need to show that from an optimal solution to I'' we can obtain, in polynomial time, an optimal



Figure 9 Extending a pin with crossovers to the top side of the channel

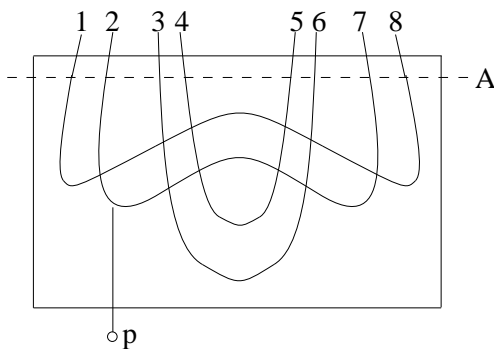


Figure 10 S crossover box

solution to I. For this, we shall first establish several properties of the crossover boxes S and T.

We shall use the term *edge* to denote a net segment that has no crossovers in its interior. Consider the net configuration of Figure 12(c). This shows five net segments labeled 1-5. Segments that cross are from different nets. The subsegments (a,b), (b,c), (c,d), (d,e), and (e,a) are five of the edges in this layout. These five edges form a cycle in the graph theoretic sense if we regard the net crossover points a, b, c, d and e as vertices of a graph. This cycle is an *odd* cycle as there are an odd number of edges on it.

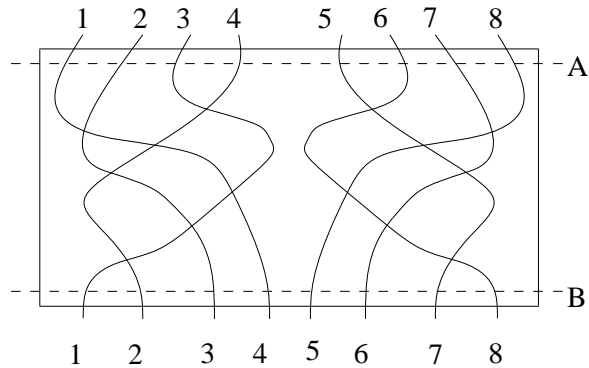


Figure 11 T crossover box

Figure 12(a) shows an odd cycle with three edges and Figure 12(b) shows an even cycle with four edges.

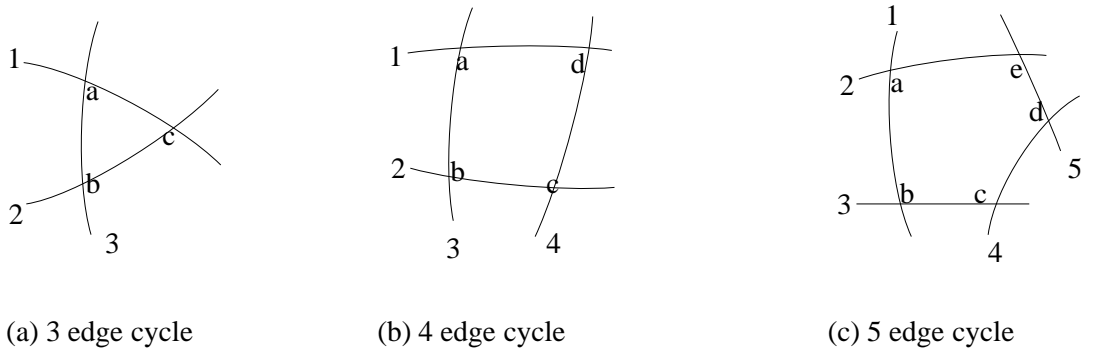


Figure 12 Cycles in a layout

Lemma 1: (a) In every two layer layout of an odd cycle there must be an odd number of vias on the cycle.

(b) In every two layer layout of an even cycle there must be an even number of vias on the cycle (we consider 0 vias to be an even number of vias).

Proof: Since adjacent edges of a cycle come from different nets, there must be a layer change at each vertex of the cycle. The layer change at a vertex uses no via and results from assigning the two edge segments that meet at a vertex to different layers. Further, if we start at any point on a cycle (not necessarily a vertex) and walk around the cycle and return to this point, the number of layer changes mod 2 must be zero as we must return to the same layer. Let e be the number of edges (and hence vertices on the cycle) and let q be the number of vias. Our discussion implies that $(e+q) \bmod 2 = 0$. If e is odd, then q must be odd. If e is even, then q must be even. \square

Figure 13 gives some possible layer and via assignments for the cycles of Figure 12. Note that we are limited to two layers. Layer numbers are noted on the edges or edge segments. Note also that the two layer assignments of Figure 13 are done using a minimum number of vias. Figure 14 depicts some of the net segments in an S crossover box that is connected to a T crossover box. All the segments of S and those segments of T below the horizontal line through points a and b of Figure 11 are included as solid lines. The five distinct nets are labeled n1 through n5. The point x is not a crossover point as the three net segments that meet at x belong to the same net n1. (a,b), (b,c), (i,k), (m,o), and (g,r) are some of the edges in this configuration.

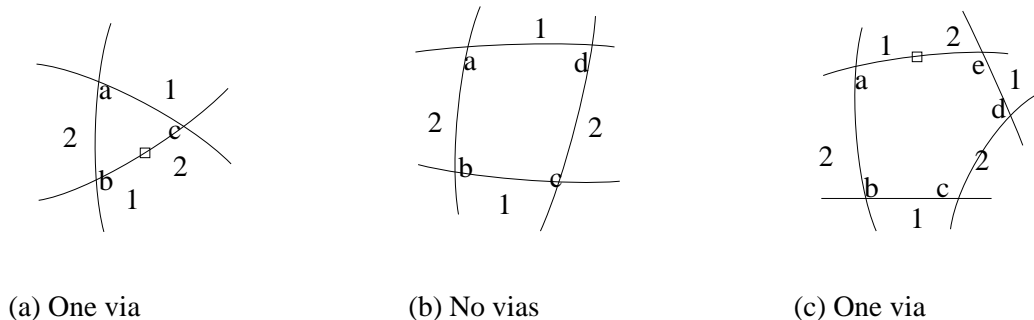


Figure 13 Possible layer and via assignment for cycles of length 3, 4, and 5

Lemma 2: If the two ends of n_5 (cf. Figure 14) are required to be on the same layer then, independent of which layer (x,y) is required to be on, a minimum of four vias are needed to complete the layout in two layers. Every four via layout places exactly two vias on n_5 .

Proof: Figure 15 shows a four via two layer assignment under the assumption that n_5 is assigned to layer l_2 . In this figure l_1 and l_2 denote the two layers and a box denotes a via location. Note that if the

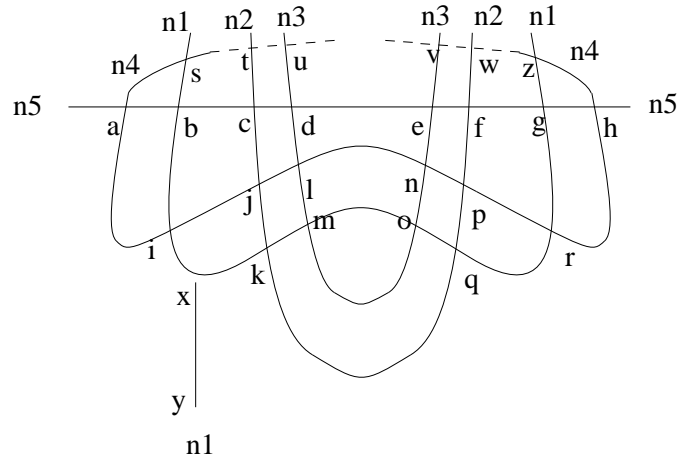


Figure 14 Configuration for Lemma 2

segment (x,y) of n1 is required to be on *l1*, then this can be accommodated by moving the via on the edge (i,k) so that it is on the segment (i,x) of this edge.

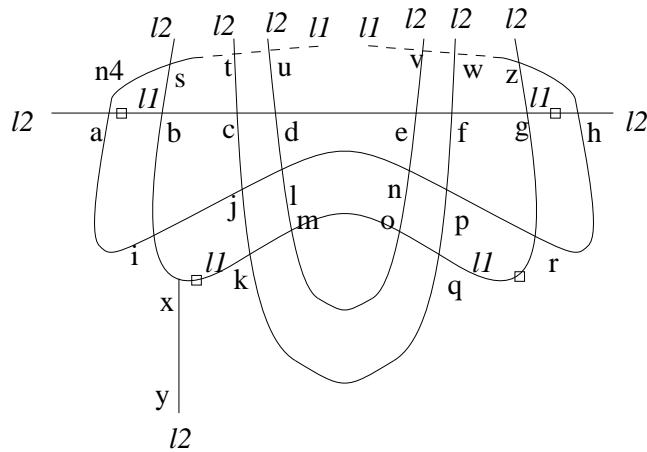


Figure 15 4 via, 2 layer assignment of Figure 7

Since the cycles abs , abi , ikj , pqr , grh , and ghz are of odd length, every two layer assignment must have at least one via on each. The cycles abs and abi can share a via if it is placed on the common edge (a,b) . Similarly, ghz and grh can share a via if it is on edge (g,h) . Hence the via requirement can be reduced to four if one via is placed on each of the edges (a,b) and (g,h) . No other via sharing is possible. Hence, in every four via layout, only two vias can be placed on $n5$. \square

Lemma 3: If the two end points of $n5$ (cf. Figure 14) are required to be on different layers then, independent of which layer (x,y) is required to be on, a minimum of five vias are needed to complete the layout in two layers.

Proof: A possible five via two layer assignment in which the ends of $n5$ are on different layers is given in Figure 16. This assumes that (x,y) is to be on layer $l2$. If it is to be on $l1$, then the via on (i,k) may be moved to the edge (i, x) . From the proof of Lemma 2, we know that at least four vias are needed to handle the six odd cycles and that every four via assignment has two vias on $n5$. However, for the ends of $n5$ to be on different layers, an odd number of vias need to be assigned to $n5$. So, no four via assignment is possible. A five via layout can have a via on each of (a,b) and (i,k) , one on each of the cycles ghz , grh , and pqr , and no via on (g,h) . \square

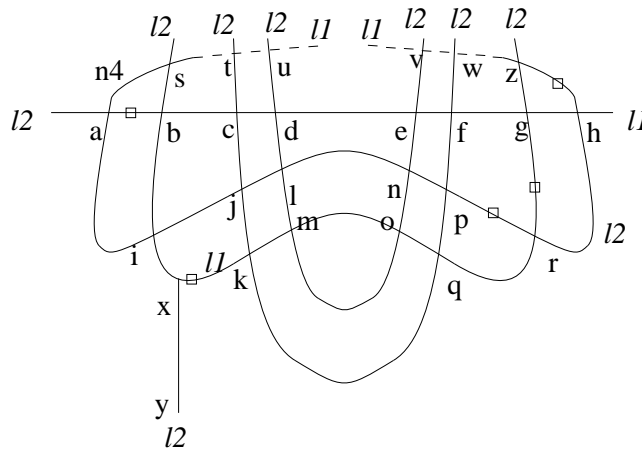


Figure 16 5 via, 2 layer assignment of Figure 7

Figure 17 gives the net segments that result when two T boxes are interconnected. The nets $n1$ - $n4$ are numbered in accordance with Figure 14 and the vertices s , t , u , v , w , and z correspond to the vertices

in Figure 14 that have the same label.

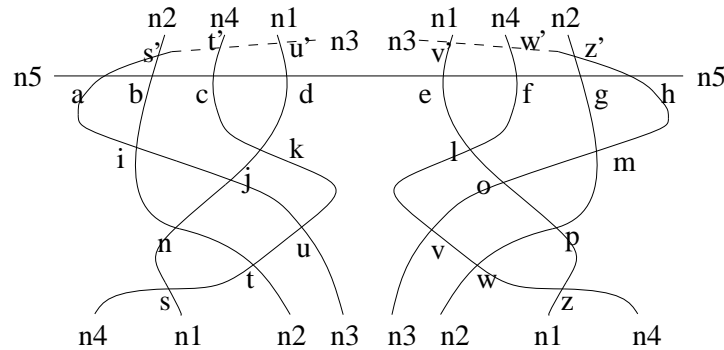


Figure 17 Nets from two interconnected T boxes

Lemma 4: If the two ends of $n5$ (cf. Figure 17) are required to be on the same layer, then a minimum of ten vias are required to complete the layout in two layers. Every ten via layout has exactly two vias on $n5$.

Proof: A ten via two layer layout that has both ends of $n5$ on the same layer is given in Figure 18. The configuration of Figure 17 contains 14 cycles of odd length (abs' , abi , inj , nst , $bijkc$, ckd , kju , elf , $flomg$, gmh , ghz' , opm , lvo , pwz). From Lemma 1, there must be a via on each. Cycles abs' and abi can share a via if the via is placed on (a,b) and ghz' and ghm can share a via if it is placed on (g,h) . The only other possibilities for via sharing are cycle $bckji$ shares a via with one or more of the cycles cdk , jku , and inj , and cycle $fgmol$ shares a via with one or more of the cycles lov , mop , and efl . This sharing by $bckji$ and $fgmol$ can reduce the via requirements by at most two. Hence at least ten vias are needed. Since the above defines every 10 via layout, every ten via layout has exactly two vias on $n5$. \square

Lemma 5: If the two end points of $n5$ (cf. Figure 17) are required to be on different layers, then a minimum of 11 vias are needed to complete the layout in two layers.

Proof: An eleven via two layer layout of Figure 17 is given in Figure 19. From the proof of Lemma 4, we know that at least ten vias are needed to layout Figure 10 on two layers and that every ten via layout has two vias on $n5$. Hence, in every 10 via layout, the two ends of $n5$ are on the same layer. So, at least 11 vias are needed if the two ends of $n5$ are on different layers. \square

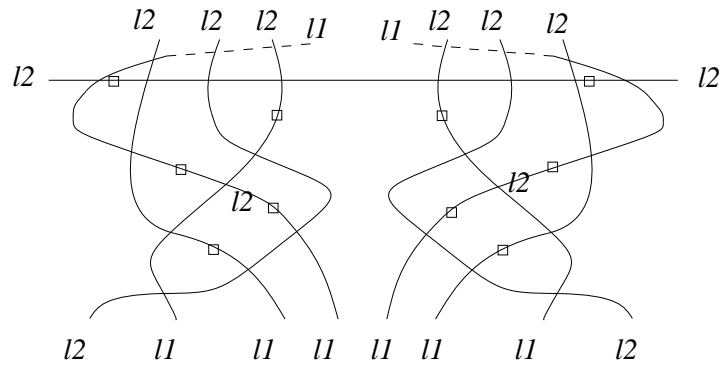


Figure 18 10 via, 2 layer assignment of Figure 17

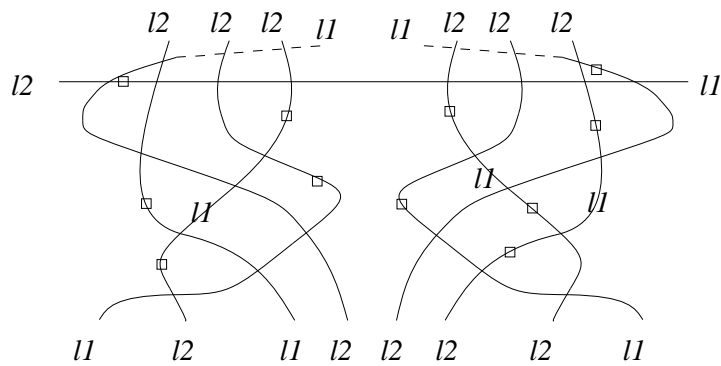


Figure 19 11 via, 2 layer assignment of Figure 17

Lemma 6: Let p be a type PB pin position that is extended to a side of the channel by using one type S and t type T crossover boxes. Suppose that q of the $t+1$ type n5 nets crossed by these $t+1$ crossover boxes are required to have their ends on different layers. The minimum number of vias needed to layout the $t+1$ crossover boxes is $4+10t+q$.

Proof: That $4+10t+q$ is a lower bound follows from Lemmas 2 through 5. To show that the layout can actually be done using this number of vias, we need to show that a type T crossover box can be layed

out using 10 vias when the ends of net A (Figure 11) are required to be on the same layer and using 11 vias when they are to be on different layers regardless of the requirements on the ends of net B (Figure 11). Figure 20 shows eight of the sixteen possibilities for the ends of A and B. The remaining eight are isomorphic to these. In Figure 20, vias are shown as o's and x's. In keeping with our development of Lemmas 2 through 5, vias placed on net B, which are represented as "o" instead of "x" in Figure 20, are charged to the crossover box below the current one. One may verify that the number of vias charged to a type T box is either 10 or 11 depending on whether the ends of net A are on the same layer or on different layers. \square

Theorem 1: The 2CCVM problem is NP-hard.

Proof: Let I be an instance of 2CVM that has a pin positions of type PA and b of type PB. Let I' be obtained from I by first extending the type PA positions to the channel sides as described earlier and then extending the type PB positions using S and T crossover boxes. Suppose that the path used to extend the i 'th type PB position crosses n_i nets, $1 \leq i \leq b$. We shall show that the optimal solution to I uses m vias iff the optimal solution to I' uses $m + 10 \sum_{i=1}^{i=b} n_i - 6b$ vias. Furthermore, from an optimal solution to I one can obtain, in polynomial time, an optimal solution to I' and vice versa.

Consider an optimal solution to I. Let m be the number of vias used. Place these m vias at exactly the same positions in I'. Also, preassign layers to all net segments of I' that are also in I. This layer assignment is the same as that used in the optimal solution for I. The layer assignment for the extensions of type PA pin positions is the same as that of the original net segments of I to which they are connected. For the extensions of the type PB pin positions, we place vias and assign layers in accordance with Figure 18. For the i 'th type PB position exactly $10n_i - 6$ additional vias are introduced. This is so because if the ends of a type A net segment (Figure 11) were preassigned to different layers, then there must also be an odd number of vias preassigned to points between these ends. Since no nets of the original instance cross between these two ends of the net segment of type A, there is exactly one such via (if there is more than one such via then the solution to I is not optimal as vias in excess of one can be eliminated without affecting the feasibility of the solution). This via can be used as one of the needed eleven vias and may be moved to any point between the ends of the type A net segment. Hence only ten additional vias are needed when the ends of this net segment are on different layers. If the ends of the type A net segment are preassigned to the same layer, there can be no vias preassigned to it as the preassignment is done from an optimal solution to I. Hence 10 additional vias are introduced by the Figure 18 assignment. Since one type S and $n_i - 1$ type T crossover boxes are used, the extension of the i 'th type PB position uses $4 + 10(n_i - 1) = 10n_i - 6$ additional vias. So, the number of vias used in the layout of

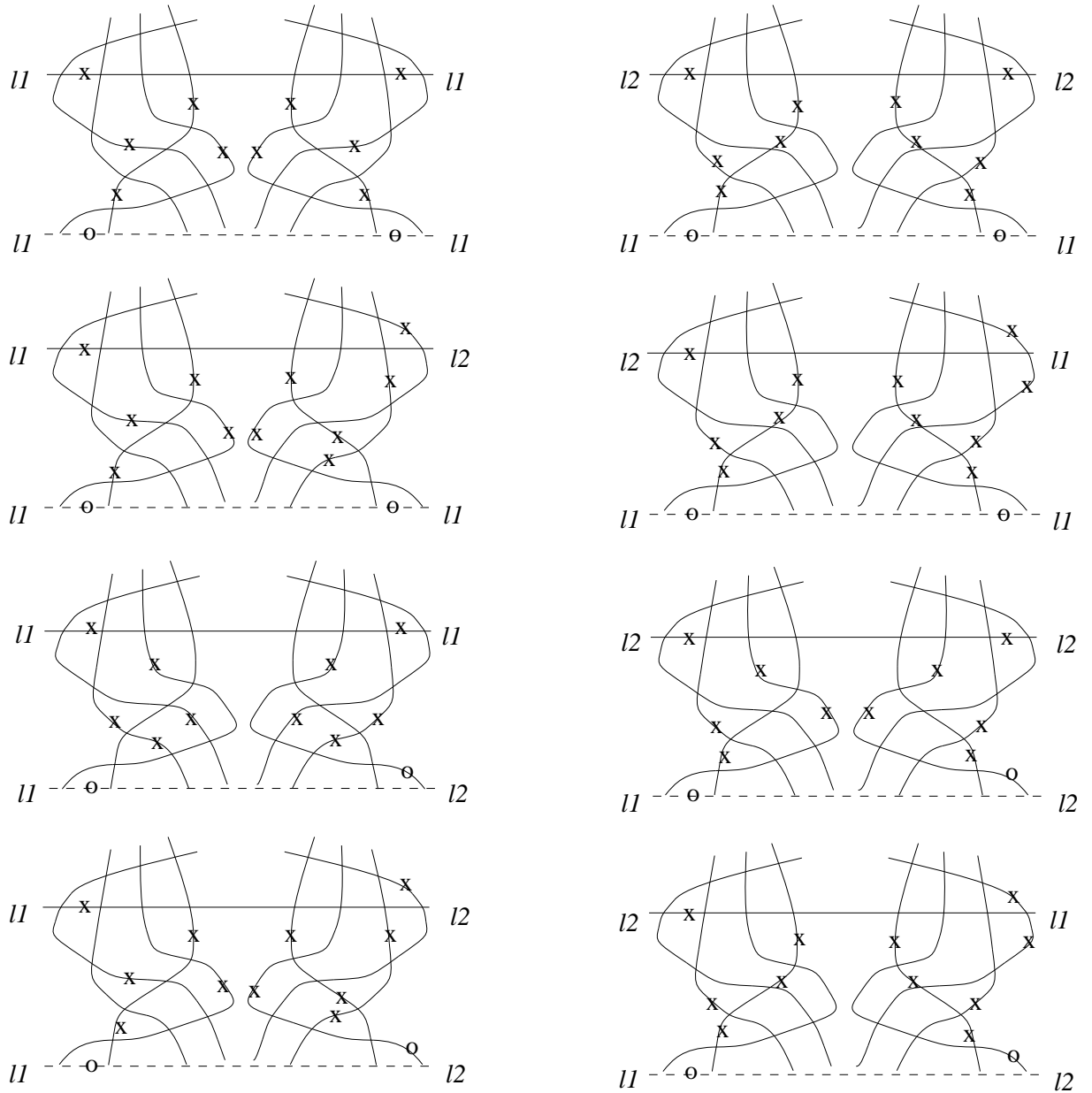


Figure 20 Cases for Lemma 6

$$I'' \text{ is } m + 10 \sum_{i=1}^{i=b} n_i - 6b.$$

Next consider an optimal solution to I'' . From this we can obtain a solution to I by eliminating the

pin position extensions and crossover boxes used in going from I to I" and also eliminating the vias on these. When eliminating vias from type A net segments, one is retained in case the number of vias on the segment is odd. From Lemmas 2 through 5, it follows that exactly $10 \sum_{i=1}^{i=b} n_i - 6b$ vias are eliminated in this way. Furthermore, following this we have a feasible solution to I that uses $10 \sum_{i=1}^{i=b} n_i - 6b$ fewer vias than used by the optimal solution to I". Hence the optimal solution to I uses m vias iff that for I" uses $m + 10 \sum_{i=1}^{i=b} n_i - 6b$ vias. Since I" can be constructed from I in polynomial time and since 2CVM is NP-hard [NACL89], it follows that 2CCVM is also NP-hard. \square

3. Complexity Of HVH CVM

We show that HVH CVM is NP-hard by showing that a polynomial time algorithm for HVH CVM implies such an algorithm to determine whether or not a planar graph is 3 colorable. This latter problem is known to be NP-hard [STOC83].

Theorem 2: The HVH CVM problem is NP-hard.

Proof: Let G be any planar graph. Consider any planar drawing of G . In this drawing, consider a vertex v and all edges incident to do. Draw a narrow closed curve [EHRL76] that encloses more than half of each edge incident to v (see Figure 21(b)). The closed curve is drawn so as not to touch any of the remaining edges in G . These closed curves are drawn for every vertex in G . Figure 21(c) shows the result for the planar graph drawing of Figure 21(a). The curves may now be rectilinearized (i.e., transformed so as to consist of horizontal and vertical segments only) and each closed rectilinear curve opened at some point so as not to affect the inter curve intersections (Figure 21(d)) and such that no two curves have overlapping horizontal or vertical segments (or portions thereof). If, now, we discard the original graph, what remains is an instance of the HVH CVM problem in which each the open curves specifies the topology of the layout for a two terminal net. The ends of the curve are the two terminals of the corresponding net. The specified construction is easily seen to be of polynomial complexity.

Observe that two open rectilinear curves (and, in fact, even the original continuous curves) of the construction cross iff the two vertices of G they are associated with are adjacent in G . Hence, the complete layout of two nets can be assigned to the same layer iff the two corresponding vertices of G can be assigned the same color. In other words, the constructed HVH CVM instance can be realized in 3 layers and with no vias iff G is three colorable. So, HVH CVM is NP-hard. \square

From the construction of Theorem 2, it follows that the following restricted versions of the HVH

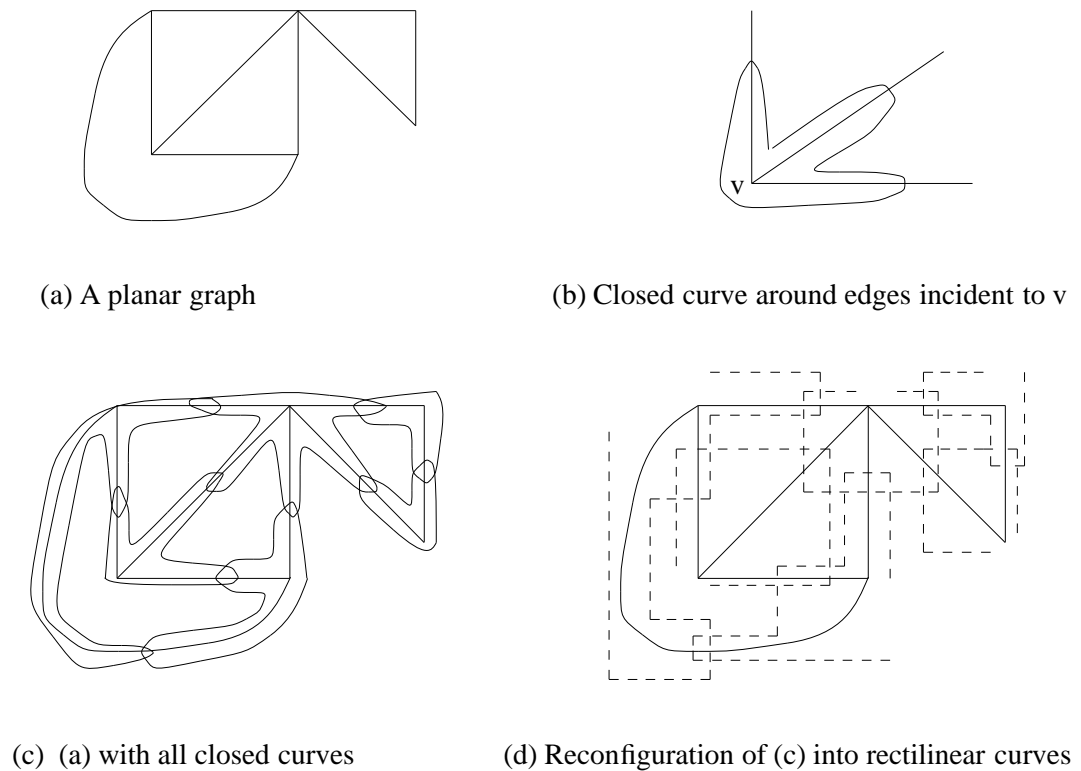


Figure 21 Construction of Theorem 2

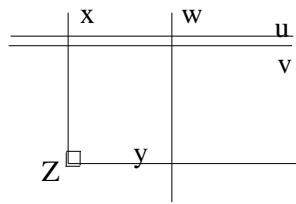
CVM problem are also NP-hard:

- 1) All nets in the HVH CVM instances are two terminal nets.
- 2) The HVH CVM instances have no horizontal or vertical segment overlaps.
- 3) The maximum junction degree is 2.
- 4) Vias are restricted to lie at junctions.

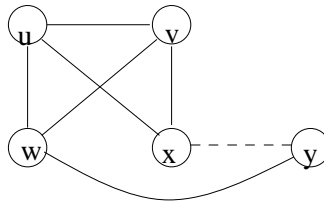
4. Observations

The via candidates of the input layout may be classified into the categories: unremovable, absolutely removable, and possibly removable. An *unremovable* via is one which must be present in every feasible via assignment. An *absolutely removable* via is one which is not present in any optimal via assignment. All other via candidates are *possibly removable*.

Consider the sublayout of Figure 22(a). The five wire segments u - y come from four different nets and the sublayout contains just one via candidate Z . Since w crosses u and v , w must be assigned to a different layer from u and v . Further, since u and v overlap, they must be assigned to different layers. Hence, u, v , and w must be assigned to different layers. If the via candidate Z is eliminated, then both x and y must be assigned to the same layer to maintain electrical connectivity. However, this cannot be done in three layers without crossing u, v , or w . So, the via candidate Z is unremovable as far as three layer layouts are concerned.



(a) A sublayout



(b) Its crossing graph

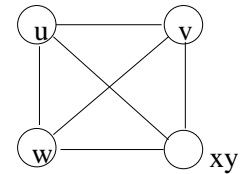
(c) Coalescing of x and y

Figure 22 Unremovable via

Unremovable vias can be identified by drawing the crossing graph of the layout. This graph was defined in [CHAN89] to be an undirected graph in which each vertex represents a distinct wire segment of the layout; there is a solid edge between two vertices iff the corresponding wire segments cross or overlap; and there is a broken edge between two vertices iff two corresponding wire segments meet at a via candidate. The crossing graph for the sublayout of Figure 22(a) is shown in Figure 22(b).

The removal of a via candidate modifies the crossing graph as such a removal results in the wire segments that meet at the via candidate being combined into a single wire segment. So, if the via candidate Z of Figure 22(a) is removed, the vertices x and y of the crossing graph coalesce into a single vertex. When two vertices x and y are coalesced, they are replaced by a single vertex xy , all edges previously incident on x or y are now incident on xy ; and the broken edge between x and y is eliminated. The resulting crossing graph is shown in Figure 22(c). This is the complete graph, K_4 , on four vertices.

Lemma 7: Let G be the crossing graph of an HVH CVM instance. Assume that G does not contain the subgraph K_4 comprised solely of solid edges. Let the broken edge $(x,y) \in G$ represent the via candidate Z . Z is an unremovable via candidate if the coalescing of vertices x and y results in a subgraph K_4

comprised solely of solid edges.

Proof: The creation of K_4 of solid edges requires that the wire segments corresponding to the four vertices of K_4 be assigned to different layers as these four wire segments cross each other. Hence there is no feasible three layer assignment for these four wire segments. Hence Z is unremovable. \square

Figures 23(a) and (b) provide two examples of absolutely removable vias. We assume that both subtopologies are part of a feasible instance of HVH CVM. We claim that the via candidate Z in each example is absolutely removable. Consider Figure 23(a). No matter which layer x is assigned to, y can be assigned to the same layer as y conflicts (i.e., overlaps or crosses) only with w and w cannot be assigned to the same layer as x since x and w also conflict. In the case of Figure 23(b), every feasible layer assignment assigns x and w to different layers. The segments v and y can be reassigned to the same layer as x as they conflict only with w which cannot be in the same layer as x because of the overlap between x and w . As a result of the reassignment of x and y , the via candidate Z is eliminated.



Figure 23 Absolutely removable vias Z

Lemma 8: Let Z be a via candidate and let x be one of the wire segments that meet at Z . Suppose that every other wire segment, y , that meets at Z overlaps or crosses only wire segments w that also overlap or cross x . Then, Z is absolutely removable.

Proof: Consider any three layer via and layer assignment for the given input topology. The wire segments that meet at Z can always be reassigned to the layer to which x is assigned as the segments with which they conflict also conflict with x and so cannot be on the same layer as x . Hence the via candidate Z may be eliminated. \square

Via candidates that do not get classified as unremovable or absolutely removable by Lemmas 7 and 8 are classified as possibly removable.

Let I be an instance of HVH CVM and let x be a wire segment of I . Let I' be an instance that results from the removal of x from I . If x can be added to every feasible layer and via assignment of I' without increasing the number of vias, then x is a *trivial wire segment*.

Lemma 9: (a) Let x be a wire segment of an instance of HVH CVM. If x is connected to no other wire segment and if all the wire segments with which x conflicts (i.e., overlaps or crosses) can be assigned to at most two layers in any feasible assignment, then x is a trivial wire segment.

(b) If x is connected to other wire segments at only one of its two ends and if every wire segment with which x has a conflict also conflicts with one of the wire segments, y , to which x is connected, then x is a trivial wire segment.

Proof: In case (a), x can be assigned to the layer that does not contain any of the wire segments with which it conflicts and in case (b), x can be assigned to the same layer as y . Figures 24(a) and (b), respectively, provide an example for cases (a) and (b). \square

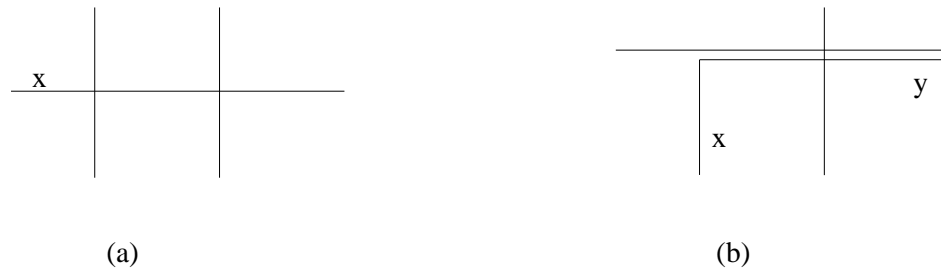


Figure 24 Examples for Lemma 9

Lemma 10: Consider the subtopology of Figure 25(a). In every three layer assignment wire segments (a, b, c, and d) of this subtopology must be placed in layers 1 and 3 and the vertical segments (e and f) in layer 2.

Proof: Suppose that one of a and b is assigned to layer 2. Without loss of generality, we may assume that the other is assigned to layer 1. So, e and f must be assigned to layer 3. Hence, neither c nor d can be assigned to layer 3. Since c and d overlap, they must be assigned to different layers. Without loss of generality, we may assume that c is assigned to layer 1 and d to layer 2. The via Z passes through layer 2 and so makes contact with wire segment d resulting in infeasibility. Hence, neither a nor b can be assigned to layer 2. Similarly, we may show that neither c nor d may be assigned to layer 2. Since a and

b overlap, one is in layer 1 and the other is layer 3. Hence neither e nor f can be assigned to layers 1 and 3 and so must be assigned to layer 2. \square

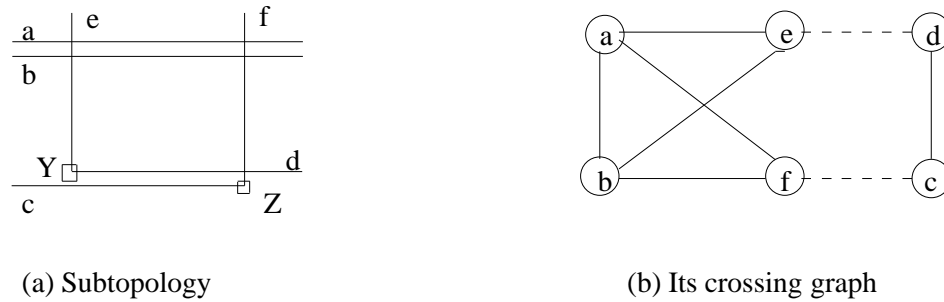


Figure 25 Subtopology for Lemma 10

By identifying the subgraph of Figure 25(b) in the crossing graph of the input layout, we can restrict certain wire segments to be in layer 2 and others to be in one of the layers 1 and 3.

4. Heuristic For HVH CVM

As indicated in Section 1, Chang and Du [CHAN88] and Chang, Jyu, and Feng [CHAN89] have proposed heuristics for the 3CVM problem. While their development is not confined to HVH inputs, their experimentation was limited to HVH CVM instances. The heuristic of Chang and Du [CHAN88] attempts to reduce the number of vias by considering the vias in the initial layout one at a time. When a via is considered, an attempt is made to eliminate it by modifying the layer assignment of some wire segments. In the heuristic of Chang, Jyu, and Feng [CHAN89] wire segments rather than vias are considered one at a time. The order in which wire segments is considered is determined by a weighting function. Suppose that wire segment x is being considered for layer assignment. Let m be the number of layers to which the wire segments that cross or overlap with x and which have already been assigned to a layer have been assigned. If $m = 3$, then a backtracking procedure is invoked to modify some layer assignments so as to accommodate x . If $m = 2$, then there is only one layer to which x may be assigned and it is assigned to this layer. If $m < 2$, then a priority function is used to determine which of the possible layers x is to be assigned.

In our heuristic, we process the horizontal tracks to which horizontal wire segments are assigned from the outside to the inside. So, the tracks of the layout of Figure 26 are processed in the order Tr_1 ,

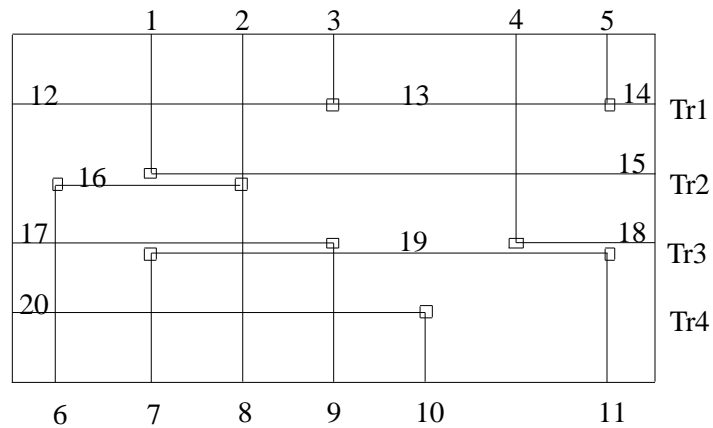


Figure 26 Example of a four track layout

Tr4, Tr2, Tr3. Since our heuristic does not make use of the initial layer assignment that may be part of the input, our example of Figure 26 does not show this. When a track is processed, the horizontal net segments assigned to it are assigned to layer one or three based on a greedy criteria that will be specified later. Vertical net segments that have one end point on the track being considered and the other on a previously considered track that is on the same side (upper or lower) of the layout region as the current track are also assigned to a layer at this time. So, when track 1 of Figure 26 is considered, the HWSs (horizontal wire segments) 12, 13, and 14 and the VWSs 3 and 5 are assigned to layers; when track 2 is considered the segments 1, 2, 15, and 16 are assigned; and when track 3 is considered the segments 4, 7, 9, 11, 17, 18, and 19 are assigned to layers. Once all tracks have been processed in this way, the unassigned vertical net segments are assigned to layers. Finally, there is a post processing step in which some of the layer assignments are changed.

Our heuristic does not deal directly with HWSs. Rather, it works with *horizontal net segments* (HNS). These are obtained by combining together adjacent HWSs from the same net that are assigned to the same track. So, in the example of Figure 26, the HWSs 12, 13, and 14 constitute a single HNS. Each of the remaining HWSs defines a distinct HNS. A high level description of our layer assignment heuristic is provided in Figure 27.

We now present the details or steps S1-S4. First consider step S1. The HNSs in a track are classified as either single or double. A single HNS is one which does not overlap with any other net

```

{phase 1: track-by-track layer assignment}
for each track  $i$  in outside-to-inside order do
begin
  S1: assign the HNSs of this track to layers;
  S2: assign the VWSs that have one end point on track  $i$ ,
      the other end point on an already considered track,
      and which do not pass the middle of the routing region;
end;
  S3: assign the as yet unassigned VWSs to layers;
{phase 2: post processing}
  S4: reassign HNSs in an effort to reduce the number of vias;

```

Figure 27 High level description of our heuristic

segment in the track. A HNS that is not a single HNS is a double HNS. In the example of Figure 26, the HNSs (12, 13, 14) and 20 are single HNSs while 15-19 are double HNSs. If x is a double HNS, then we compute a set X of HNSs in the same track as x by beginning with $X = \{x\}$ and adding to x other HNSs that overlap with one or more HNSs already in X . This addition of HNSs to X is continued until no more HNSs can be added to X . So, if $x = 17$ (cf. Figure 26), then $X = \{17, 18, 19\}$. For every double HNS x , the set X may be partitioned into two sets A_x and B_x such that the HNSs in A_x do not overlap and those in B_x do not overlap. For the case $x = 17$, $A_{17} = \{17, 18\}$, and $B_{17} = \{19\}$.

For each single HNS, x , in the track under consideration, we compute n_1 (n_3), the number of via candidates that can be eliminated by assigning x to layer 1 (3). This is done by considering the via candidates that are on x . Let Z be such a via candidate. Let V be all the VWSs (at most two) that connect to Z . If all the HNSs that cross VWSs in V have been assigned to layers and if x and all the VWSs in V can be assigned to layer 1 (3) with no conflicts then increase n_1 (n_3) by 1. In case x is a double HNS then n_1 (n_3) is computed for the set X by determining how many of the via candidates on the HNSs of X can be eliminated if the HNSs in A_x (B_x) are assigned to layer 1 and those in B_x (A_x) are assigned to layer 3. For example, when track 2 of Figure 26 is being considered and $x = 16$, $X = \{15, 16\}$, $A_{16} = \{15\}$, and $B_{16} = \{16\}$. Suppose that the HNSs $\{12, 13, 14\}$, and $\{20\}$ have been assigned to layer 1. If $A_{16} = \{15\}$ is assigned to layer 1, then $B_{16} = \{16\}$ must be assigned to layer 3. The via candidate at the junction of 1 and 15 can be saved only by assigning these two segments to the same layer. However the

VWS 1 cannot be assigned to layer 1 as it crosses the HNS {12, 13, 14}. For the via candidates at the junctions of segments 6 and 16 and 2, 8, and 16 such a consideration is not made at this time as each has a VWS (6 and 8, respectively) that crosses an unassigned track (i.e., track 3). So, $n1 = 0$. If A_{16} is assigned to layer 3, then B_{16} must be assigned to layer 1. In this case the via at the junction of segments 1 and 15 is saved as the VWS 1 can be assigned without conflict to layer 3. So, $n3 = 1$. Under the conditions described above, our heuristic would assign A_{16} to layer 3 and B_{16} to layer 1.

For a single HNS, x , $nn1$ ($nn3$) estimates the number of via candidates that might be saved by assigning x to layer 1 (3). As in the case of $n1$ and $n3$, we consider the via candidates on x . Let Z be such a via candidate that has exactly two VNSs connected to it and such that for one of these VNSs, y , all HWSs that cross it have already been assigned to a layer and for the other VNS, z , there is at least one HNS that crosses it and has not been assigned to a layer. If x , y , and z can be assigned to layer 1 (3) without a conflict with any of the already assigned segments then $nn1$ ($nn3$) is increased by 1. When x is a double HNS, $nn1$ and $nn3$ are defined in a similar way but by considering all the HNSs in X as for the case of $n1$ and $n3$. Note that the via candidates Z that can contribute to the count of $nn1$ and $nn3$ are different from those that can contribute to the count of $n1$ and $n3$. As an example consider the assignment for track 2 of Figure 26 under the assumption of that {12, 13, 14} and 20 have been assigned to layer 1. As before, assume $x = 16$, $X = \{15, 16\}$, $A_{16} = \{15\}$, $B_{16} = \{16\}$. The via candidate at the junction of segments 2, 8, and 16 could not be considered in the count for $n1$ or $n3$ but it can be so considered in the count for $nn1$ and $nn3$. If A_{16} is assigned to layer 1, then B_{16} is assigned to layer 3. Segments 2 and 8 can be assigned to layer 3, without crossing any HNS already assigned to layer 3. So, $nn1 = 1$. However, $nn3 = 0$ as if A_{16} is assigned to layer 3, then B_{16} is assigned to layer 1 and if the VWS 2 is assigned to layer 1 it will cross the preassigned HNS {12, 13, 14} that is on this layer.

A third set of counts $nnn1$ and $nnn3$ may also be defined for each single HNS x . Consider the set, Y , of VWSs that cross x . For each $z \in Y$ increase the $nnn1$ ($nnn3$) by 1 iff all the HNSs that both cross z and are assigned to a layer have been assigned to layer 1 (3). The extension to the case when x is a double net segment is analogous to the extensions for the cases of $n1$, $n3$, $nn1$, and $nn3$. As an example, we again consider the assignment of track 2 of Figure 26 under the same assumptions as before. Since $X = \{15, 16\}$, we consider the VWSs that cross both 15 and 16. So, $Y = \{4\}$. The only HNS that crosses 4 that has already been assigned a layer is {12, 13, 14}. Since this has been assigned to layer 1 and since $A_{16} = \{15\}$, $nnn1 = 1$. Also, $nnn3 = 0$.

The counts described above are used in the implementation of step S1 of Figure 27. A high level description of this implementation is given in Figure 28. As can be seen, $nn1$, $nn3$, $nnn1$, and $nnn3$ are used only as tie breakers and so need to be computed only in case of a tie (i.e., $nn1$ and $nn3$ are to be

computed only if $n1 = n3$; $nn1$ and $nn3$ are needed only when $n1 = n3$ and $nn1 = nn3$).

Let x be an unassigned HNS on the current track;
if x is a single HNS **then** [let $X = \{x\}$, $A_x = \{x\}$, and $B_x = \emptyset$]
else let X , A_x , and B_x be as defined for a double HNS;
case
 $(n1 < n3)$ **or** $((n1 = n3)$ **and** $(nn1 < nn3)$ **or** $((n1 = n3)$ **and** $(nn1 = nn3)$ **and** $(nnn1 < nnn3))$:
 assign the HNSs in A_x to layer 1 and those in B_x to layer 3;
 else assign the HNSs in A_x to layer 3 and those in B_x to layer 1;
endcase

Figure 28 Step S1 of Figure 27

For step S2 of Figure 27, we consider only those unassigned VWSs that have one end point on the current track, the other on a previously considered track, and which do not pass through the middle of the routing area. These VWSs have the property that all HNSs that cross them have already been assigned to a layer. If any of these can be assigned to layer 1 or 3 without a conflict they are so assigned. Otherwise, they are assigned to layer 2. When the HNSs of track 2 of Figure 26 have been assigned to layers, we consider the VWSs 1 and 2. The VWSs 6 and 8 are not considered as they pass through the middle of the routing region (i.e., from the top half to the bottom half). If $\{12, 13, 14\}$ and $\{16\}$ have been assigned to layer 1 and $\{15\}$ to layer 3, then VWS 1 can be assigned to layer 3 without conflict. However, VWS 2 cannot be assigned to either layer 1 or 3 without conflict. So, it is assigned to layer 2 in step S2.

In step S3 the unassigned VWSs are considered one at a time (in any order) and assigned to layer 1 or 3 if possible (i.e., if such an assignment creates no conflict). Otherwise the VWS is assigned to layer 2. In step S4 we attempt to further reduce the number of vias by moving all segments associated with a via from layers one and three to layer 2. This can be done only if these segments do not cross or overlap with segments already in layer 2.

By using appropriate data structures, our heuristic can be implemented so as to run in $O(N+C)$ time where N is the number of wire segments (both horizontal and vertical) and C is the number of crossovers between HWSs and VWSs. The observed run times can be reduced somewhat by not attempting to eliminate unremovable vias (Lemma 7); noting that absolutely removable vias (Lemma 8) can be eliminated easily; by considering trivial wire segments in the end (i.e., they are eliminated from

the initial layout and reintroduced after the heuristic has been run) and by observing the restriction of certain wire segments to layer 2 or to layers 1 or 3 (Lemma 10).

5. Backtracking Algorithm

We propose a standard backtracking algorithm [HORO78] to obtain an optimal solution. For this we need to specify the order in which the net segments are to be considered for layer assignment and also the bounding functions to be used.

Net Segment Ordering

By using Lemma 10 certain VWSs can be restricted to be on layer 2 and certain HWSs can be restricted to be on layers 1 or 3. As a result of this, only segments not restricted to layer 2 need to be ordered for layer assignment. Furthermore, trivial net segments can be excluded as these may be easily assigned to layers in the end. For HWSs restricted to be on layers 1 or 3 a two way branching of the search tree is used while a three way branching is used for the remaining segments. The segment ordering is obtained in the following way. If a segment is required to be assigned to a particular layer as a result of layer assignments already made, then it is considered next. For example, a HWS may be required to be on layer 3 because it overlaps with one already assigned to layer 1 and it crosses a VWS assigned to layer 2. If a next segment cannot be chosen in this way, then the unassigned segments are assigned a priority and the one with highest priority selected. In case of a tie, a tie breaker is used. For each segment, we compute the number of different layers to which the segments that cross it or overlap with it have been assigned. This is the *priority* of the segment. Note that the priority is either 0 or 1 because a priority of 2 would mean this segment can be assigned to only one layer feasibly. So it should have been selected as the next segment without a priority computation. A priority of 3 would mean that the segment cannot be assigned to any layer without conflict. So, the current partial assignment is infeasible and should be aborted.

For the tie breaker, we use the number of unassigned HWSs and VWSs that cross or overlap with it. A segment with a highest tie breaker value is used as the next segment. If there is still a tie, it is broken arbitrarily. Further when there is a choice, layers are considered in the order 2, 1, 3.

Bounding Functions

The infeasibility criteria mentioned above is used. Additionally, the unremovable vias are not used in the count for the number of vias in the current partial solution. I.e., we are only concerned with the vias in the possibly removable category that are actually used in the current layer assignment. If this equals or exceeds that of the best solution found so far, the current solution may be aborted.

The backtracking algorithm just described may be converted into an ϵ -approximation backtracking algorithm by aborting the current solution if the number of possibly removable vias used by it plus the number of unremovable vias exceeds $(1+\epsilon)(\text{number of unremovable vias} + \text{number of possibly removable vias in the best solution found so far})$.

6.Experimental Results

We programmed our heuristic and backtracking algorithm in Pascal and conducted experiments on an Apollo 3500 workstation. For test data we used the data of Chang, Jyu, and Feng [CHAN89]. Table 1 gives the characteristic of the 11 instances used in our experiments. The first column gives the number of wire segments in the instance and the second gives the number of vias in an optimal three layer assignment. These optimal values were obtained using our backtracking algorithm. Values with an asterisk are the best values obtained. These were obtained by our heuristic as our backtracking algorithm was unable to do better in the allotted (30 hours) time. These values may not be optimal. The third column gives the number of via candidates in the instance; the fourth gives the number, N_r , of absolutely removable vias; the fifth column gives the number, N_u , of unremovable vias; and the number of trivial wire segments is given in column 6. The last two instances of Table 1 correspond to Deutsch's difficult example [DEUT83].

Table 2 gives the number of vias in the solution generated by five different algorithms and the time taken by each. These figures are provided only for the eight largest instances of Table 1. The column labeled Optimal Alg. is for backtracking, the next is for the 90% approximate version of backtracking (i.e., $\epsilon = 0.9$), the next column is for the heuristic of Section 4, the next is for a heuristic that differs from that of Section 4 only in that the tracks are considered in an inside-to-outside order rather than the outside-to-inside order. The last column is for the heuristic of Chang, Jyu, and Feng [CHAN89]. In this column, the times are on a Sun3/110c workstation and the programming language used is C. This column is taken from [CHAN89]. We do not present the corresponding figures for the heuristic of Chang and Du [CHAN88] as Chang, Jyu, and Feng [CHAN89] have established that their heuristic is superior. As can be seen from Table 2, our outside-to-inside heuristic outperforms the

WSs	Opt.Vias	Via Cand.	Nr	Nu	Trivial WSs
29	1	20	11	0	13
35	0	23	20	0	30
42	1	28	11	0	22
101	33	57	11	27	18
121	48	68	9	47	15
190	77	107	16	77	23
220	96	125	22	91	32
258	90	150	27	78	44
316	104*	180	36	67	64
520	206*	290	32	163	49
586	238*	347	33	186	73

* value obtained by our heuristic

Table 1 Instance characteristics

remaining heuristics in seven of the eight cases. The solution is optimal for only two of the five instances for which the backtracking algorithm was able to complete in 1 hour. The run time of the outside-to-inside heuristic is also quite satisfactory.

7. Conclusions

We have shown that the 2CCVM and HVH CVM problems are NP-hard. A new heuristic and a backtracking algorithm have been proposed for the HVH CVM problem. The backtracking algorithm may also be used for three layer non HVH CVM instances. It is recommended for small instances (say with 200 or fewer segments) and the heuristic (outside-to-inside) is recommended for larger instances.

	Optimal Alg		90% Approx. Alg		Heur.(Out-to-In)		Heur.(In-to-Out)		Heur. by Chung	
	vias	time	vias	time	vias	time	vias	time	vias	time
101	33	1.8s	36	1.19s	40	.36s	40	.37s	34	.16s
121	48	1.1s	50	.63s	48	.43s	52	.44s	50	.22s
190	77	1.4s	82	.87s	78	.63s	80	.64s	83	.58s
220	96	3.1s	100	1.43s	96	.72s	98	.74s	99	.58s
258	90	17m	90	1.47m	94	.79s	102	.80s	105	.75s
316	114*	1h*	117*	1h*	104	.87s	114	.89s	108	1.3s
520	219*	1h*	218*	1h*	206	1.6s	213	1.6s	214	2.5s
586	255*	1h*	267*	1h*	238	1.7s	242	1.8s	245	2.7s

* the value after one hour of execution

Table 2 Experimental results
(s:seconds, m:minutes, h:hours)

8. References

- [CHAN87] K. Chang and D. Du, 'Efficient Algorithm for Layer Assignment Problem', IEEE Trans. on Computer-Aided Design, Vol.CAD-6, No.1, Jan 1987. pp 67-78.
- [CHAN88] K. Chang and D. Du, 'Layer Assignment Problem for Three-Layer Routing', IEEE Trans. on computers, Vol.37, No.5, May 1988, pp 625-629.
- [CHAN89] K. Chang, H. Jyu, and W. Feng, 'The Constrained Via minimization for Three-Layer Routing', unpublished manuscript.
- [CHEN82] R. Chen, Y. Kajitani, and S. Chan, 'Topological Considerations of the Via Minimization Problem For Two-Layer PC Boards', in ISCAS, 1982, pp 968-971.
- [DEUT83] D. Deutsch, 'A Dogleg Channel Router', in Proc. 20th Design Automation Conf., June 1983, pp 591-597.
- [EHRL76] G. Ehrlich, S. Evan, and R. Tarjan, Intersection Graphs of Curves in the Plane, J. of combinatorial theory (B) 21, 8-20(1976), pp 8-20
- [HASH71] A. Hashimoto and J. Stevens, 'Wire routing by optimizing channel assignment within large apertures', in Proc. 8th Design Automation Workshop, 1971, pp 155-169.
- [HORO78] E. Horowitz and S. Sahni, "Fundamentals of Data Structures", First Edition, Computer

Science Press, Maryland, 1978.

- [HSU83] C. Hsu, 'Minimum-via topological routing', IEEE Trans. on Computer-Aided Design, Vol.CAD-2, No.4, Oct.1983, pp 235-246.
- [MARE84] M. Marek-Sadowska, 'An Unconstrained Topological Via Minimization Problem for Two-Layer Routing', IEEE Trans. on Computer-Aided Design, Vol.CAD-3, No.3, July 1984, pp 184-190.
- [NACL87] N. Naclerio, S. Masuda, and K. Nakajima, 'Via Minimization for Gridless Layouts', 24th ACM/IEEE Design Automation Conference, Paper8.4, 1987. pp 159-165.
- [NACL89] N. Naclerio, S. Masuda, and K. Nakajima, 'The Via Minimization is NP-complete', IEEE Trans. on Computer-Aided Design, Vol.CAD-38, No.11, Nov,1989, pp 1604-1608.
- [STOC73] L. Stockmeyer, Planar 3-colorability is Polynomial Complete, SIGACT News (ACM Publication), 3(July 1973), pp 19-25
- [XION88] X. Xiong and E. Kuh, 'The Constrained Via Minimization Problem for PCB and VLSI Design' in Proc. 25th ACM/IEEE Design Automation Conference, 1988, pp 573-578.