watermark's strength. We explain the basic approach and develop specific techniques for various classes of VLSI CAD problems. The new approach is compatible with all the existing watermarking techniques. With the help from organizations pushing for design standards, for example VSIA, this method has the potential of solving eventually the IP protection problem.

REFERENCES

[1] A. Adelsbach, B. Pfitzmann, and A. Sadeghi, "Proving ownership of digital content," in *Proc. 3rd Int Information Hiding Workshop*, Sept 1999, pp. 126–141.
[2] A. E. Caldwell, H. Choi, A. B. Kahng, S. Mantik, M. Potkonjak, G. Qu, and J. L. Wong, "Effective iterative techniques for fingerprinting design IP," *36th ACM/IEEE Design Automation Conf. Proc.*, pp. 843–848, June 1999.
[3] E. Charbon, "Hierarchical watermarking in IC design," in *Proc. IEEE 1998 Custom Integrated Circuits Conf.*, May 1998, pp. 295–298.
[4] E. Charbon and I. Torunoglu, "Copyright protection of designs based on multi source IP's," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1999, pp. 591–595.
[5] S. Craver, "Zero knowledge watermark detection," in *Proc. 3rd Int. Information Hiding Workshop*, Sept. 1999, pp. 102–115.
[6] F. Hartung and B. Girod, "Fast public-key watermarking of compressed video," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 1997, pp. 528–531.
[7] I. Hong and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," *Proc. 36th ACM/IEEE Design Automation Conf. Proc.*, pp. 849–854, June 1999.
[8] A. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe, "Watermarking techniques for intellectual property protection," *35th ACM/IEEE Design Automation Conf. Proc.*, pp. 776–781, June 1998.
[9] A. B. Kahng, D. Kirovski, S. Mantik, M. Potkonjak, and J. L. Wong, "Copy detection for intellectual property protection of VLSI design," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1999, pp. 600–604.
[10] D. Kirovski, Y. Hwang, M. Potkonjak, and J. Cong, "Intellectual property protection by watermarking combinational logic synthesis solutions," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1998, pp. 194–198.
[11] D. Kirovski, D. Liu, J. L. Wong, and M. Potkonjak, "Forensic engineering techniques for VLSI CAD tools," *37th ACM/IEEE Design Automation Conf. Proc.*, pp. 581–586, June 2000.
[12] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, "FPGA fingerprinting techniques for protecting intellectual property," *Proc. IEEE 1998 Custom Integrated Circuits Conf.*, pp. 299–302, May 1998.
[13] ——, "Signature hiding techniques for FPGA intellectual property protection," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1998, pp. 186–189.
[14] A. L. Oliveira, "Robust techniques for watermarking sequential circuit designs," *36th ACM/IEEE Design Automation Conf. Proc.*, pp. 837–842, June 1999.
[15] B. Pfitzmann, "Information hiding terminology," in *1st Int. Information Hiding Workshop*, May 1996, pp. 347–350.
[16] G. Qu, "Publicly detectable watermarking for intellectual property authentication in vlsi design," Univ. Maryland Inst. Advanced Computer Studies (UMIACS), UMIACS-TR-2002-17, 2002.
[17] G. Qu and M. Potkonjak, "Analysis of watermarking techniques for graph coloring problem," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1998, pp. 190–193.
[18] G. Qu, J. L. Wong, and M. Potkonjak, "Optimization-intensive watermarking techniques for decision problems," *36th ACM/IEEE Design Automation Conf. Proc.*, pp. 33–36, June 1999.
[19] R. L. Rivest. (1992) The MD5 Message-Digest Algorithm. [Online]. Available: http:// www.cis.ohio-state.edu/htbin/rfc/rfc1321.html
[20] N. A. Sherwani, *Algorithms for VLSI Physical Design Automation*, 3rd ed. New York: Kluwer, 1999.
[21] "Architecture Document Version 1.0," Virtual Socket Interface Alliance, 1997.
[22] "Intellectual property protection white paper: Schemes, alternatives and discussion Version 1.0," Virtual Socket Interface Alliance, 2000.
[23] "Virtual component identification physical tagging standard Version 1," Virtual Socket Interface Alliance, 2000.
[24] G. Wolfe, J. L. Wong, and M. Potkonjak, "Watermarking graph partitioning solutions," *38th ACM/IEEE Design Automation Conf. Proc.*, pp. 486–489, June 2001.
[25] [Online]. Available: ftp://ftp.sunet.se/pub3/vendor/sco/skunkware/uw7/fileutil/md5/src
[26] [Online]. Available: ftp://ftp.ox.ac.uk/pub/crypto/misc/rc4.tar.gz
[27] [Online]. Available: http://www.siidtech.com/
[28] [Online]. Available: http://dimacs.rutgers.edu/
[29] [Online]. Available: http://mat.gsia.cmu.edu/COLOR/instanc5pes.html

Preferred Direction Steiner Trees

Mehmet Can Yildiz and Patrick H. Madden

***Abstract*—The planar rectilinear Steiner tree problem has been extensively studied. The common formulation ignores circuit fabrication issues such as multiple routing layers, preferred routing directions, and vias between layers. In this paper, the authors extend a previously presented planar rectilinear Steiner tree heuristic to consider layer assignment, preferred routing direction restrictions, and via minimization. They use layer-specific routing costs, via costs, and have a minimum cost objective. Their approach combines the low computational complexity of modern geometry-based methods with much of the freedom enjoyed by graph-based methods. When routing costs mirror those of traditional planar rectilinear Steiner problems, the authors' approach obtains close to 11% reductions in tree lengths, compared to minimum spanning trees; this is on par with the performance of the best available Steiner heuristics. When via costs are significant and layer costs differ, they observe average cost reductions of as much as 37%. Their method can also reduce the number of vias significantly.**

***Index Terms*—Interconnect synthesis, optimization, routing, Steiner tree.**

I. INTRODUCTION

The Steiner problem has a long history in very large scale integration (VLSI) computer-aided design; global and detail routers use Steiner trees to define routes for signal nets, and many interconnect optimization approaches begin with Steiner constructions and then apply wire sizing, driver sizing, and buffer insertion to improve performance. Steiner tree research has focused on *planar rectilinear* formulations, which are appropriate for single layer routing and are

Manuscript received June 5, 2001; revised February 1, 2002. This work was supported in part by the National Science Foundation under Grant CCR-9988222. This paper was recommended by Associate Editor T. Yoshimura.

The authors are with the Computer Science Department, State University of New York, Binghamton, NY 13092 USA (e-mail: pmadden@binghamton.edu).

Digital Object Identifier 10.1109/TCAD.2002.804105

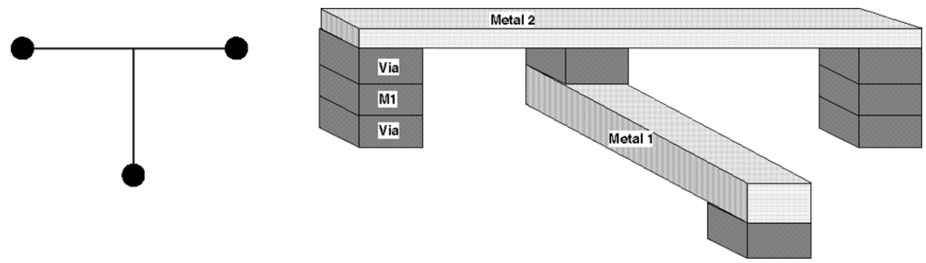

Fig. 1. Two-dimensional Steiner tree and three-dimensional implementation.

adequate when only two metal layers are available. While the Steiner problem is NP-Hard, there are a number of effective planar rectilinear Steiner heuristics.

The planar abstraction is imperfect, however, as is shown in Fig. 1. First, in modern designs we have an abundance of routing layers, and normally assign *preferred routing directions* to each; all connections on a given routing layer are either "horizontal" or "vertical." Vias must be inserted between layers; if we wish to minimize the number of vias, we face a layer assignment problem which is not considered in planar Steiner tree formulations. Second, most global routers attempt to minimize congestion by weighting routing resources differently; in order construct a *minimum cost* Steiner tree, our optimization objective cannot be simply length reduction, but a weighted sum of edge lengths and resource costs.

In this paper, we expand on earlier work [14] and consider *preferred direction multilayer Steiner trees*. This is a multilayer routing model, where each layer may restrict or penalize routing in a given direction. Steiner trees in this domain differ from the well studied planar (or n-Dimensional Manhattan) problem [3]; a Steiner tree which is optimal for a planar problem is not necessarily good quality when embedded into the multilayer routing framework.

Our approach, based on an earlier algorithm by Borah *et al.* [1], is computationally and memory efficient. The complexity of the approach is $O(n^2)$, efficient implementation is relatively easy, and tree lengths obtained are comparable to the best heuristics. Unlike planar heuristics, we directly and efficiently consider preferred direction routing, differing layer costs, and a variety of via costs. We note that graph-based Steiner heuristics can also handle the preferred direction routing model; in general, these methods have higher computational complexity.

II. Formulation and Previous Work

The Steiner tree problem is well studied, and a number of good surveys are available. Hwang and Richards [6] presented a study of general Steiner problems, while Kahng and Robins [8] focused on formulations and heuristics directly related to VLSI interconnect.

The general Steiner tree problem is to construct a tree of minimal cost or minimal length, spanning a set of *demand points* and possibly some additional *Steiner points*. Without Steiner points, the problem is simply one of constructing a minimum spanning tree (MST), and the algorithms of Kruskal [9] and Prim [11] are well known and computationally efficient. When Steiner points are considered, construction of a Steiner minimal tree (SMT) is NP-hard [2]. In this paper, *vertices* and *points* are used interchangeably; vertices are perhaps more appropriate when discussing three-dimensional or graph-based formulations, while points are more commonly used in planar formulations.

More formally, our objective is the following. Given a set of vertices $V = \{v_1, v_2, \ldots, v_n\}$, a set of optional Steiner vertices $S = \{s_1, s_2, \ldots, s_m\}$, and a set of edges E connecting the vertices, construct a minimum cost tree spanning V and a (possibly empty) subset of S. In most planar rectilinear Steiner research, it is assumed that length minimization is equivalent to cost minimization.

A. Planar Rectilinear Steiner Trees

For the planar rectilinear Steiner tree problem, a number of properties are known. Hanan [3] showed that possible Steiner points can be restricted to the *Hanan grid*, formed by the intersections of vertical and horizontal lines passing through the initial demand points. Hwang [5] showed that the ratio of MST to SMT cost cannot exceed 3 : 2.

Well-known heuristics for the Steiner problem include a "corner-flipping" method [4], and the 1-Steiner algorithm of Kahng and Robins [7]. The Iterated RV algorithm [10] of Mandoiu *et al.* utilizes a 3 : 2 approximation algorithm for quasibipartite graphs, obtains slightly better results than the 1-Steiner approach, and also enjoys somewhat reduced run times. Recently, Warme *et al.* have made their *GeoSteiner* [12] algorithm available. This algorithm is able to find *optimal* Steiner trees for surprisingly large planar Steiner problems. The running time for this algorithm is comparable to that of 1-Steiner. In our work, we use the algorithm of Borah *et al.* [1] and discuss this in more detail in the Section IV. While there have been a number of substantial advances for planar Steiner research, we note again that modern routing problems are not necessarily planar.

B. The Preferred Direction Rectilinear Steiner Problem

The routing model we focus on differs from traditional planar rectilinear formulations. We consider VLSI global and detail routing applications and directly consider congestion cost, preferred routing directions, and via costs. For the preferred direction routing problem, we are given a set of $L = \{l_1, l_2, \ldots, l_m\}$ *routing layers*. We are allowed to traverse from one layer to another with *vias* that have nonnegative cost. For each layer, we are allowed to make connections in only restricted directions, either "horizontal" or "vertical," and associate a *routing cost* with each direction. In the following, we will use $C_{l_i, H}$ and $C_{l_i, V}$ to indicate the routing cost in the horizontal and vertical directions on layer l_i and will refer to a portion of a tree edge on a layer which follows a particular direction as a *segment*. Our objective is to minimize the cost of a Steiner tree (in contrast to length minimization). We define tree cost to be the sum of via costs and segment costs. A horizontal segment of length D on layer i would have cost $D \times C_{l_i, H}$; each tree edge would contain vias and segments, and computing tree cost is straightforward.

To allow a preference in routing direction, we might assign a low cost to that direction on a layer. To prevent use of a particular direction, infinite cost could be assigned. If a layer is heavily congested, a global or detail router might assign a high routing cost to the available directions on the layer, resulting in a preference for routing on other layers. Our formulation can clearly handle *wrongway routing*, in which we allow connections that go against the preferred direction.

III. Routing Model Observations

In this section, we make a number of observations regarding our routing model. While our formulation allows some of the freedom available in a graph-based approach, we still enjoy a number of

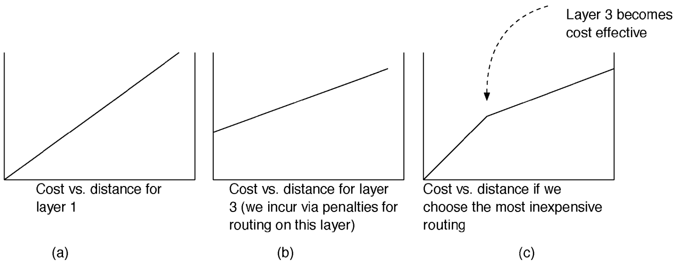

Fig. 2. Routing cost is not linear with distance: (a) uses the first metal layer, (b) uses the third metal layer, and (c) is minimum cost routing using either layer.

geometric properties that allow for the construction of an efficient Steiner tree heuristic. We summarize these observations and outline proofs for properties that are nonobvious.

First, we note that **routing cost is not linear with distance**. For example, if metal layers 1 and 3 have different costs, and vias have nonzero cost, the best layer to use to connect a pair of points may depend on the distance between the points. This is illustrated in Fig. 2. Related to this is the observation that if upper routing layers are not "less expensive," they will not be used. Using the upper layers incurs a via penalty, and the reduction in routing cost must outweigh this penalty if the layers are to be used.

Second, for connections of three demand vertices, **the optimal Steiner vertex location is *not* necessarily at the median of the coordinates of the three demand vertices**. This property may be somewhat surprising. In traditional planar rectilinear Steiner tree formulations, the optimal Steiner point for three vertices is *always* at the median of their coordinates. In Fig. 3, a simple example illustrates this point. In this figure, one point is on the first layer, two other points are on the third layer, and the first layer and vias have high cost; the routing in (c) is less expensive than either (a) or (b).

These differences may make us suspect that construction of a high-quality Steiner tree is difficult. Many current planar rectilinear approaches rely heavily on the *Hanan grid* to reduce the size of the solution space; nonlinear costs and preferred direction constraints may suggest that this grid no longer simplifies matters. Fortunately, we can prove **the Hanan grid still contains an optimal Steiner tree.** This is not affected by the layer costs, via costs, the number of layers, or preferred directions on each layer. Restricting our consideration to Hanan grid points does not bound our solution away from optimality.

We outline a proof of this property with the following. Assume we have an optimal Steiner tree, which includes Steiner point o. If this point is not aligned with the initial demand points, there are edges (using no more than two segments each) from o to other points in the optimal Steiner tree. If all layer assignments remain fixed, shifting o horizontally results in linear changes in tree cost—clearly, we can shift o in a direction which improves cost (contradicting the assumption that our tree is optimal), or we have no change in cost (indicating that we may shift the point until it falls onto or above the Hanan grid).

Finally, we note that for the three-dimensional model, the familiar **3 : 2 performance bound of MST to SMT does not hold**. If via costs are nonzero, a set of horizontally or vertically aligned points can result in a spanning tree with a large number of vias. A Steiner construction could eliminate close to half of the vias. If via costs dominate tree cost, the ratio can be arbitrarily close to 2 : 1. The ratio will not exceed 2:1; a proof known for general graphs can be applied to our formulation.

IV. Algorithmic Approach

Our approach is based on a geometric-based Steiner tree heuristic by Borah *et al.* [1]. We will first describe the construction of spanning trees under the routing model proposed and then show how elements

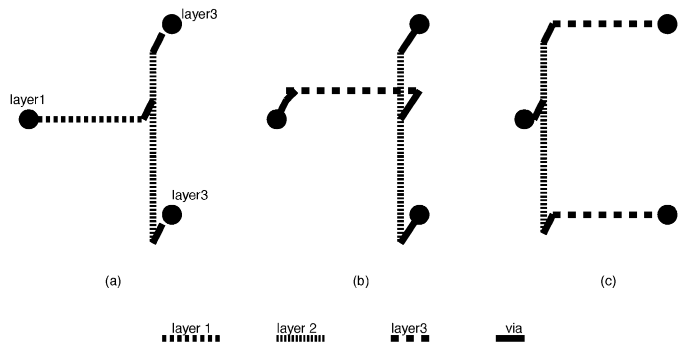

Fig. 3. An optimal Steiner point to connect three demand points may not be at the median of the point locations in the preferred direction model. In (a) and (b), we either utilize layer 1, or add additional vias. If layer 3 is inexpensive, (c) may be the most efficient method to connect the three points.

used by the spanning tree approach can be integrated into the Steiner tree heuristic.

A. Preferred Direction Spanning Trees

The spanning tree problem is well known and can be easily solved with either Prim's or Kruskal's algorithm. To address problems under the proposed routing model, we simply need a method to construct a minimum cost connection between a pair of vertices.

It should be obvious that *optimal* vertex-to-vertex connections contain only two segments. By enumerating two-segment connections using routing directions from each layer, and adding the appropriate vias required, we can find the cost of any spanning tree edge. As we have a small number of layers and routing vectors, finding the cost of any tree edge is constant time. If we have n vertices, there are at most $O(n^2)$ possible edges; by maintaining a heap, and inserting and removing edges as required, we can implement Prim's algorithm for the preferred direction routing model in $O(n^2)$. For a graph-based formulation, we would at the very least need to perform an all-pairs shortest path algorithm, with complexity $O(n^3)$; clearly, the geometric formulation would be substantially faster in practice.

B. Preferred Direction Steiner Trees

The planar rectilinear Steiner tree heuristic presented in [1] is an MST based heuristic, obtaining a Steiner tree through repeated modifications of an initial MST. Thus, it clearly classifies as an approximation algorithm, with a 3 : 2 performance bound. This heuristic is the basis for our work, and we show pseudocode for it in Fig. 1. We briefly describe a simplified version of the original heuristic and then show how it may be adapted to the preferred direction model. We also note a minor correction to the original heuristic.

The heuristic operates as follows. We first identify minimum cost pairings of edges and vertices; if we connect an edge to the vertex (at a suitable "merge" location), a cycle is created. We can break each cycle by removing a single edge. If the change in tree cost from connecting the vertex and edge pair, followed by the removal of the highest cost edge along the cycle, is beneficial, the modification has *positive gain*. The heuristic operates in a series of passes. In each pass, pairings are determined, high-cost edges on the cycles that might be generated are determined, and then tree modifications are implemented one at a time.

Construction of a list of candidate modifications is done in a simplified manner; we check all combinations of edges and vertices to find nearby pairings [$O(n^2)$, resulting in a list of $O(n)$ pairs], and then determine the highest cost edge on a cycle that would be generated by each pairing [$O(n)$ for each pairing, or $O(n^2)$ for the complete set]. The original algorithm uses a slightly more complex method to find the

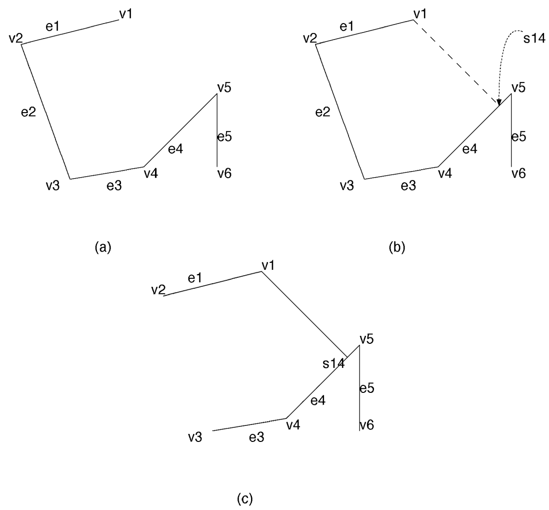

Fig. 4. The BOI algorithm.

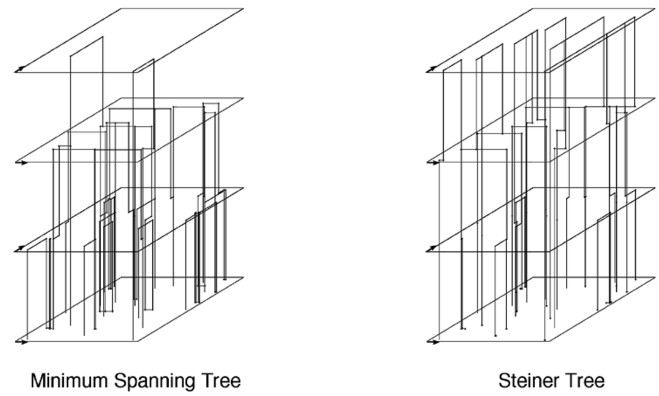

Fig. 5. A preferred direction Minimum Spanning Tree and the Steiner tree obtained by the our heuristic.

best gain combinations for all pairings of vertices and edges. Modifications are performed in a greedy manner, with the process repeating until no further improvement can be made.

Fig. 4, from [1], shows a sample modification for a planar Euclidean graph; the principle is the same for our preferred direction model. In this example, we might consider connecting edge e_4 to vertex v_1, as this is a nearby vertex that does not connect to e_4 directly. Making this connection introduces a cycle, with edge e_2 being the longest edge on the cycle. A possible tree modification would be the introduction of a new edge from v_1 to a point along e_4 and the removal of edge e_2; the *gain* for this modification would be the change in total tree cost.

Algorithm 1: A simplified variation on the Steiner tree heuristic of Borah *et al.* The initial MST is constructed with consideration of via and layer costs. The possible merge locations are taken from Hanan grid points and not simply the median of three points.

Find the MST T for the set of vertices.
repeat
  for all Edge $e_i \in T$ **do**
    Find vertex $v_j \in V$ that would connect to e_i with minimum cost (v_j is not an endpoint of e_i). We are attempting to locate suitable "merge" locations.
  end for
  for all Pairings of e_i to v_j **do**
    find the highest cost victim edge e_{victim} on the generated cycle
    Compute the gain if the modification is performed.
    If the gain is positive, store the candidate modification in a list.
  end for
  Sort the gain list in descending order
  for Each pairing in the gain list **do**
    Recompute the highest cost edge on the cycle
    If the gain is positive, modify the tree.
  end for
until no improvement is made to tree cost.

We have made a minor correction to the algorithm; in practice, it is possible for the original algorithm to create output that is disconnected. Our modification is the boldface line in the pseudocode listing. The original algorithm can fail to produce correct results if the following occurs. Assume that we have a pairing $\langle v_i,\, e_j \rangle$ with victim edge e_k, and a second pairing $\langle v_l,\, e_m \rangle$ with victim edge e_n. If e_k and e_n have identical cost, the cycles generated by both possible modifications intersect, unless we check the validity of a modification, we may obtain a disconnected graph (as both e_k and e_n might be removed). This modification does not increase the complexity of the algorithm; the authors of [1] were kind enough to provide us with their implementation, and the pseudocode above is in fact what had been implemented.

Aside from a modification to how individual edges are routed, and an enhancement in the manner that a vertex and edge are joined, the heuristic of [1] remains relatively unchanged. Our observations allow the identification of optimal Steiner points in the merge operation. By limiting candidate Steiner point locations to the multilayer Hanan grid, we maintain the low computational complexity of the original algorithm. Fig. 5 illustrates a Minimum Spanning Tree, and the Steiner tree obtained by the heuristic, for a preferred direction problem with four routing layers.

The complexity of our heuristic for the preferred direction Steiner tree problem is only $O(n^2)$, making it as fast as Minimum Spanning Tree algorithms for the routing model.

V. Experimental Results

To evaluate the performance of the preferred direction Steiner tree heuristic, we perform experiments on random point sets and compare SMT and MST costs; this is common practice in Steiner tree research. Points are randomly distributed on a 1000 by 1000 grid, with all points being located on the lowest layer. While we have performed many experiments with differing routing costs, we report only a subset here. In our experiments, routing directions being restricted to alternate horizontally and vertically. Via costs range from 1 to 200. To model the differences in routing capacity between layers, we either double cost with each layer (layer i from the "top" costs 2^{i-1} per unit) or increase by a factor of 1.1 (layer i from the "top" costs 1.1^{i-1} per unit). We considered nets with 3 to 30 pins, and 100 randomly generated problems for each combination. Note that these values are used to illustrate the flexibility of the algorithm; they are not meant to represent actual layer or via costs.

Results showing average percentage reduction in tree costs of SMTs over MSTs are in Fig. 6. When via costs are low, or differences between routing costs on each layer are modest, performance of the Steiner heuristic is comparable to other current planar methods. When via costs are high and the routing capacity of layers differs substantially, significant reductions in tree costs are obtained. While tree length reductions are on average only 11% for rectilinear planar formulations, we observe cost reductions of as much as 37% for the preferred direction model.

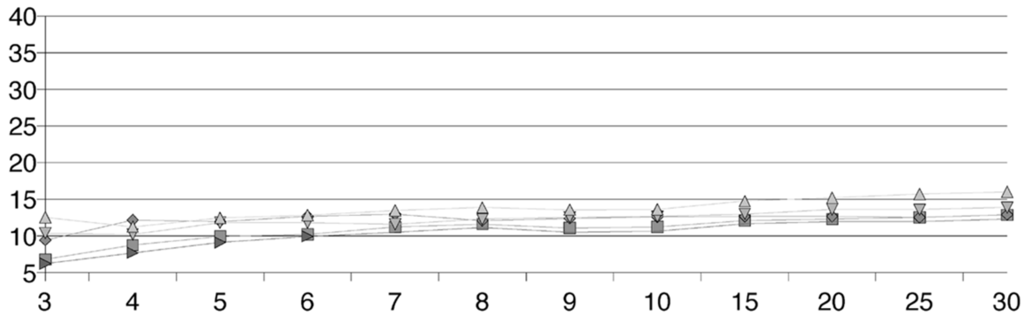

Fig. 6. Percentage improvement in tree cost over MST for nets of various sizes. We route using eight layers, with via costs (V) of 1, 50, 100, or 200, and routing layer cost factors (R) of 1.1 or 2. Experiments were performed with nets of 3 to 30 pins.

We have also performed studies to evaluate the number of vias required. We have observed via reductions ranging from 20% to as much as 43%, when compared to minimum spanning trees. In some respects, we can expect minimum spanning trees to have worst case performance, as insertion of Steiner points (on intermediate layers) is an excellent way to reduce the number of inter-layer connections.

VI. Summary and Conclusion

In this paper, a routing model which reflects current design constraints is considered. To our knowledge, this is the first work to explicitly consider a realistic three-dimensional routing model, while preserving the computational efficiency of geometric approaches.

In experiments with random point sets, tree cost reductions of as much as 37% on average were observed, and the number of vias required was reduced by as much as 43%. Mapping a planar result into multiple layers, or treating the problem as one on an arbitrary graph, is unlikely to be as efficient or effective.

Some principles from planar Steiner tree formulations no longer hold under this routing model; there are closely related principles, however, which cover many interesting aspects of the problem.

Tree cost minimization is clearly not the only objective for practical applications. We are also interested in performance optimization through wire sizing, buffer sizing, and buffer insertion. Our current work is on the integration of these techniques into our preferred direction Steiner tree heuristic, and in the use of this heuristic in practical VLSI global and detail routing tools.

Acknowledgment

The authors would like to thank the anonymous reviewers for their comments and suggestions; one reviewer pointed out a deviation in our algorithm from the original work of Borah *et al.*, and this was particularly helpful.

References

[1] M. Borah, R. M. Owens, and M. J. Irwin, “An edge-based heuristic for Steiner routing,” *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 1563–1568, Dec. 1994.

[2] M. R. Garey and D. S. Johnson, “The rectilinear steiner tree problem is NP-complete,” *SIAM J. Appl. Math*, vol. 32, pp. 826–834, 1977.

[3] M. Hanan, “Steiner’s problem with rectilinear distance,” *SIAM J. Appl. Math.*, vol. 14, pp. 255–265, 1966.

[4] J. M. Ho, G. Vijayan, and C. K. Wong, “New algorithms for the rectilinear Steiner tree problem,” *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 185–193, Feb. 1990.

[5] F. K. Hwang, “On Steiner minimal trees with rectilinear distance,” *SIAM J. Appl. Math.*, vol. 30, no. 1, pp. 104–114, Jan. 1976.

[6] F. K. Hwang and D. S. Richards, “Steiner tree problems,” *Networks*, vol. 22, no. 1, pp. 55–89, Jan. 1992.

[7] A. B. Kahng and G. Robins, “A new class of iterative Steiner tree heuristics with good performance,” *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 893–902, July 1992.

[8] ——, *On Optimal Interconnections for VLSI*. New York: Kluwer, 1994.

[9] J. B. Kruskal, “On the shortest spanning subtree of a graph,” in *Proc. Amer. Math. Soc.*, vol. 7, 1956, pp. 48–50.

[10] Ion I. Mandoiu, Vijay V. Vazirani, and Joseph L. Ganley, “A new heuristic for rectilinear steiner trees,” *IEEE Trans. Computer-Aided Design*, vol. 19, pp. 1129–1139, Oct. 2000.

[11] R. C. Prim, “Shortest connecting networks,” *Bell System Tech. J.*, vol. 31, pp. 1398–1401, 1957.

[12] D. M. Warme, P. Winter, and M. Zacharisen. The geosteiner 3.0 package. [Online]. Available: ftp.diku.dk/diku/users/martinez/geosteiner-3.0.tar.gz.

[13] ——, “Exact solutions to large-scale plane steiner tree problems,” in *Proc. 10th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA-99)*, 1999, pp. 979–980.

[14] M. C. Yildiz and P. H. Madden, “Preferred direction multi-layer steiner trees,” in *Proc. GLSVLSI2001*, 2001, pp. 56–61.