

# Preferred Direction Steiner Trees \*

Mehmet Can Yildiz    Patrick H. Madden  
State University of New York at Binghamton  
Computer Science Department  
pmadden@cs.binghamton.edu    http://vlsicad.cs.binghamton.edu

## ABSTRACT

Interconnect optimization for VLSI circuits has received wide attention. To model routing surfaces, multiple circuit layers are frequently abstracted as a single rectilinear plane, ignoring via costs, layer dependent routing costs, and congestion impact for routing in a particular direction. In this paper, we consider *preferred direction multi-layer routing*, which more closely models practical applications. We adapt a well known rectilinear planar Steiner tree heuristic, resulting in a new method to construct low cost Steiner trees under a realistic model. Our implementation is fast and effective, obtaining reductions in tree cost of 11% to 37% *on average* for random problems.

Our results include a proof that the performance bound of Minimum Spanning Tree cost to Steiner Minimal Tree cost under this model is 2:1 (in contrast to 1.5:1 for planar problems). We adapt the Hanan grid to this model, and show that a Steiner tree which would be optimal under a planar formulation is suboptimal for the multi-layer preferred direction model.

## 1. INTRODUCTION

Interconnect optimization has been an area of active research in VLSI physical design automation[5]. With the rapid scaling of feature sizes, interconnect now constitutes a substantial portion of system delay. Modern performance driven design cannot ignore the impact of interconnect wires.

Most interconnect optimization research focuses on a *rectilinear planar* abstraction of the routing surface. Issues such as via counts, preferred direction routing, and even layer dependent design rules are frequently ignored. In this paper, we consider a *preferred direction multi-layer* routing model, and develop an effective Steiner tree approximation algorithm for it. We also present a number of results relevant to this routing model.

Our formulation is intended for use in global and detail routing, where layer assignment and via counts have a large impact on so-

---

\*This research was supported by NSF under grant CCR-9988222.

lution quality. Recent global routers (such as MCG[2], SEGRA[3], or DECIMATE[4]) decompose multi-pin nets into two-pin edges, and then find routings for each edge. The method used to perform this decomposition will have a significant impact on congestion and routability.

There are significant differences between the rectilinear planar Steiner tree formulation, and the model which we present here. In our formulation, we can consider the cost and location of vias, restriction of routing on a layer to a given direction, and layer-dependent routing costs. This flexibility allows our global and detail routing tools to construct topologies appropriate for the available routing resources. We note that many of the results obtained for the planar Steiner tree problem do not hold for the preferred direction multi-layer model.

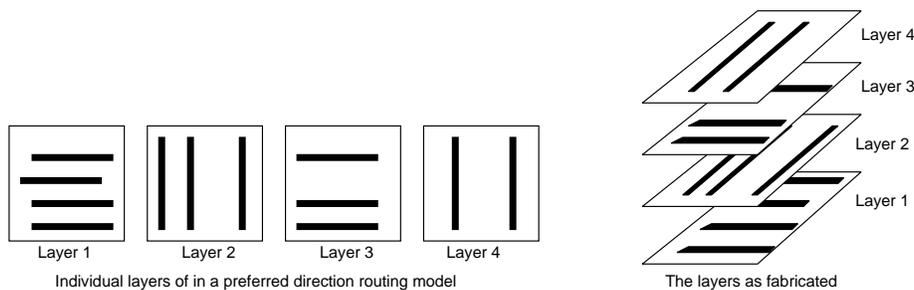
The remainder of this paper is organized as follows. We first discuss our preferred direction multi-layer routing model in detail. Next, we briefly summarize previous planar Steiner tree algorithms, as they form the basis of many current interconnect optimization approaches. A number of lemmas and theorems relevant to our routing model are presented in Section 5, followed by summary of our algorithmic approach. We conclude this paper with a set of experimental results, and a summary of the work presented.

## 2. ROUTING MODEL

In modern design, circuit interconnect can be placed on a number of different routing *layers*. Current fabrication processes may have as many as eight layers, with four layers being common. Multichip modules and printed circuit boards may also have large numbers of routing layers. Each layer has a *preferred* direction, usually alternating horizontally and vertically. Vias are used to connect adjacent layers, and may have varying cost in terms of electrical resistance, routing area required, and reliability.

Routing cost on each layer is dependent on the material used; for ICs, electrical resistance on lower metal layers is quite different from that of an upper layer, and each layer may have a unique minimum feature size and spacing. Some layers may be heavily *congested*: routing space may be limited, and a global router may wish to weight individual layers differently, resulting in a preference for Steiner tree topologies which use less congested areas.

Under this model, we are interested in obtaining a *minimum cost* Steiner tree, as compared to *minimum length*. An interconnect tree which uses only the first few metal layers may have low length; a higher length tree which utilizes low-resistance upper layers (or less congested layers) may be much more desirable. We utilize



**Figure 1: Circuits are generally implemented using multiple routing layers, with each layer having a preferred routing direction (and possibly also varying width and spacing requirements). Wrong-way routing, while possible, severely reduces routing capacity.**

different costs for each routing layer to allow tree construction to adjust to match the resources available.

This routing model is illustrated in Figure 1. If, for example, we wish to connect two points on the lowest layer at coordinates  $(5, 7)$  and  $(9, 20)$ , we would first have a horizontal segment 4 units long on the first layer, a via to the second layer, a vertical segment 13 units long, and then another via connecting back to the first layer. The *cost* of this connection is not necessarily equal to its length: the vias themselves might incur routing cost, and the congestion on the two layers might differ substantially.

We model cost by associating a constant *cost factor* with routing in a particular direction on each layer, and also a fixed cost for a via between a pair of layers. In our formulation, a single layer might have two possible routing directions, with differing *cost factors*: we can model “wrong-way” routing easily, penalizing segments using the non-preferred direction heavily. Steiner tree cost is the sum of costs for each edge; each edge is simply the sum of via costs and the lengths of each segment multiplied by the appropriate *cost factor*.

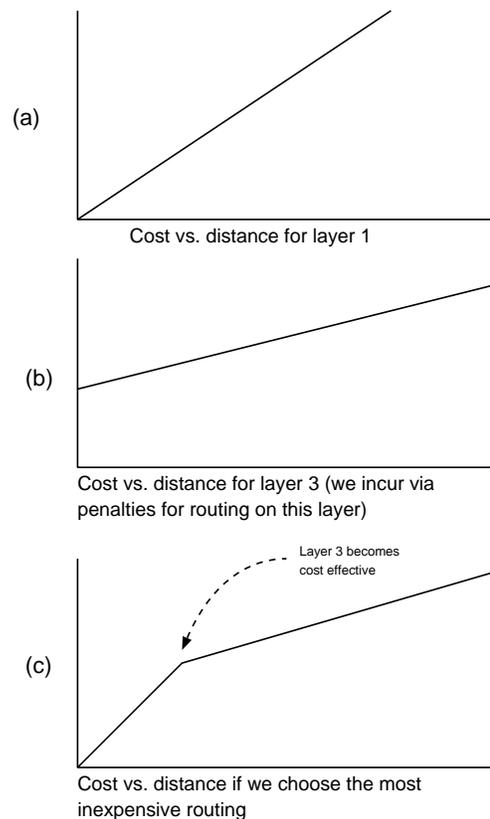
We assume that routing cost for a particular direction is uniform across an entire layer; this simplification allows the development of an efficient geometry-based algorithm. A graph based Steiner tree algorithm could also support this routing model, but would have substantially higher computational complexity.

The use of differing routing costs on each layer leads to an important observation, which we illustrate with a brief example; Figure 2 shows this example graphically. If we have a horizontal edge of length 10, via costs of 15 each, and routing cost factors of 2 and 1 for the first and third layers (assuming the direction of the edge is compatible with both layers), the cost for routing on the first layer is  $10 \times 2 = 20$ , while the cost of routing on second layer is  $10 \times 1 + 15 + 15 + 15 + 15 = 70$  (we have two vias at each end of the edge). Clearly, an edge of length 10 is best routed on the first layer. If the length of the edge is 100, then the two routings cost  $100 \times 2 = 200$  and  $100 \times 1 + 15 + 15 + 15 + 15 = 160$ ; for the longer edge, we would prefer routing on the second layer.

### 3. PREVIOUS WORK

The routing model described in the previous section has a number of obvious contrasts to the more common rectilinear planar formulation. As the bulk of the work relevant to VLSI CAD utilizes rectilinear planar formulations, we discuss this briefly here.

The general Steiner tree problem is as follows. We are given a set of *demand points* which must be connected by tree edges. We are al-



**Figure 2: The cost of connecting a pair of locations is dependent on distance, and the layers used to connect. In some cases, it may be more efficient to insert additional vias in order to utilize a routing layer with lower cost. In general, routing cost is not linear with distance.**

lowed to insert of additional *Steiner points*. An optimal Steiner tree is a tree which minimizes total cost, using as many Steiner points as necessary. If we construct a minimum cost tree over only the demand points, this is a Minimum Spanning Tree (MST); algorithms to compute this optimally are well known[9, 10]. The Steiner Minimal Tree (SMT) problem is well known to be NP-hard.

For the rectilinear planar Steiner tree problem, possible Steiner points can be restricted to the *Hanan grid*[6], formed by the intersections of vertical and horizontal lines passing through the initial demand points. The ratio of MST to SMT cost cannot exceed 1.5:1[7]. While the rectilinear planar SMT problem is quite difficult, a number of effective heuristics have been proposed. As the ratio of MST to SMT cost is bounded, many algorithms use the MST as a starting point.

The 1-Steiner algorithm[8] constructs an SMT by evaluating all possible Steiner points for their impact on MST cost. The algorithm operates in a series of passes; in each pass, the single Steiner point which provides the greatest improvement in spanning tree cost is selected and added to the set of demand points. Points are added until no further improvement can be obtained. This algorithm is extremely effective, although efficient implementation can be challenging. In experiments with random point sets, this algorithm obtains nearly 11% improvements over MST on average.

The heuristic that is adapted to the preferred direction routing model is the “BOI” or “edge-removal technique” of Borah, Owen, and Irwin [1]. In this approach, we repeatedly determine an edge and vertex pair in an MST that are close to each other. If we generate a connection from the vertex to some location on the edge, we introduce a cycle in our tree; this cycle can be eliminated by removal of a second edge along the cycle. The BOI approach determines a set of these pairings, along with appropriate “victim” edges that are to be removed; new connections are inserted and victim edges are removed until no improvements can be obtained. The approach has low complexity, with performance comparable to that of 1-Steiner. This approach is discussed in greater detail in the next section.

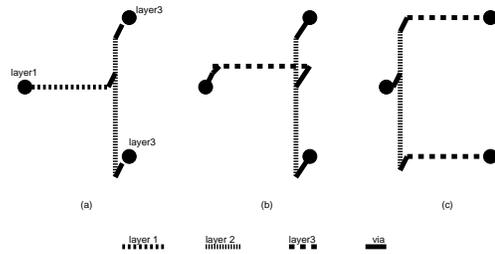
## 4. PREFERRED DIRECTION STEINER TREES

In this section, we present a number of observations relevant to optimal Steiner trees under the preferred direction multi-layer routing model, followed by a description of our algorithmic approach.

### 4.1 Geometric Properties

A number of observations can be made regarding Steiner trees under this model.

- Routing cost is not linear with distance. If layer  $l_i$  has a lower per-unit cost than layer  $l_j$ , it may be beneficial to insert additional vias (which have costs themselves) to take advantage of the lower cost. For short distances, the via cost may outweigh the savings of an alternate layer. In contrast to planar rectilinear formulations, cost under this model is monotonically increasing, but not necessarily linear with distance.
- In the multi-layer model, 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.



**Figure 3: An optimal Steiner point to connect three demand point 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.**

- For the three dimensional model, the familiar 1.5:1 performance bound of MST to SMT does not hold.

We now present a few results, and short proofs for the properties which are not obvious.

**Lemma 1: All tree edges will consist of at most two segments.**

We assume that the distance between the two points connected is  $\Delta X + \Delta Y$ . If we use two different layers to span the  $\Delta X$  distance, we pay the routing costs for each multiplied by the distance traveled. This sum is  $\Delta X_i \times \text{cost}(l_i) + \Delta X_j \times \text{cost}(l_j)$ . If  $\text{cost}(l_i) \geq \text{cost}(l_j)$ , we could clearly improve the connection by utilizing only  $l_j$ .

**Lemma 2: An optimal Steiner point which joins three vertices is not necessarily at the median of their X and Y coordinates.**

The difference in cost on the various layers, and the cost of additional vias, change the location of an optimal Steiner point. Consider Figure 3; if 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) may be 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 rectilinear approaches rely heavily on the *Hanan grid* to reduce the size of the solution space; non-linear costs and preferred direction constraints may suggest that this grid no longer simplifies matters. Fortunately, we can prove that optimal Steiner points still lie on the *Hanan grid*, passing through each routing layer; this is not affected by the layer cost, via cost, number of layers, or direction of each layer.

**Theorem 1: Optimal Steiner points for the preferred direction model lie on or directly above the Hanan grid points.**

*Proof:* 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).

**Lemma 3: The triangle inequality holds for the preferred direction routing model.** If we have three vertices  $v_a, v_b, v_c$ , the sum of costs for edges from  $v_a$  to  $v_b$  and  $v_b$  to  $v_c$  is greater than or

equal to the cost of an edge directly from  $v_a$  to  $v_c$ .

**Theorem 2: The ratio of costs of a Minimum Spanning Tree to that of a Steiner Minimal Tree is no greater than 2:1 for the preferred direction multi-layer routing model.**

*Proof:* We utilize methods found in general graph theory for our proof. We assume we have an optimal Steiner tree  $T$ , as shown in Figure 4(a). We can construct a tour of all vertices with a depth-first search starting from any leaf vertex, as shown in Figure 4(b). The cost of this tour is clearly twice the cost of the optimal Steiner tree. If we eliminate intermediate Steiner points from this tour as shown in Figure 4(c), we reduce the cost of the tour: our third lemma ensures that a direct edge has cost no greater than that of an edge which includes an intermediate point. The cost of a Minimum Spanning Tree is clearly less than the cost of a tour which visits each demand point; thus, the ratio of MST to SMT costs is no greater than 2:1.

It is a simple matter to construct a routing problem in which the MST to SMT ratio comes arbitrarily close to the 2:1 bound. If we place demand points in a horizontal line, and only allow horizontal connections on an upper layer, each edge of a Minimum Spanning Tree will require two vias; an optimal SMT would have only one via per demand point. If the demand points are close together, via cost will dominate tree cost, and the ratio will approach 2:1.

**4.2 Algorithmic Approach**

Our approach to the preferred direction multi-layer Steiner tree problem is based on the rectilinear planar Steiner tree heuristic of [1]. We discuss their approach in more detail, and then describe how it is adapted to the new routing model.

The approach of [1] constructs a Steiner tree through repeated modification of an initial Minimum Spanning Tree. The modifications performed are as follows. We first identify a pairing of an edge to a nearby vertex; if we connect this edge to the vertex (at a suitable point), a cycle is created. We can break this cycle by removing a single edge. If the change in tree length from connecting the vertex and edge pair, followed by the removal of the edge along the cycle, is beneficial, the modification has *positive gain*.

Construction of a list of candidate modifications is a simple matter; we check all combinations of edges and vertices to find nearby pairings, and then determine the longest edge on any generated cycle. Modifications are performed in a greedy manner, with the process repeating until no further improvement can be made. Figure 6 from [1] shows a sample modification. Pseudo code for the algorithm is given in Figure 5.

The structure of the tree changes with each modification, so we verify that a modification is appropriate before it is performed. Without these checks, it is possible that the tree may become disconnected, or that we delete a suboptimal element on the cycle. The complexity of finding possible pairings is  $O(n^2)$ , as is the cost of implementing all modifications found in a pass. The number of passes required is generally small, resulting in a complexity of  $O(n^2)$  for the entire algorithm. Our implementation has been used to construct Steiner trees for problems with tens of thousands of points.

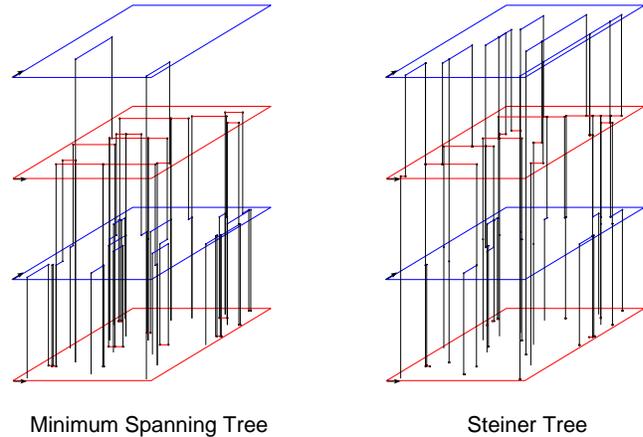
The Steiner heuristic described above was intended for planar problems. We adapt the heuristic to the preferred direction model (only

```

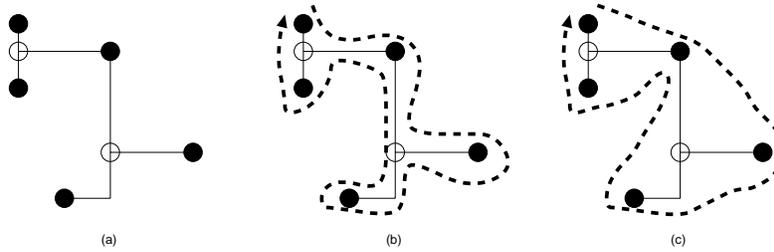
Begin
- Find the MST for the set of vertices.
- For each < vertexi, edgej > pair of the spanning tree,
  find the optimum Steiner point to merge the endpoints
  of edgej with vertexi
  find the longest edge on the generated cycle
  compute the gain if the modification is performed
  store the pair and the gain in the gain list
  if the gain is positive
- Sort the gain list in descending order
- while (there exists a pair in the list)
  recompute the longest edge on the cycle and the gain
  if (the edges to be replaced are in the tree
  and the gain is positive)
    modify the tree
  end if
end while
end.

```

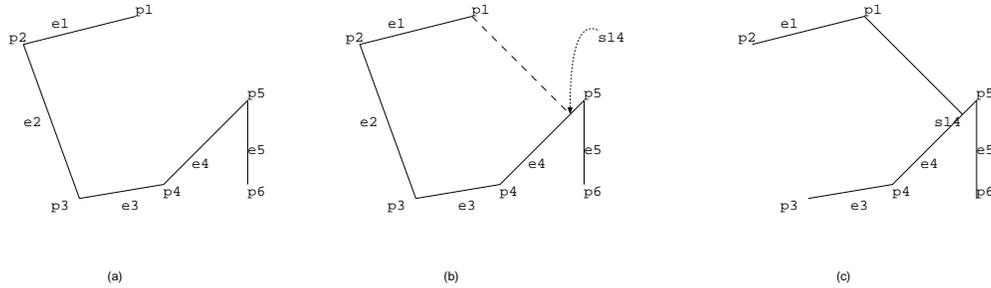
**Figure 5: A variation on the Steiner tree heuristic of Borah, Owen, and Irwin. We modify the approach to consider a variety of candidate Steiner points, and also recompute the longest edge on a generated cycle prior to performing a modification.**



**Figure 7: A preferred direction Minimum Spanning Tree, and the Steiner tree obtained by our approximation algorithm. Demand points are located on the bottom layer; via costs make the upper layers relatively expensive to use, resulting in heavy congestion on the lower layers. The Steiner tree introduces Steiner points on all layers, resulting in lower tree cost, and more uniform usage of routing resources.**



**Figure 4: The ratio between MST and SMT cost is at most 2:1 for the preferred direction model. Beginning with an optimal SMT, we can construct a *tour* of all points as shown in (a). Eliminating Steiner points from this tour reduces the cost of a tour, as shown in (b). The cost of this reduced tour must be greater than the cost of a Minimum Spanning Tree.**



**Figure 6: The BOI approach modifies an initial spanning tree by determining a pairing of an edge and nearby vertex. If we connect this edge to the vertex (inserting a Steiner point), we generate a cycle; removal of the longest edge on this cycle can result in a reduction in overall tree length or cost.**

relatively minor changes are required). The most significant modification is in the estimation of edge cost: we can determine the lowest cost routing between a pair of points by simply enumerating the different possible layer assignments for a pair of segments connecting the points (and including the necessary vias). We consider all possible combinations when determining the routing layers and directions (note that we cannot, in general, perform “corner flipping” without incurring additional via cost).

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. First, we construct a minimum spanning tree using Prim’s algorithm and the appropriate edge cost method. We then perform gain computations for each  $\langle vertex_i, edge_j \rangle$  pair. As the optimal Steiner point to merge an edge and vertex is not necessarily the median of the coordinates, we use a different approach: we check all Hanan grid points on each layer generated by the endpoints of the edge and the vertex (we are assured that one of these locations will be optimal). For three points, there are 9 possible Hanan grid locations; if we have  $l$  layers,  $l \times 9$  possible Steiner points are considered. The Steiner point which provides the minimum total cost (the sum of edge costs from the Steiner point to the three initial points) is selected. We can then determine the *gain* for this pair by identifying the longest edge on the cycle created if we connect  $vertex_i$  to  $edge_j$ . For example, in Figure 6, the gain of  $\langle p_1, e_5 \rangle$  is

$$gain_{15} = cost_2 + cost_5 - cost_{s1} - cost_{s5} - cost_{s6}$$

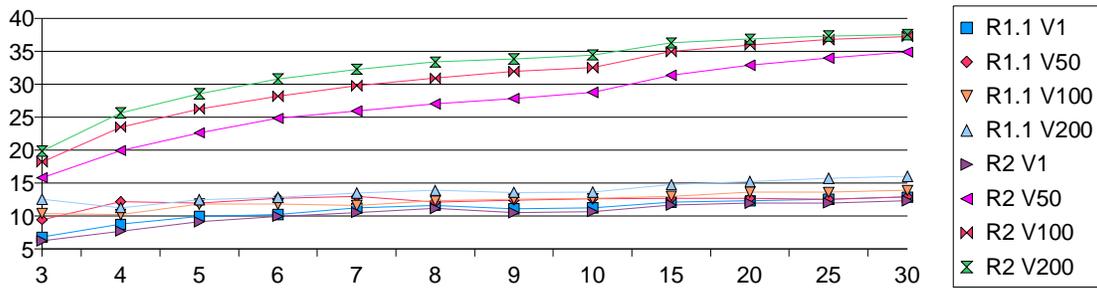
The theorem and lemmas presented allow the identification of optimal Steiner points in the merge operation. By limiting candidate

Steiner point locations to the multi-layer Hanan grid, we maintain the low computational complexity of the original algorithm; our implementation has been applied to problems with thousands of points. As the resulting Steiner Tree is obtained through improvements to an initial Minimum Spanning Tree, the resulting tree cost is less than twice that of an optimal Steiner tree, and our approach could be classified as an approximation algorithm. Figure 7 illustrates a Minimum Spanning Tree, and the Steiner tree obtained by our method, for a preferred direction problem with four routing layers.

As was mentioned in the introduction, we are interested in this formulation as it relevant to global routing. We integrate our Steiner tree algorithm into our global router in the following way. Initial layer and via costs are selected, and Steiner trees for all nets are generated. We consider routing resource utilization, and adjust via and layer costs if any resource is over utilized. Repeating this process, routing resource utilization converges to a uniform distribution.

## 5. EXPERIMENTAL RESULTS

To evaluate the performance of the Preferred Direction Steiner Tree algorithm, we perform experiments on random point sets, and compare SMT and MST costs. Points are randomly distributed on a 1000 by 1000 grid, with all points being located on the lowest layer. In this paper, we present results with 8 routing layers, with 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



**Figure 8: Percentage improvement in tree cost over MST for nets of various sizes. We route using 8 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.**

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 Figure 8. When via costs are low, or differences between routing costs on each layer are modest, performance of the Steiner heuristic is comparable to that of 1-Steiner or BOI. When via costs are high and the routing capacity of layers differs substantially, significant reductions in tree costs are obtained. While average rectilinear planar Steiner tree algorithms obtain only 11% improvements, we observe *average* reductions of as much as 37%.

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.

## 6. SUMMARY AND CONCLUSIONS

In this paper, a routing model which reflects current design constraints is presented. We directly consider preferred routing directions and via costs, while maintaining a computationally efficient approach. In experiments with random point sets, tree cost reductions of as much as 37% on average were observed. On problems with low via cost, our results are comparable to the best available Steiner tree heuristics. Previous methods might either map a planar result into multiple layers (degrading solution quality), or approach the problem with a graph-based formulation (losing computational efficiency).

We have shown that some results for planar Steiner tree formulations do not hold for our routing model. In particular, the ratio of minimum spanning tree to Steiner minimal tree cost is 2:1, and not 1.5:1. While edge cost is not linear with distance, we show that a construction quite similar to the Hanan grid contains all necessary Steiner points.

Our focus has been on *minimum cost* Steiner trees; we are currently integrating this routing model with performance driven interconnect optimization methods, where we consider wire sizing, buffer sizing, and buffer insertion.

We are also directly applying this Steiner tree approach to performance driven multi-layer VLSI global routing. By adjusting routing costs on each layer, we can balance tree lengths, layer usage, and via counts, reducing global routing congestion substantially

compared to previous methods.

## 7. REFERENCES

- [1] M. Borah, R. M. Owens, and M. J. Irwin. An edge-based heuristic for Steiner routing. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 13(12):1563–1568, December 1994.
- [2] Jo Dale Carothers and Donghui Li. The MCG autorouter for multichip modules. *IEEE Trans. on Circuits and Systems*, 46(5):572–578, May 1999.
- [3] Young-Jun Cha, Chong S. Rim, and Kazuo Nakajima. Segra: A very fast general area router for multichip modules. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 18(5):659–666, May 1999.
- [4] J. Cong and P. H. Madden. High performance global routing for standard cell design. In *Proc. Int. Symp. on Physical Design*, pages 73–80, 1997.
- [5] Jason Cong, Lei He, Cheng-Kok Koh, and Patrick H. Madden. Performance optimization of VLSI interconnect layout. *Integration, the VLSI Journal*, 21:1–94, 1996.
- [6] M. Hanan. On Steiner’s problem with rectilinear distance. *SIAM Journal of Applied Mathematics*, 14:255–265, 1966.
- [7] F. K. Hwang. On Steiner minimal trees with rectilinear distance. *SIAM Journal of Applied Mathematics*, 30(1):104–114, January 1976.
- [8] A. B. Kahng and G. Robins. A new class of iterative Steiner tree heuristics with good performance. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 11(7):893–902, July 1992.
- [9] J. B. Kruskal. On the shortest spanning subtree of a graph. *Proc. American Math Society*, 7:48–50, 1956.
- [10] R. C. Prim. Shortest connecting networks. *Bell System Technical Journal*, 31:1398–1401, 1957.