OPTIMAL ROUTING OF CRITICAL BUT TYPICAL NETS

Joseph W. Lavinus
James P. Cohoon

Computer Science Report No. TR-92-38
October 20, 1992

OPTIMAL ROUTING OF CRITICAL BUT TYPICAL NETS

Joseph W. Lavinus and James P. Cohoon

Department of Computer Science
University of Virginia
Charlottesville, VA 22903
USA

They were of eternal importance, like baseball
— Sinclair Lewis, *Babbitt*

Abstract

Routing folklore tells us that a typical net has somewhere between two and four terminals. We present algorithms TRIPLE and HOMER that compute an optimal rectilinear Steiner minimal tree for nets with three or four terminals respectively. Unlike traditional rectilinear Steiner minimal tree algorithms, TRIPLE and HOMER account for the presence of logic elements and pre-placed wire segments. The algorithms exploit a theorem that we prove that limits the number of potential routing segments that are examined while constructing an optimal solution. The potential routing segments are derived from a generalization of Hightower's line search escape segments.

I. Introduction

The approach to routing a critical net has generally depended upon whether the net has more than two terminals. For nets with two terminals, numerous researchers have proposed algorithms that analyze the circuit surface to produce routing solutions in the presence of obstacles (logic cells and previously-placed routing segments). In fact, the first published methods were strategies that could quickly produce optimal routing solutions for what were then reasonable-sized problem instances [11, 12]. Since than other researchers have developed optimal algorithms for the much larger-sized instances that arise in the physical design of VLSI systems [3, 10].

The options available to VLSI system developers for the routing of multi-terminal critical nets are vastly different than those for two-terminal routing. In general, a VLSI system developer is forced to use a non-optimal variant of a minimum spanning tree algorithm. This difference is in part due to the complexity of the underlying problem. If we ignore the presence of obstacles, then the problem is equivalent to the NP-complete rectilinear Steiner minimal tree (*RSMT*) problem [8].

The lack of options is also due to the fact that most algorithmic routing research concentrates on the classic RSMT problem, and thus shows little concern for the fact that routing must be done in the presence of obstacles. As examples, research efforts have produced fast (i.e., polynomial running time) algorithms for the RSMT problem where the vertices are constrained to lie on the perimeter of a rectangle [1,4] and for determining a bounded radius minimal Steiner tree [5]. While such research has definite applicability—for the former example it is the routing of a critical net within a channel, and for the latter it is the routing of a critical net with respect to performance-driven layout criteria—it is left to the VLSI system developer to modify the algorithms to produce heuristic solutions for the obstacle-versions of the problems.

We are concerned here with the real problem—the routing of critical nets in the presence of obstacles. In particular, we are concerned with the optimal routing of critical nets with a typical number of terminals. A folk theorem of routing is that a typical net has between two and four terminals. In an effort to experimentally verify this conjecture, we examined the SIGDA Benchmark Suite [13]. Table 1 summarizes the

Number of Terminals in Net	Percentage of Total Nets
2	53
3	28
4	7
>4	12

Table 1: Distribution of Primary 1 nets with respect to number of terminals.

net distribution of the benchmark instance Primary 1. Although the majority of nets for this instance are two-terminal nets, three- or four-terminal nets do comprise slightly more than one-third of the total nets. Thus, they represent a significant fraction of the total nets. As other popular benchmark instances have similar distributions, developing a method to optimally route a critical three- or four-terminal net is very worthwhile. Hence, we focus in this paper on quickly producing optimal routes for such nets.

In the remainder of the paper, we first prove that there is a a limited set of potential routing segments that one can examine to determine an optimal solution for critical three- or four-terminal nets. We then discuss algorithms TRIPLE and HOMER that exploit this theorem to quickly construct optimal rectilinear Steiner minimal trees for instances with three and four terminals respectively. We note that we are currently investigating generalizations for the optimal routing of critical nets with k terminals in the presence of obstacles, where k is a small number.

II. Basics

Consider the custom layout instance depicted in Figure 1(a). In particular, five logic cells and the three terminals of a critical net are shown. Figure 1(c) shows a collection of *escape segments*. These dashed segments can be used to determine an optimal routing for the instance. One such routing is depicted in Figure 1(d).

This form of escape segments was first used in the optimal two-terminal interconnection technique, line intersection routing or LIR [3]. These possible routing segments are a generalization of the line search escape segments used by the heuristic routing technique, LSR, for the same problem [9].

Escape segments are formally a collection of possible routing segments that are generated from the contours of obstacles (cells and previous laid wire) that border routing channels. To show informally that such segments can be used as a basis for determining a net's physical interconnection, we appeal to an analogy to how one travels by automobile from one major city to another. A trip typically begins on local roads that connect to a beltway that surrounds the city. From the beltway one jogs off along a series of highways. One eventually reaches the beltway of the destination city from which local roads are taken to complete the trip.

For the transportation analogy, a circuit surface obstacle corresponds to a city. On each side of an obstacle that borders a routing channel, we generate a potential routing segment that forms a portion of the

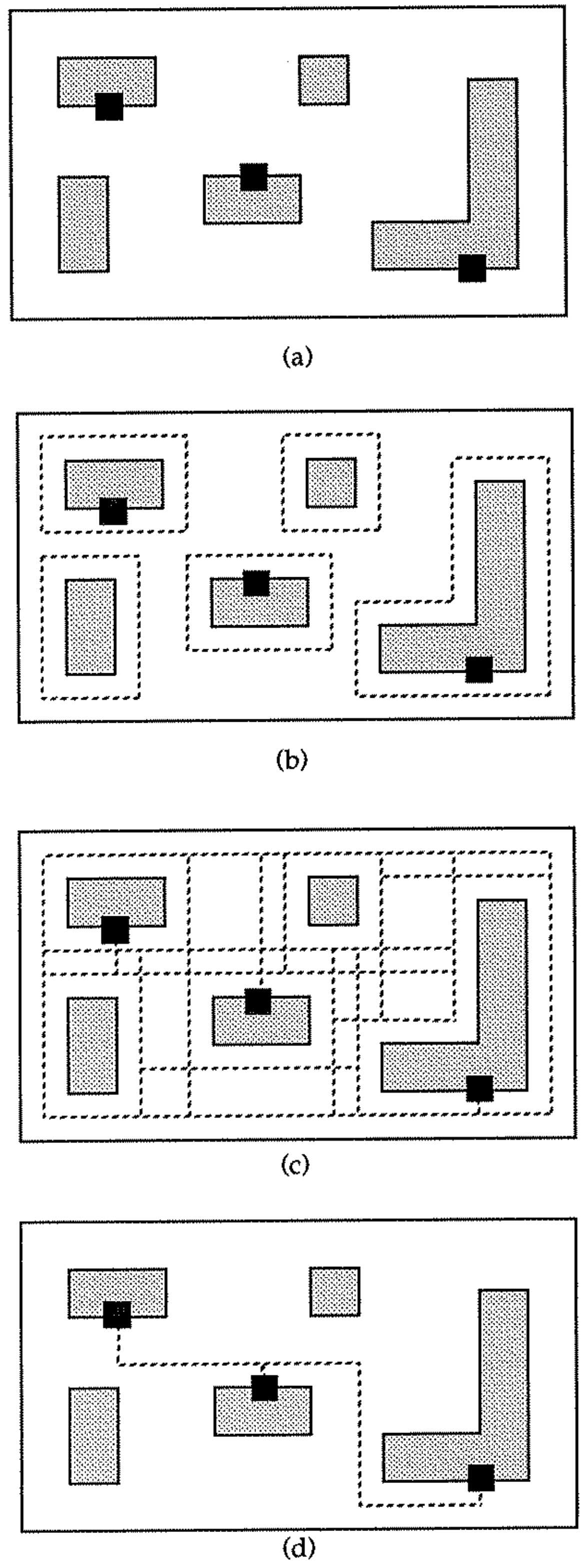


Figure 1.

obstacle's 'beltway.' For the instance depicted in Figure 1(a), these are the dashed segments of Figure 1(b). To allow for connections to the different obstacles, each beltway escape segment is continued to also form a 'highway' escape segment. A highway escape segment is a maximal segment with respect to the routing region (i.e., it ends with its abutment to either the external or internal perimeter of the routing region). Lastly, we introduce potential routing segments that extend from the terminals. These segments are also maximal in a manner similar to the highway escape segments. In our analogy, these segments correspond to the local roads that one uses to connect to the beltway and highway system. Thus, as stated above, the dashed segments shown in Figure 1(c) represent the set of escape segments for the instance depicted in Figure 1(a). It is possible to continue this analogy to use potholes to represent electrical shorts, but that is the subject of another paper.

We now proceed to show formally that there is an optimal routing for three- or four-terminal nets using only escape segments. We then show how to quickly generate an optimal interconnection by using standard routing techniques.

The proof will make use of the following theorem that was proven for the line intersection routing technique [3].

LIR Theorem:
If two terminals are routeable then there is an optimal routing using only escape segments and sub-segments.

III. Formal Multi-Terminal Proof

Suppose there exists a three- or four-terminal problem instance I for which there is no optimal solution that uses only escape segments. Let T be an optimal Steiner tree for instance I such that among all optimal solutions for I, T uses the minimal number of routing segments while maximizing its number of escape segments. We prove that no such instance can exist. Hence, every instance with three or four terminals can be routed using only escape segments.

Since we are concerned with rectilinear solutions, it is case that all vertices have degree four or less. In fact, we know there are at most two vertices with degree greater than two. This follows as each subtree rooted at a vertex in an optimal routing contains a terminal, and if there are three or more vertices with degree three or four then there are at least six terminals in the net.

Suppose T does not have a vertex of degree three or four. Then T is a simple path through all the terminals in I. However, by the LIR Theorem there exists an optimal routing between any pair of terminals that uses only escape segments. Thus, this path T can be converted in piecemeal fashion to one that uses only escape segments. Hence for I to exist, T must contain a vertex of degree three or four.

If the only vertices with degree three or four are terminals, then the above construction can be repeated. In fact, the same argument can be applied if such vertices lie at the intersection of escape segments. Thus, we only need to consider the case where T has at least one vertex of degree three or four that is a Steiner point (i.e., a non-terminal with degree three or four) such that the incident vertical or horizontal segments or both to that Steiner vertex do not lie on escape segments. We call such a vertex a *problem* vertex.

We next the consider the case where T has a single vertex v of degree three or four. Based on the above discussion regarding piecemeal application of the LIR theorem, we can assume that v is a problem vertex. Following this discussion, we consider the case where there are two vertices of degree three or four.

Since v is a problem vertex, there is a non-escape segment s incident to v. Since s is not an escape segment, neither of its endpoints are terminals. Hence, there is an orthogonal edge incident to s at its non-v endpoint. There is only one such edge as v is the only vertex with degree greater than two. Suppose v has degree four. Without loss of generality, assume s is an upward vertical segment incident to v. If s is a connected to a leftward incident edge l as in Figure 2(a), then it is possible to slide s to left and correspondingly shrink l without affecting feasibility. The result of such a sliding is depicted in Figure 2(b). Since the new tree contradicts T's optimality, it is instead the case that v has degree three. We next show that this case also cannot occur for v. Thus implying T has two vertices of degree three or four, at least one of which is a problem vertex.

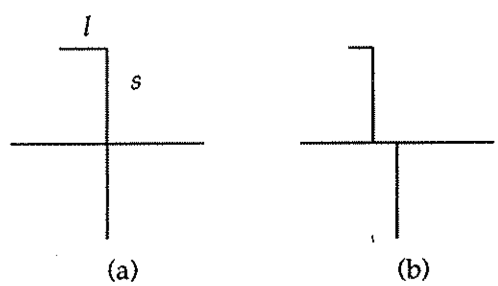


Figure 2. A sliding maneuver to reduce total wire length.

A degree-three vertex necessarily has two segments with the same orientation (horizontal or vertical) and one segment of the other orientation. We call a degree-three vertex, a T-vertex. The two segments incident to it with same orientation form the *head* and the other incident segment forms the *leg*. Without loss of generality, we assume that v's head segments l and u are vertical and that its leg segment s is a horizontal segment incident to v's right. The case is depicted in Figure 3.

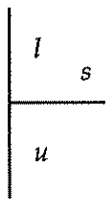


Figure 3.

Suppose leg s is not an escape segment. Necessarily, there is at least one segment incident to s's non-v endpoint. Since we are dealing with a case where there is a single vertex of degree greater than two, there is in fact exactly one such segment. Using an argument similar to the one applied for the case depicted in Figure 2, we can apply a sliding maneuver to contradict T's optimality. Hence, the leg is an escape segment.

Since vertex v is a problem vertex and its leg s is an escape segment, it is the case that head segments l and u are not escape segments. Given that v is the only vertex with degree greater than two, we can conclude that vertical head segments l and u are both incident respectively to single horizontal segments, a and b, at their non-v endpoints. Since l and u are non-escape segments, it is the case that their endpoints are not terminals. If either a or b are rightward incident from the head, then the previously discussed sliding maneuver is again applicable. Hence, we have the case depicted in Figure 4(a). For this case, we can simultaneously slide the head segments l and u leftward, increasing s's length but decreasing the lengths of a and b by the same amount. The result of this maneuver is depicted in Figure 4(b). Since the resulting tree would contradict T's optimality, the case cannot occur.

Thus, if there is to be a three- or four-terminal instance I without an optimal solution using only escape segments, then there are two vertices with degree greater than two. Using a terminal counting argument, we can conclude both that I is a four-terminal instance and that the two vertices in T with degree greater than two are Steiner T-vertices. Call the two

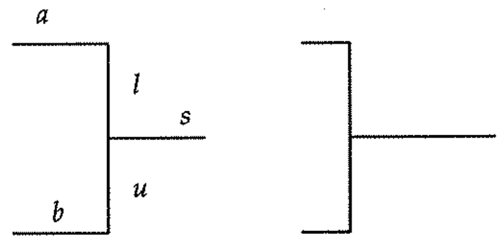


Figure 4. Another sliding maneuver to reduce total wire length.

Steiner vertices u and v. At least one of u and v is a problem vertex, or else a piecemeal reconstruction can be performed to produce an optimal solution that uses only escape segments. Without loss of generality, we assume v is a problem vertex with canonical T-vertex orientation (i.e., the head is horizontal and the vertical leg is incident from below v). Call the left and right head segments of v respectively l and r. Call v's leg segment s. The case is depicted in Figure 5.

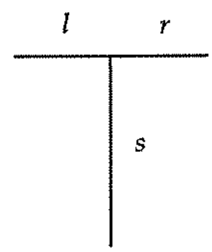
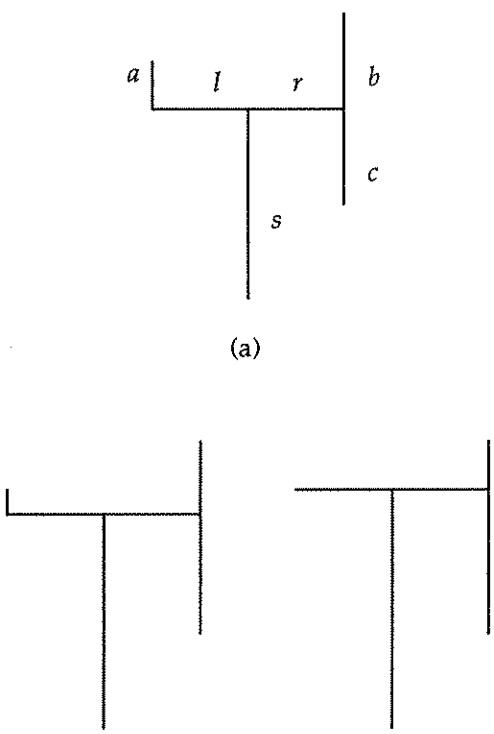


Figure 5.

Since T is connected there is a path between u an v. We first consider the case where the path between u and v does not use s. Without loss of generality, we assume that the path from v to u uses v's right head segment r. Based on a sliding maneuver argument, we can conclude that leg s is an escape segment. Since v is a problem vertex, head segments l and r are not escape segments and their endpoints are not terminals. Thus, we can conclude that segment l is incident to a single vertical segment. Call this vertical segment a. As not to contradict T's optimality, it is segment a's lower endpoint that is incident to l. A similar argument shows that there is a vertical segment b whose lower endpoint is incident to r. If head segment r is distinct from u's leg segment, then a is the only segment incident to r's endpoint. Thus, we would have a situation similar to Figure 4(a) and could apply a similar maneuver. Hence, it is instead the case that there exists a vertical segment c whose upper endpoint is incident to r and b.

Suppose segment a is shorter than segment b. We then have the case depicted in Figure 6(a). Since l and r are not escape segments and since a's lower endpoint is not a terminal, we can slide non-escape segments l and r upward and correspondingly increase and decrease respectively the lengths of s and a until either the shifted head segments are escape segments or the shifted head is incident to upper endpoint of a. The former case is depicted in Figure 6(b). Since this new solution is also optimal and uses more escape segments than T, it contradicts our assumption regarding T's nature with respect to the number of escape segments. Thus, it is instead the case that the head can be shifted so that it is incident to a's upper endpoint as in Figure 6(c). However, this solution is also optimal and uses less segments than T. Hence, it contradicts T's nature with respect to the number of segments in its solution. Thus, the case cannot apply. Since a similar argument can be made if segment b is shorter than segment a, it must instead be the case that the path from v to u uses leg s.

Thus, it remains to consider the case where the path between v and u uses leg s. We divide this case into two sub-cases—whether or not u and v share the same leg.



(a)

(b) (c)

Figure 6. Sliding maneuvers to affect the nature of an optimal solution.

Suppose the legs of u and v are distinct. If leg s is not an escape segment, then its lower endpoint is not a terminal and there is a single segment h incident to that endpoint. Without loss of generality, we assume that h is incident to the left side of s as in Figure 7. As a sliding maneuver can be applied here to

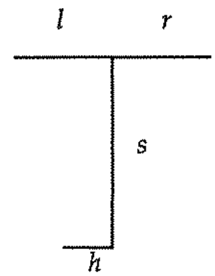


Figure 7.

contradict T's optimality, it is instead the case that s is an escape segment. As v is a problem vertex, segments l and r are not escape segments. This implies that the endpoints of l and r are not terminals. We can further conclude that there are single vertical segments a and b incident respectively to the non-v endpoints of l and r. As sliding maneuvers can be applied individually to a and b to produce a tree that contradicts T's optimality, it is not the case the legs of u and v are distinct.

Thus s is the leg of both u and v. Let a and b be the head segments of u. Hence, the case depicted in Figure 8 applies. Note, we do not assume that any of a, b, l, or r have the same length. Suppose s is an escape segment. Since v is a problem vertex, segments l and r are not escape segments. Using reasoning similar to the previous case where the two legs are distinct, we can conclude there must be single vertical segments c and d incident respectively to the non-v endpoints of l and r. Since a sliding maneuver is applicable that results in a tree that contradicts T's optimality no matter how the segments c and d are incident to the non-v endpoints, it is instead the case that s is not an escape segment.

As s is not an escape segment, it can be slid leftward until either the shifted leg is an escape segment or the shifted leg is incident to a left endpoint of a or l. Call the shifted leg c. If the former case applies where c is an escape segment, then the resulting tree is optimal and contradicts our assumption that T has a maximal number of escape segments.

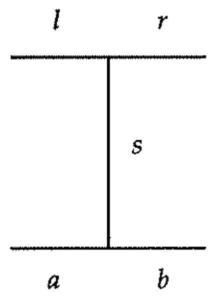


Figure 8.

Suppose only the latter case applies (i.e., c is incident to a left endpoint of a or l, and c is not an escape segment). Without loss of generality, assume that the shifted segment is incident to l's left endpoint. Since c is a not an escape segment, its common endpoint with l is not a terminal. Hence, in the original configuration there must be a single vertical segment d incident to l. It cannot be the case that d lies below l, as the segment would over overlap c. The removal of this overlap would not effect feasibility and hence would contradict T's optimality. Thus, segment d must lie above l. Since c is not an escape segment, d is not an escape segment. Since d is not an escape segment, its upper endpoint is not a terminal and there is a single horizontal segment e that is incident to d's upper endpoint. If e is incident to the left of d, then it is the case depicted in Figure 9. For this case a sliding maneuver can be applied to shift d rightward and correspondingly reduce the length of e. As the resulting tree contradicts T's optimality, it is instead the case that segment e is incident to the left of d. However, this case has been considered above—it is similar to the case depicted in Figure 6—and can be similarly processed.

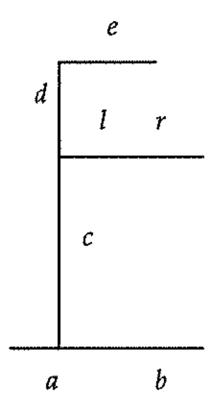


Figure 9.

This completes the case analysis. We have shown that there is no instance for which a three- or four-terminal net cannot be optimally solved using escape segments. ■

We next discuss TRIPLE and HOMER. These algorithms quickly construct optimal solutions for three- and four-terminal nets respectively, by effectively constructing and searching the set of escape segments.

IV. Algorithm TRIPLE

TRIPLE begins by constructing the escape segments associated with the obstacles (i.e., circuit elements and pre-laid wire). By using standard computational geometry line-sweeping techniques, the escape segments can be constructed in $O(n \log n)$ time, where n is the number of escape segments [2]. The intersections of these segments are then computed. Let m be the number of intersections. If there are n escape segments, then m can be at most $O(n^2)$. The extreme requires that all the horizontal escape segments intersect all the vertical escape segments. In practice, we believe m is proportional to n. An all-pairs shortest path algorithm is then applied to resulting escape segment graph. Since the graph is planar, the all-pairs path computation can be done in $O(m^2)$ time [7]. We then individually consider each of m intersection points in conjunction with the three terminals and evaluate in constant time for each set of four points, the length of the associated tree. We also consider in constant time the lengths of the three simple paths among the terminals. The overall minimum of these $O(m)$ calculations is an optimal routing solution. Since the all-pairs shortest path computation dominates the running time, TRIPLE has time complexity $O(m^2)$.

V. Algorithm HOMER

HOMER is similar to TRIPLE in that it first calculates the escape segments and their intersection points. HOMER then considers each of the $O(m^2)$ possibilities with two Steiner points, the m possibilities with one Steiner point, and the twelve possible simple paths. Each of these $O(m^2)$ possibilities can be processed in constant time. Hence, like TRIPLE, HOMER can run in $O(m^2)$ time.

VI. Implementation and Experimental Details

We find that the asymptotic analysis mirrors the time required in practice. We are currently using a naive implementation that is to be replaced with a sophisticated computational geometry approach. However, the running times even with a naive implementation are acceptable. For example, the time to construct the escape segments for a five-hundred logic cell example is on the order of a fraction of a second on a Sparc IPC. A naive all-pairs shortest path computation [6] requires on the order of two minutes. We expect to reduce this running time to several seconds when we finish the new implementation.

VII. Summary

We have shown that escape segments are a sufficient set of possible routing segments for determining an optimal solution for net instances with three or four terminals. We have also described our two algorithms, TRIPLE and HOMER, that exploit this result to quickly compute an optimal routing for such critical nets.

VIII. ACKNOWLEDGEMENTS

The authors' work have been supported in part through National Science Foundation grants MIP-9107717 and CDA-8922545 and Virginia CIT award 5-30971. Their support is greatly appreciated.

IX. References

[1] A. V. Aho, M. R. Garey, and F. K. Hwang, Rectilinear Steiner Trees: Efficient Special-Case Algorithms, *Networks*, vol. 7, 1977, pp. 37-58.

[2] J. L. Bentley and T. Ottmann, Algorithms for Reporting and Counting Geometric Intersections, *IEEE Transactions on Computers*, vol. C-28, no. 9, pp. 643-647, September 1979.

[3] J. P. Cohoon and D. S. Richards, Optimal Two-Terminal α-β Routing, *Integration: The VLSI Journal*, February 1988, pp. 37-57.

[4] J. P. Cohoon, J. S. Salowe, and D. S. Richards, An Optimal Steiner Tree Algorithm for a Net Whose Terminals Lie on the Perimeter of a Rectangle, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 4, April 1990, pp. 398-407.

[5] J. Cong, A. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong, Performance-Driven Global Routing for Cell Based IC's, *IEEE International Conference on Computer Design: VLSI in Computers & Processors*, Cambridge, MA, 1991, pp. 107-173.

[6] R. W. Floyd, Algorithm 97: Shortest Paths, *Communication of the ACM*, vol. 5, no. 6, 1962, pp. 345.

[7] G. N. Frederickson, Fast Algorithms for Shortest Paths in Planar Graphs with Applications, *SIAM Journal of Computing*, vol. 16, no. 6, December 1987, pp. 1004-1022.

[8] M. R. Garey and D. S. Johnson, The Rectilinear Steiner Tree Problem is NP-Complete, *SIAM Journal of Applied Mathematics*, vol. 32, 1977, pp. 826-859.

[9] D. W. Hightower, A Solution to the Line-Routing Problem on the Continuous Plane, *6th Design Automation Workshop*, Miami Beach, FL, 1969, pp. 1-24.

[10] J. JáJá and S. A. Wu, On routing Two-terminal Nets in the Presence of Obstacles, *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, vol. 8, no. 5, May 1989, pp. 563-570.

[11] C. Y. Lee, An Algorithm for Path Connections and Its Applications, *IRE Transactions on Electronic Computers*, vol. EC-10, no. 3, September 1961, pp. 346-365.

[12] E. F. Moore, Shortest Path Through a Maze, *Annals of the Computational Laboratory of Harvard University*, vol. 30, 1959, pp. 285-292.

[13] B. T. Preas and K. Roberts, *IEEE/ACM SIGDA Physical Design Workshop: Placement and Floorplanning*, Hilton Head, South Carolina, April 1987.