An Approximation Algorithm for the Via Assignment Problem

by

Teofilo F. Gonzalez
Programs in Computer Science
The University of Texas at Dallas
Richardson, Texas 75080

ABSTRACT: We consider the via assignment problem that arises in multilayered printed circuit board routing. An efficient approximation algorithm for this problem is presented. The algorithm is of (low) polynomial time complexity and guarantees solutions that use no more than 3 * s via columns, where s is the number of via columns in an optimal solution. Several issues relating to the computational complexity of via assignment problems are discussed.

KEYWORDS: multilayer printed circuit boards, routing, via assignment, polynomial time approximation algorithms, NP-complete problems.

I. Introduction.

Large-scale computer systems are built by interconnecting silicon chips. The interconnections are carried out by placing the chips on a multilayer printed circuit (MPC) board. The components (chips) are mounted on top of these boards and their terminals are inserted in drilled-through holes called pins. Terminals from different chips are connected by printed wires located on any of the layers in the MPC board. Enough vias have been added for interlayer connections and it is known precisely the vias and pins that need to be connected. The MPC routing problem consists of finding physical routes for the printed wires.

In this paper we make the same basic assumptions made by So [SO]. These are:

(1) The pins and vias are at fixed locations. Vias appear only column-wise.
(2) Only points (pins or vias) on the same line (row or column) can be connected directly and the physical routes must be confined within the channels on both sides of the line.
(3) All row connections are in one layer and all column connections are on the other.

Under these assumptions So has shown that the MPC routing problem can be reduced to several single-line single-layer routing problems. The single-line single-layer problem can be solved by the algorithms given in [KKF] and [RS].

In practice we are given a set of components placed on a MPC board together with sets of terminals to be interconnected. Each set of terminals to be interconnected is called a net. One can solve this problem with any algorithm for the MPC routing problem if one adds some via columns and specifies the vias to be connected to each point (pin or via). Since the addition of via columns increases the cost of the board, it is desirable to add the least number of via columns. This problem is known as the via assignment problem (VAS) and it was initially studied in [TKS] and [TSA]. In [TKS] it was shown that this problem can be solved efficiently when only one via column is needed for all the interconnections. However, when three or more via columns are required, the problem is NP-hard ([TKS], [TSA]). These results rely on the restriction that no net is connected to vias from more than one via column. If this restriction is relaxed, it has been conjectured [TKS] that the problem remains in the class of problems known as NP-hard. We shall refer to this relaxed version of the problem as the multi-via assignment problem (MVAS). In this paper it is shown that the MVAS problem is NP-hard. Furthermore, it is shown that the VAS and MVAS problems are NP-hard even when only two via columns are required for all the interconnections. Our results finely separate "difficult" from "easy" cases of the problems.

Because of the computational difficulty in obtaining an optimal solution to the VAS and MVAS problems, we turn our attention to the study of algorithms that generate suboptimal solutions to these problems, i.e., generate solutions that use a number of via columns that is not far from the optimal number. In [TKS] and [TSA] several approximation algorithms to solve these problems are presented. However these algorithms do not always generate solutions that are close to optimal. In fact the solutions given by these algorithms can be arbitrarily far from the optimal ones. The problem of obtaining a solution within 100% of the optimal solution for the VAS problem is NP-hard. This can be easily shown by using the reduction outlined in [TKS] and the results in [GJ] relating to the complexity of generating approximate solutions to the graph coloration problem. The results in [TKS] can also be used to show that any approximation algorithm for the VAS problem is also an approximation algorithm for the graph coloration problem. The converse does not seem to be true. The approximation problem for graph coloration has been widely studied [GJ] and seems to be computationally intractable. Therefore it is unreasonable to try to obtain an efficient approximation algorithm for the VAS problem before one can be found for the graph coloration problem. Because of this we turn our attention to the MVAS problem. Our main result is an efficient algorithm that guarantees solutions close to optimal for the MVAS problem.

Our contribution, on the theoretical aspects of the VAS and MVAS problem, is to settle some issues relating to their computational complexity. On the practical side, for the MVAS problem we present an efficient algorithm that generates solutions with an objective function value that in the worst case is not far from the objective function value of an optimal solution.

In what follows we define the via assignment problem and in section II we show that some versions of this problem are NP-complete. An approximation algorithm for the MVAS problem is presented in section III.

Let P be a multilayered printed circuit board. Board P consists of r rows and c columns. In the intersection of a row with a column there is a pin. Each pin belongs to at most one net. The nets are represented by $N_1, N_2, \ldots, N_n$ and each N_i consists of a set of pins that need to be interconnected. All pins in each net have to be made electrically common. This is accomplished by interconnecting the pins by wires. Vias can be added for interlayer connections, however vias appear only column-wise. Wire segments can only connect points (pins or vias) located in the same line (row or column). It is assumed that all the wire segments connecting points in the same row are in one layer and the ones connecting points in the the same column are in the other layer. A wire in one layer can only be connected to a wire in another layer if both wires are connected to the same pin or via. The VAS problem consists of adding the least number of via columns in such a way that all pins in each net can be interconnected by sets of wires that satisfy the above requirements and no net is interconnected by using vias from more than one via column. The MVAS is defined similarly, except that the restriction that each net must be connected by using vias from at most one via column is relaxed, i.e., it is possible to use vias from two or more via columns for the interconnection of all the pins in a net. Since every feasible solution to the VAS problem is also a feasible solution to the MVAS problem, the MVAS problem has an optimal solution value that is never worse than the optimal solution value for the corresponding VAS problem. It is simple to show that the converse is not true [TKS]. An instance of the VAS problem is shown in example 1.1 (figure 1.1) and one of its solutions is depicted in figure 1.2.

Example 1.1: An instance of the VAS or MVAS problem.

$r = 5$, $c = 5$, $n = 3$,

$N_1 = \{ (2,1) , (3,4) , (4,1) , (5,5) \}$,

$N_2 = \{ (1,4) , (4,3) , (4,5) , (5,3) \}$,

and $N_3 = \{ (1,3) , (1,5) , (3,2) , (3,5) , (5,2) \}$.

row \ col	1	2	3	4	5
1			3	2	3
2	1				
3		3		1	3
4	1		2		2
5		3	2		1

Fig. 1.1: The VAS or MVAS problem in example 1.1.

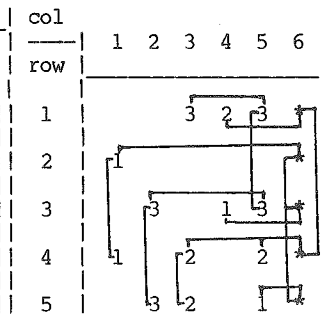

Fig. 1.2: A solution to the problem in example 1.1.

Our NP-completeness results are established by reducing the exact cover by 3-sets (XC3) problem to the VAS and MVAS problems. The XC3 problem was shown to be NP-complete by Karp [K].

II. NP-completeness Results.

In this section it is shown that the MVAS and VAS problems are NP-complete even when only two via columns are needed for all the interconnections. This result is obtained by reducing the XC3 problem to the VAS problem. The NP-completeness of the MVAS problem is established by using the same reduction and the observation that in the instance constructed from XC3 no net can be connected using vias from more than one column in any solution that requires two via columns. The problem of deciding whether an instance of the VAS problem can be routed by using at most two via columns is referred to as the 2-VAS problem. The 2-MVAS problem is defined similarly.

Theorem 2.1: The 2-VAS problem is NP-complete.
Proof: For brevity the proof is omitted. []

Theorem 2.2: The 2-MVAS problem is NP-complete.

Proof: The proof is similar to the one used for theorem 2.1.[]

III. Approximation Algorithm.

In this section we describe an efficient approximation algorithm for the MVAS problem. The algorithm is of polynomial time complexity and generates solutions that use at most $3 * \alpha$ via columns, where α is the number of via columns in an optimal solution. As input we are given an instance, Y, of the MVAS problem. Our approximation algorithm consists of three major steps. In the first two steps of the algorithm we construct an instance of the VAS problem, X, in which all nets have exactly two pins. Problem X has the property that the maximum number of pins sharing the same row in the MPC board is not greater than $2 * \alpha$, where α is a lower bound on the number of via columns in an optimal solution to Y. Furthermore, a simple algorithm can be used to construct from any solution to X a solution to Y with the same number of via columns as the one in the solution for X. The specific computations performed in these steps are to construct a bipartite graph and then find a "special" subset of edges in it. The "special" subset of edges and a decomposition of the nets in Y are then used to define the nets for X.

The third step of the algorithm consists of finding an approximate solution to X. An approximate solution to this problem is required since it is an NP-hard problem. The specific computations performed in this step are to construct a multigraph from X, then the edges of this multigraph are "colored" and from any such coloration we obtain a solution to X. The number of via columns used in our solution to X is not greater than $(3/2) * \beta$, where β is the maximum number of nets sharing the same row in X. A detailed description of the algorithm appears in appendix A.

IV. References.

[BM] Bondy, J. and U. S. R. Murty, "Graph Theory with Applications," Elsevier North Holland Inc., 1976.

[GJ] Garey M. and D. Johnson, "The Complexity of Near-Optimal Graph Coloring", JACM, 23, No. 23, 43-49.

[G] Gonzalez T., "Remarks on the Via Assignment Problem," Technical Report # 129, The University of Texas at Dallas, April 1983.

[HK] Hopcroft, J., and R. Karp, "An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs," SIAM J. Comput., 1973, pp. 225-231.

[K] Karp R., "Reducibility Among Combinatorial Problems," In Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, Eds., Plenum Press N.Y., 1972, pp.85-104.

[KKF] Kuh E., T. Kashiwabara and T. Fujisawa, "On Optimal Single-Row Routing", IEEE Transactions on Circuits and Systems, Vol. CAS-26, No. 6, June 1979.

[RS] Raghavan R. and S. Sahni, "Optimal Single Row Router", IEEE TC, Vol. C-32, #3, March 1983, pp. 209-220.

[SO] So H., "Some Theoretical Results on the Routing of Multilayer Printed-Wiring Boards", IEEE Symposium on Circuits and Systems 1974, pp. 296-303.

[TKS] Ting B., E. Kuh and A. Sangiovanni-Vincentelli, "Via Assignment Problem in Multilayer Printed Circuit Board", IEEE Transactions on Circuits and Systems, Vol. CAS-26, No. 4, April 1979.

[TSA] Tsukiyama S., I. Shirakawa and S. Asahara, "An Algorithm for the Via Assignment Problem in Multilayer Backboard Wiring", IEEE Transactions on Circuits and Systems, Vol. CAS-26, No. 6, June 1979.

[V] Vizing, V. G., "On an estimate of the chromatic class of a p-graph," Diskret. Analiz. (in Russian), 3, 1964, pp. 25-30.

Appendix A: The Algorithm.

Before presenting our algorithm we outline in more detail the three major steps in it. Additional definitions as well as some intermediate computations have been introduced to clarify our algorithm.

STEP I:
The set of pins in each net N_i is partitioned into the sets $N_{i,1}$, $N_{i,2}$,..., N_{i,ℓ_i} such that all the

pins in each set $N_{i,j}$ can be made electrically common without the use of vias and all the pins in each set $N_{i,j}$ cannot be made electrically common with any pin in $N_{i,k}$ ($k \neq j$) without the use of vias. We shall refer to the sets $N_{i,j}$ as subnets. For $N_{i,j}$ we define $R_{i,j}$ as the set of rows in which the pins in $N_{i,j}$ lie. The sets $N_{i,j}$ and $R_{i,j}$ for the MVAS problem given in example 1.1 are:

$N_{1,1} = \{ (2,1) , (4,1) \}$ $\quad R_{1,1} = \{ 2,4 \}$,
$N_{1,2} = \{ (3,4) \}$ $\quad R_{1,2} = \{ 3 \}$,
$N_{1,3} = \{ (5,5) \}$ $\quad R_{1,3} = \{ 5 \}$,
$N_{2,1} = \{ (1,4) \}$ $\quad R_{2,1} = \{ 1 \}$,
$N_{2,2} = \{ (4,3) , (4,5) , (5,3) \}$ $\quad R_{2,2} = \{ 4,5 \}$,
and $N_{3,1} = N_3$ $\quad R_{3,1} = \{ 1,3,5 \}$.

Every net with $\lambda_i = 1$ can be eliminated since it needs no via column for its connection. If all nets are eliminated then our solution is an optimal solution since it uses no via columns for the connections. In example 1.1, net N_3 is the only net eliminated. From now on, n represents the number of nets not yet been eliminated.

In this step we select a row with at least one pin from each subnet $N_{i,j}$. Any pin from subnet $N_{i,j}$ located in the row selected will be used for the connection of $N_{i,j}$ to all other subnets of net N_i. In other words, one element from each $R_{i,j}$ will be selected. The selection of these elements is performed in such a way that the maximum number of subnets sharing the same row for its connection, is minimized. We shall refer to this minimum value as α. The value for α can be shown to be a lower bound on the number of via columns used in an optimal solution to the original MVAS problem. Our final solution uses no more than 3 * α via columns.

The selection of the row in each subnet is made by constructing an optimal complete s-matching in a bipartite graph. This is explained in what follows. Let $S = \{ (i , j) \mid N_{i,j}$ is a subnet$\}$; $T = \{ 1 , 2 , \ldots , r \}$ and $E = \{ ((i , j) , k) \mid k \in R_{i,j} \}$. Clearly $G = (S \cup T , E)$ is a bipartite graph. The bipartite graph obtained for example 1.1 is depicted in figure A.1.

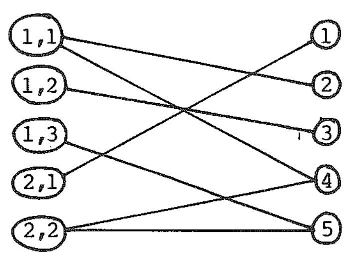

Fig. A.1: Bipartite graph constructed for example 1.1.

An s-matching, I, for G is a subset of edges such that no node in set S is adjacent to more than one edge in set I. A complete s-matching is an s-matching that includes every node in set S. For a complete s-matching, I, we define as M(I), the maximum number of edges in I adjacent to any node in set T. We say that the complete s-matching, I, is an optimal complete s-matching (ocs-matching) for G if every complete s-matching Z for G has $M(Z) \geq M(I)$. The set of edges $I = \{ ((1,1),2) , ((1,2),3) , ((1,3),5) , ((2,1),1) , ((2,2),4) \}$ is an ocs-matching for the bipartite graph depicted in figure A.1. The problem of selecting a row for the connection of each subnet reduces to finding an ocs-matching for G. In appendix C we present an efficient algorithm to construct an ocs-matching for G. Given an ocs-matching, I, for G we find $r_{i,j}$, the row to be used for the connection of subnet $N_{i,j}$ to all other subnets of net N_i, as follows: $r_{i,j} = k$ if edge ((i , j) , k) is in the ocs-matching I. Clearly, from the definition of an ocs-matching we can see that each subnet will have one and only one row selected and in such a row there is at least one pin and if "I" is an ocs-matching for G then $\alpha = M(I)$. The optimality of the complete s-matching will guarantee that there is no feasible solution to Y that uses less than α via columns, where α is defined above. The $r_{i,j}$ values for the ocs-matching obtained above are: $r_{1,1} = 2$, $r_{1,2} = 3$, $r_{1,3} = 5$, $r_{2,1} = 1$ and $r_{2,2} = 4$. []

STEP II:

At this point we construct an instance, X, of the VAS problem, such that from a solution to X it is simple to construct a solution to Y that uses the same number of via columns as the ones used in the solution to X. Also, it is shown that no row in X contains more than 2 * α pins.

Let $L = \sum_{i=1}^{n} \lambda_i$. The instance X consists of $n' = L - n$ nets and the board has $r' = r$ rows and $c' = 2n'$ columns. We shall refer to the n' nets as $N'_{i,1} , \ldots , N'_{i,\lambda_i - 1}$ for $1 \leq i \leq n$. Net $N'_{i,j}$ has two pins, one is located at position $(r_{i,j},k)$ and the other pin at position $(r_{i,j+1},k+1)$, where $k = 2 * (j - 1 + \sum_{z=1}^{i-1} (\lambda_z - 1)) + 1$. Figure A.2 shows the instance X constructed for the problem in example 1.1.

$r = 5, c = 6, n = 3,$
$N'_{1,1} = \{ (2,1) , (3,2) \}$,
$N'_{1,2} = \{ (3,3) , (5,4) \}$,
and $N'_{2,1} = \{ (1,5) , (4,6) \}$.

Figure A.2: Instance X constructed for example 1.1.

The proof for the claim that from any solution to X one can construct a solution to Y is left to the reader as an exercise. []

STEP III:

In this step we find an approximate solution to X, the restricted VAS problem obtained in step II. This solution is obtained by constructing a suboptimal solution to the edge coloration of a multigraph problem. The multigraph that we construct is $M = (V' , E')$, where V' is the set of edges and E' is a multiset of edges, is defined as follows:

$V' = \{ i \mid i$ is a row in X $\}$
and $E' = \{ \{i,j\} \mid$ a net in X uses rows i and j $\}$.

The multigraph M constructed from X given in figure A.2 is shown in figure A.3. An edge coloration for multigraph M consists of assigning the least number of colors to the edges E is such a way that no two edges adjacent to some node have the same color. For the multigraph depicted in figure A.3, an edge coloration for it is: edge {3,5} is colored "one" and all the remaining edges are colored "two". Given a coloration for M, a solution to X can be obtained by simply connecting a net using ith via column if the edge used to represent this net is assigned color "i". Clearly the number of colors used in M is the same as the number of via columns in our solution for X. By construction it is simple to show that the least number of colors in any edge coloration for M is at least β, where β is the is the maximum number of pins in a row of X. In the next section we present an algorithm that finds an edge coloration for M that uses no more than (3/2) * β colors. Hence, in our solution to the original MVAS problem there are at most 3 * α via columns. The solution obtained for the MVAS problem in example 1.1 is shown in figure A.4. []

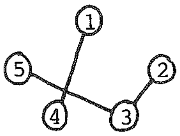

Fig.A.3: Multigraph for ex. 1.1.

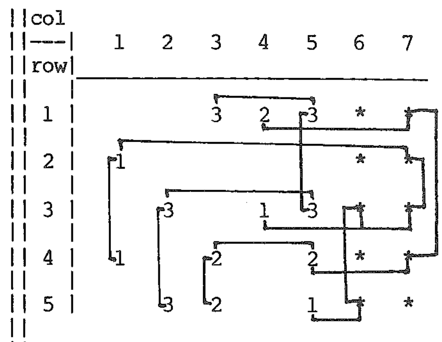

Fig. A.4: Solution constructed for Example 1.1.

Our algorithm is depicted in figure A.5. Additional definitions as well as intermediate steps have been introduced to clarify the exposition of the algorithm.

```
    Algorithm APPROX ( Y //an instance of the MVAS
                            problem// )
      Use Y to construct
                G = ( S (_) T , E )  // step I //
      Obtain an optimal complete s-matching, I,
                for G                    //step I//
      Define r_{i,j} from I          //step I//
      Construct X from r_{i,j}       // step II//
      Use X to construct M = ( V' , E' )
                                        //step III//
      Find a coloration, C, for M       //step III//
      Use C to construct solution S_X for X
                                        //step III//
      Use S_X to obtain the solution S_Y for Y
      Return ( S_Y )
    End of Algorithm
```

Figure A.5: Algorithm APPROX.

Theorem A.1: Algorithm APPROX constructs a solution to Y by introducing at most 3 * OPT via columns, where OPT is the number of via columns in an optimal solution to Y.
Proof: By the above discussion. []

In appendix B we show that an ocs-matching for a bipartite graph $G = (S \cup T, E)$ can be obtained in $O(e\ s^{3/2}\ t^{1/2} \log s)$ time, where $s = |S|$, $t = |T|$ and $e = |E|$. In appendix B it is shown that a coloration using no more than (3/2) * d colors for a multigraph $M = (V', E')$ can be obtained in $O(m\ n^2)$ time, where d is the maximum degree of any node in M, $n = |V'|$ and $m = |E'|$. Using these results we compute the overall time complexity of the algorithm.

Theorem A.2: Algorithm APPROX can be implemented to execute no more than than $O(r^{1/2}\ p^{5/2} \log p)$, where $p = \Sigma s_i$, s_i is the number of pins in net N_i. and r is the number of rows in the board.
Proof: For brevity the proof of this theorem is omitted. []

Appendix B: OCS-Matchings.

We present an algorithm to construct an ocs-matching for G, where $G = (S \cup T, E)$ is a bipartite graph and an ocs-matching is defined in appendix A Let $S = \{ x_1, x_2, \ldots, x_s \}$, $T = \{ y_1, y_2, \ldots, y_t \}$ and $e = |E|$. For a complete s-matching Z, let M(Z) be the maximum degree of any node in $G' = (S \cup T, Z)$. It is simple to show that an ocs-matching, I, has $1 \leq M(I) \leq |S| = s$. Our procedure performs a binary search on the set of integers { 1 ... s } to find the least value, λ, such that G has a complete s-matching, I, with $M(I) \leq \lambda$. Clearly, at most $O(\log s)$ of these tests are required. The problem of testing whether G has a complete s-matching, I, with $M(I) \leq \lambda$ can be reduced to the problem of testing whether a bipartite graph $H = (S \cup T', E')$ has a complete matching for S, where $T' = \{ y_{i,j} \mid y_i \in T \text{ and } 1 \leq j \leq \lambda \}$ and $E' \{ \{x_k, y_{i,j}\} \mid x_k \in S, y_{i,j} \in T' \text{ and } \{x_k, y_i\} \in E \}$. See [BM] for the definition of a complete matching for S in H.

Theorem B.1: Let G, H and λ be as defined above. G has a complete s-matching, I, with $M(I) \leq \lambda$ iff H has a complete matching for S.
Proof: The proof is straightforward and therefore it is omitted. []

Theorem B.2: An ocs-matching for G can be obtained by the above procedure in $O(e\ s^{3/2}\ t^{1/2} \log s)$ time.
Proof: For brevity this proof is omitted. []

In a subsequent paper we present another algorithm to construct an ocs-matching. The algorithm is more efficient than the one presented in this paper.

Appendix C: Coloring the Edges in a Multigraph.

Any multigraph M with maximum node degree, d, can be colored by using at most (3/2) * d colors. The proof of this fact is based on a generalization of Vizing's theorem [V] (see [BM]). It is simple to design an algorithm that constructs such a coloration. The algorithm colors one edge at a time. When considering an edge for coloration, the algorithm first checks if it is possible to color the edge without recoloring any previously colored edge. This operation takes $O(d)$ time. If some edges need to be recolored, this is accomplished by following a procedure similar to the one used in the proof of Vizing's theorem outlined in [BM]. In the worst case, the time complexity of this recoloration is $O(d\ n)$. Since d is $O(n)$, then the time complexity is $O(n^2)$. As there are m edges in M, the overall time complexity of our procedure is $O(m\ n^2)$.

Theorem C.3: Given a multigraph $G = (N, E)$ with maximum node degree d, the edges in E can be colored with no more than (3/2) * d colors by a procedure that requires $O(m\ n^2)$ time, where $n = |N|$ and $m = |E|$.
Proof: By the above discussion. []