A Formal Algorithm for Routing Traces on a Printed Circuit Board

David R. Hedgley, Jr.
A Formal Algorithm for Routing Traces on a Printed Circuit Board

David R. Hedgley, Jr.
Dryden Flight Research Center
Edwards, California
ABSTRACT

This paper addresses the classical problem of printed circuit board routing: that is, the problem of automatic routing by a computer other than by brute force that causes the execution time to grow exponentially as a function of the complexity. Most of the present solutions are either inexpensive but not efficient and fast, or efficient and fast but very costly. Many solutions are proprietary, so not much is written or known about the actual algorithms upon which these solutions are based. This paper presents a formal algorithm for routing traces on a printed circuit board. The solution presented is very fast and efficient and for the first time speaks to the question eloquently by way of symbolic statements.

NOMENCLATURE

The following nomenclature do not appear in alphabetical order as each definition is predicated upon previous definitions.

- $\mu$: the major axis (horizontal or vertical direction) where the sequence $\{J_i\}$ is associated with $\mu$.
- $\beta$: the minor axis (horizontal or vertical direction) where the sequence $\{M_k\}$ is associated with $\beta$.
- RM: a rectangular matrix of evenly spaced grid points where a node is defined to be an arbitrary grid point belonging to RM.
- RM$_h$: RM represented as a sequence of horizontal lines composed of grid points.
- RM$_v$: RM represented as a sequence of vertical lines composed of grid points (fig. 1).

Figure 1. A representation of an internal matrix.
the number of available nodes from left to right at the node, \((p, l)\), that belongs to \(RM_h\) (fig. 1).

the number of available nodes from bottom to top at the node, \((p, l)\), that belongs to \(RM_v\).

the \(i^{th}\) designated node on the major axis where sequence \(\{J_i\}\) is monotonic and bounded. That is, if \((IB, MB)\) and \((IE, ME)\) are the initial and terminal nodes respectively, then \(IB = J_1 > J_2 > J_3 > J_4 \ldots J_n = IE\), or \(IB = J_1 < J_2 < J_3 < J_4 \ldots J_n = IE\). (These inequalities are consistent with the horizontal axis as the major axis. Otherwise, the sequence \(\{J_i\}\) is bounded by MB and ME, and the inequalities will be consistent with the vertical axis as the major axis. Moreover, \(|J_i - J_{i-1}|\) represents the number of available nodes between \(J_i\) and \(J_{i-1}\). That is, the sequence \(\{J_i\}\) is determined by availability (fig. 2). If \(IB = IE\) or \(MB = ME\), then only one major axis (direction) exists.)

the \(k^{th}\) designated node on the minor axis that intersects the major axis at \(J_i\), where \(M_{ki}\) may not be monotonic or bounded for any \(i\). The sequence \(\{M_{ki}\}\) is designated by the assignment of nodes in a cluster, if possible (fig. 2).

the discrete function that first interrogates the availability of a path belonging to either \(RM_h\) or \(RM_v\), which is initiated at the node \((J_i, M_{ki})\), in the major direction. If applicable, the function then subsequently interrogates the availability of a path in the minor direction if, and only if, \(J_i + 1 = IE\) or \(ME\). The values of \(S(J_i, M_{ki})\) are defined as follows:

- NA = No availability (not any space on the major axis is free).
- PA = Partial availability on the major axis.
- TA = Total availability (both axes are free to complete the trace).

Figure 2. A representation of a typical trace.
Definitions of Operations:

\[ C \leftrightarrow D \] \quad \text{the exchange of definitions of } C \text{ and } D.

\[ C \leftrightarrow \neg D \] \quad \text{the exchange of definitions of } C \text{ and } D \text{ cannot take place.}

\[ E[S(b, c)] \] \quad \text{evaluate } S(b, c).

INTRODUCTION

An automated solution to the printed circuit board problem that is both efficient and cost effective continues to be in great demand. Most of the present solutions\(^1\)–\(^4\) are either inexpensive but not efficient and fast, or efficient and fast but very costly. In either case, no solution exists that describes its algorithm symbolically with any degree of elocution, nor are any algorithms sufficiently fast enough to facilitate parts placement for optimization. Many solutions are proprietary, so not much is written or known about the actual algorithms upon which these solutions are based.

The purpose of this paper is to accomplish three objectives:

- To facilitate optimization of arrangement of parts (for example, resistors and capacitors) on printed circuit boards for more sophisticated and expensive systems.
- To make a very fast and efficient algorithm accessible at no cost for individual users or small companies.
- To serve as a pedagogical tool that has heuristic value by presenting a formal symbolic structure that describes the entire algorithm.

FORMAL SYMBOLIC STATEMENTS

The purpose of the following algorithm, presented as a list of symbolic statements, is to construct a path from the source node, \((IB, MB)\), to the target node, \((IE, ME)\). A typical path (fig. 2) is described by the sequence of nodes \((J_1, M_{1_1}), (J_2, M_{1_2}), (J_3, M_{1_3})\), and \((J_3, M_{1_3})\). In this case, \(M_{1_1} = M_{1_2} = M_{5_2} = M_{1_3}\). In general, if \((J_i, M_{k_i})\) is an arbitrary element in a sequence leading to a solution, then the next element in the sequence will be either \((J_{i+1}, M_{1_{i+1}})\) or \((J_i, M_{k+u})\), where \(u \geq 0\). Thus, the algorithm is as follows:

(a) \(N_i > 1, k_i < N_i, \) and \(S(J_i, M_{k_i}) = NA \Rightarrow E[S(J_i, M_{k+1})]\)

(b) \(i > 1, N_i \geq 1, k_i = N_i, \) and \(S(J_i, M_{k_i}) = NA \Rightarrow E[S(J_{i-1}, M_{k+1})]\)

(c) \(i = 1, N_i \geq 1, k_i = N_i, \) and \(S(J_i, M_{k_i}) = NA \Rightarrow \mu \Leftarrow \beta \) or \((IB, MB) \rightarrow (IE, ME)\) and \(E[S(J_1, M_{1_1})]\)

(d) \(i = 1, N_i \geq 1, k_i = N_i, S(J_i, M_{k_i}) = NA, \mu \rightarrow \beta \) and \((IB, MB) \rightarrow (IE, ME) \Rightarrow NI\)

(e) \(S(J_i, M_{k_i}) = PA \Rightarrow E[S(J_{i+1}, M_{1_{i+1}})]\)

(f) \(S(J_i, M_{k_i}) = TA \Rightarrow NI\)
The efficiency of the algorithm is also predicated upon the process that determines the availability of rows and columns and determines the maintenance of the two respective directories. That is, a directory of rows, \( RM_h \), and a directory of columns, \( RM_v \), are maintained where each directory theoretically addresses different sides of the routing board. If required, any subsequent layers or surface mounts are handled similarly.

In general, the determination of the availability at an arbitrary node, \((p, 1)\), with respect to either \( RM_h \) or \( RM_v \) may be addressed accurately, keeping in mind that \( H_{p.1} \) and \( V_{p.1} \) are interchangeable as they relate to the following explanation (fig. 1). For the sake of specificity, \( H_{p.1} \) will be employed here. Thus, let \((u_o, l_o)\) be a target with respect to \((p_o, l_o)\) and where \( u_o > p_o \). Then the availability at \((p_o, l_o)\) is \( \min[H_{p_o,1_o}, u_o - p_o] \). However, if \( u_o < p_o \), then construct the equation \( T_o = H_{u_o,1_o} - (p_o - u_o) \). Consider the two cases \( T_o \geq 0 \) and \( T_o < 0 \). If \( T_o \geq 0 \), then the availability at \((p_o, l_o)\) with respect to \((u_o, l_o)\) is \( p_o - u_o \). If \( T_o < 0 \), let \( T = H_{u_o,1_o} - (p_o - u) \), where \( u_o < u < p_o \). Then the availability at the node \((p_o, l_o)\) that belongs to \( RM_h \) is calculated using the bisection method whose direction of seek is governed by the sign of \( T \). Moreover, because the maximum availability is sought, the number of iterations will be \( n = \lfloor \text{LOG}_2(p_o - u_o) \rfloor + 1 \).

The integrity of the algorithm, which is implemented by recursive descent, is not disturbed by either the definition of the major and minor axes or the specificity of the terminal and initial nodes. Therefore, all four combinations of axes and nodes are transparent to the algorithm itself. Furthermore, an additional criterion to interchange axes or nodes could be predicated on the value of \( i \) or the magnitude of \( k_i \) for some \( i \).

The number of vias can be minimized from the contents of the directories (\( RM_v \) and \( RM_h \)) after the solution is complete. This reduction results in a lower cost for the actual physical implementation of the solution. Finally, the updating of the directories dynamically is trivial, as the \( RM_v \) and \( RM_h \) contents are corrected based on the values of \( J_i \) and the corresponding \( M_{k_i} \) resulting from the monotonic sequence \( \{J_i\} \).

The application of the theory presented confirms the objectives sought. Appendix A presents examples of typical solutions found. Moreover, a comparison study of the state-of-the-art algorithms is made. Appendix B addresses the efficiency, the cost, the ratio of solution to the Manhattan distance, and the central processing unit time.

**CONCLUSION**

An automated solution to the printed circuit board problem that circumvents the devastation of exponential growth as a function of complexity has been presented. Moreover, application of the theory displayed in Appendix A demonstrates its versatility and range. The statistics presented in Appendix B confirm the objectives sought. The computer program based upon the ideas presented here will be offered by COSMIC, located in Athens, Georgia.

*The author wishes to acknowledge Glenn Bever, Rodney Bogue, and Harry Chiles for their many helpful suggestions.*
APPENDIX A
EXAMPLES OF TYPICAL SOLUTIONS*

*Dash lines and solid lines are on different sides of the layer.

Figure A-1. Board without grid interference considerations; grid size = 0.05 in.
Figure A-2. Board with grid interference considerations; grid size = 0.025 in.
Figure A-3. Board without grid interference considerations; grid size = 0.025 in.
APPENDIX B
TABLE OF COMPARISONS*

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Ratio, actual distance/Manhattan distance</th>
<th>Completion rate, percent</th>
<th>Cost, dollars</th>
<th>CPU time, minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shape tech</td>
<td>1.05</td>
<td>98–100</td>
<td>12,000</td>
<td>10</td>
</tr>
<tr>
<td>Pcad</td>
<td>1.10</td>
<td>92</td>
<td>4,000</td>
<td>30</td>
</tr>
<tr>
<td>Hedgley</td>
<td>0.97</td>
<td>93</td>
<td>0</td>
<td>0.25</td>
</tr>
</tbody>
</table>

*Chart statistics are reflective of a one-layer board that is two-sided with equivalent grid sizes.
REFERENCES


A Formal Algorithm for Routing Traces on a Printed Circuit Board

David R. Hedgley, Jr.

NASA Dryden Flight Research Center
P.O. Box 273
Edwards, California 93523-0273

This paper addresses the classical problem of printed circuit board routing: that is, the problem of automatic routing by a computer other than by brute force that causes the execution time to grow exponentially as a function of the complexity. Most of the present solutions are either inexpensive but not efficient and fast, or efficient and fast but very costly. Many solutions are proprietary, so not much is written or known about the actual algorithms upon which these solutions are based. This paper presents a formal algorithm for routing traces on a printed circuit board. The solution presented is very fast and efficient and for the first time speaks to the question eloquently by way of symbolic statements.