A Steiner Tree Construction for VLSI Routing

Andrew B. Kahng

UCLA CS Dept., Los Angeles, CA 90024-1596
kahng@cs.ucla.edu

Abstract

Construction of a minimum rectilinear Steiner tree (MRST) is a fundamental problem in VLSI circuit design, where we wish to connect the pins of a signal net using minimum wirelength. The MRST problem is NP-complete, and well over a hundred papers in the combinatorial optimization and VLSI CAD literatures discuss various heuristics. In this paper, we propose a new parallel, "analog" approach which intuitively shrinks a bubble around the pins of the signal net until a Steiner tree topology is induced. This method maps well to parallel neural-style architectures, as well as to fairly generic two-dimensional array topologies. We describe the basic approach along with extensive preliminary simulation results which show better performance than existing MRST approaches (i.e., almost 10% average improvement over minimum spanning tree cost). Thus, the result is of practical interest for VLSI CAD applications, and is a rare instance where an "analog" heuristic for an NP-complete problem outperforms existing combinatorial methods, both in time complexity and average-case performance.

1 Introduction

The *Minimum Rectilinear Steiner Tree* (MRST) problem in the plane is as follows: Given a set P of n points, find a set S of *Steiner points* such that the minimum spanning tree over $\{P \cup S\}$ has minimum cost. The cost of any edge in the tree is the rectilinear, or Manhattan, distance between its endpoints, and the cost of a tree is the sum of its edge costs. This is a fundamental problem in global routing and wire estimation for VLSI circuit layout, where we are interested in the Steiner trees connecting the terminals P of a signal net. The MRST problem is NP-complete [5], and over the years a number of heuristics have been developed. Because one typically constructs a global routing solution by calculating Steiner trees over all signal nets, then uses the result to iteratively improve the module placement, efficiency in the MRST approximation is important. However, accuracy is also an issue; even small improvements in Steiner tree heuristics have significantly improved global routing solutions, with attendant wirelength savings and reduced layout congestion. Furthermore, accurate Steiner tree computation is essential to predicting layout area and wireability for high-level synthesis and floorplanning. A detailed discussion of these VLSI CAD issues may be found, e.g., in [13].

A large body of work in the MRST literature involves refining an initial *minimum spanning tree* (MST) topology over P to yield a heuristic Steiner tree (see Figure 1). Because Hwang [9] showed that the worst-case *performance ratio* of MST cost to MRST cost is $\frac{3}{2}$, such *MST-based* methods are theoretically attractive. The most recent work is by Ho, Vijayan and Wong [7], who give a method for finding the *optimal* Steiner tree lying in the union of bounding boxes of MST edges. Other methods are surveyed, e.g., by Richards [13].

The standard testbed for MRST heuristics is simply pointsets randomly chosen from a uniform distribution in the unit square; this reflects the statistical properties of placed netlists. For n points selected randomly from a uniform distribution in the unit square, the minimum spanning tree (MST) cost and the MRST cost have the same growth rates, and thus MRST heuristics are usually evaluated by their average cost improvement over the MST. Virtually all algorithms average between 7% to 9% improvement over MST cost for random instances. The optimal MST-based method of [7] averages just over 9% improvement on typical problem sizes. Current MRST approximations can be impractical: geometric approximation

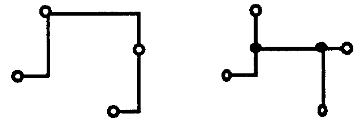

Figure 1: The minimum spanning tree (MST) on four points is shown at left. The introduction of two Steiner points in the MRST at right reduces the total interconnection cost.

algorithms have high constant-factor overheads, and probabilistic estimates of Steiner tree length do not yield an actual routing *topology*. Furthermore, current methods are based on incremental constructions and iterative improvements, and are not easily parallelized. Ideally, we would like to have a fast MRST heuristic which outputs a complete tree topology, and which also gives good performance as measured by average improvement over MST cost. To this end, we propose a novel "analog" approach to Steiner tree VLSI routing, based on the following simple idea. Suppose that we represent the points P of an instance of the Steiner problem as pegs that are fixed on a flat surface, and suppose that we have an elastic band which initially bounds a region containing all of the pegs. Consider what happens if the air is "sucked out" of this region (or, perhaps, imagine a strong external force imploding the band inward from all sides): the band will collapse in on itself until it encloses zero area, and *this induces a heuristic Steiner tree topology.*

This picture is reminiscent of early ideas in the calculus of variations, e.g., for Plateau's problem on minimum surfaces. As early as 1931, C. V. Boys mentioned similarities between soap-bubble formation and the Steiner problem [1], and Courant and Robbins discussed some physical experiments in [3]. There are also resemblances between our picture and the "elastic-net" [2] [4] [15] and "adaptive-ring" [8] [11] methods proposed in the neural network literature for traveling-salesman variants. (The obstacle-avoidance path planning work of Wong and Funka-Lea [15] in fact uses the term "Steiner point", in the sense of an intermediate point in a *path*, but this is not the standard sense.)

The purpose of this paper is to state a formal algorithm which generically embodies the "bubble-shrinking" analog heuristic for Steiner tree construction. We find that this approach has several appealing features, the most important being a natural mapping to parallel architectures. For example, there are straightforward mappings to connectionist, neural-style algorithms such as the elastic net approach of Durbin and Willshaw [2] [4]. Implementations using cellular automata [14] are also reasonable, and have the added attraction that the heuristic Steiner tree computation in a fixed-size grid will take *constant* time, regardless of the number of pins (in practice, implementations on a fixed grid will simply yield a tree topology which has limited resolution). Our generic algorithm, described below, has been simulated with very promising results: heuristic tree constructions are superior to those of almost all existing methods, and the runtimes are very fast. Diligent search in the literature has failed to uncover even remotely similar constructions, probably because the required *tree* topology is fundamentally different from the *path* topologies of traveling-salesman or path-planning solutions, i.e., a homeomorphism between a minimum Steiner tree and a "necklace" of self-learning nodes in a neural network is not obvious.

The remainder of this paper is organized as follows. Section 2 presents a basic "bubble-shrinking" algorithm, **Shrink-MRST**, and also gives a theoretical performance analysis. Section 3 presents results of extensive Shrink-MRST simulations. Finally, in Section 4 we briefly propose two simple implementations: the first uses "physical" attractive forces to capture the bubble-shrinking idea, and the second uses a natural mapping to a two-dimensional processor array with local, cellular automata-like rules.

2 A Bubble-Shrinking Heuristic

We motivated our algorithm with the picture of n points in the plane surrounded by an imaginary "bubble". Without loss of generality, we assume integer coordinates. Furthermore, because of the rectilinear metric, we may define the bubble, which we call B, to be composed of a cycle of alternating horizontal and vertical *segments* $S_1, S_2, ..., S_m$ in the underlying grid. Again, the central idea is that when the bubble collapses upon itself, segments will meet and fuse together, thereafter remaining fixed in the final tree topology.

It is instructive to consider a simple example of the shrinking process. Figure 2 shows an imaginary bubble shrinking around four points which lie at the corners of a rectangle. Clearly, the longer parallel segments should move toward each other if we are to induce the optimal Steiner tree topology. In an "elastic net" algorithm, this choice can be enforced by an attractive force between nodes on parallel segments that is superlinear in the segment lengths.

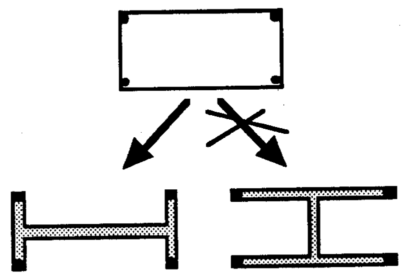

Figure 2: Intuitively, long parallel segments should have greater attraction to each other in order to shrink the bubble properly.

Practical implementation of any "analog" algorithm must entail concessions to digital hardware. For example, we have modeled the bubble as having rectilinear segments, i.e., it is not a smooth closed curve, but this allows us to consider only interactions between parallel segments in modeling attractive forces. We also exploit a result of Hanan [6] which states that if one draws horizontal and vertical lines through each point of P, all Steiner points in the MRST will be intersection points of the resulting ***Hanan grid*** (see Figure 3). Thus, in practice all segments of the shrinking bubble will always lie on the Hanan grid.

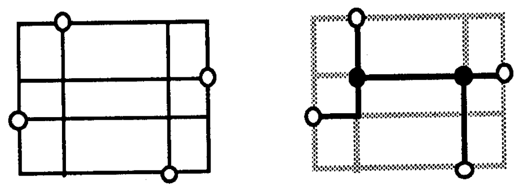

Figure 3: Steiner points of the MRST in the Hanan grid.

Finally, we note that if we begin with a bubble around the periphery of the Hanan grid, bubble-shrinking is equivalent to deleting extremal edges in the grid. In fact, we may view our construction as iteratively removing ***boundary edges*** to induce ***fixed edges*** and new boundary edges until the final tree results. In what follows, we use the following terms (see Figures 3 and 4).

A line segment between adjacent intersection points of the Hanan grid is a ***Hanan edge***. A ***boundary edge*** is a Hanan edge that is in the current boundary, i.e., it is eligible for the flip and shrink operations defined below. A ***tree segment*** is a grid segment that is fixed in the solution, i.e., where boundary edges have already met and fused. The ***current solution*** is the connected union of tree segments and (cycles of) boundary edges. A ***current Steiner point*** is a point in the Hanan grid that is incident to at least three three or more fixed tree segments or boundary edges in the current solution. A ***valid segment*** in the grid is a ***maximal*** union of contiguous boundary edges on a single gridline such that no point of P nor any current Steiner point is strictly interior to the segment.

The actual bubble-shrinking is accomplished by the following operations. (1) A ***flip*** deletes two (valid) segments out of four segments in the grid which form a rectangle; there are four possible flip operations, depending on the orientation of the segments. (2) A ***shrink*** deletes a valid segment and all edges that are perpendicular to the valid segment at gridpoints strictly interior to the segment.

The high-level outline of heuristic **Shrink-MRST** is thus

A typical execution of this Shrink-MRST construction is shown in Figure ??.

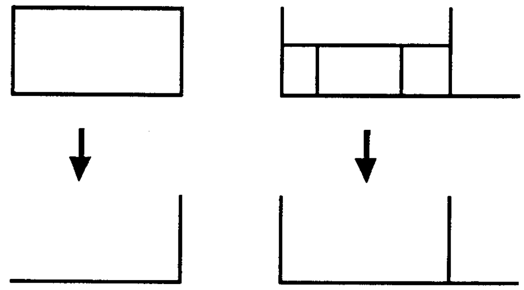

Figure 4: The left figure shows the *flip* operation (four orientations: NW, NE, SE, SW) and the bottom figure shows the *shrink* operation (four orientations: N, W, S, E). The shrink operation models the attractive force between parallel boundary edges in the shrinking bubble.

Shrink-MRST Heuristic Algorithm
Build-Grid;
(grid has total segment cost C)
Initialize-Bubble;
(set boundary edges to be perimeter of grid; use flips to get convex hull of P)
Shrink-Bubble;
(while C still decreasing, shrink longest edge in the grid)

Figure 5: Outline of heuristic Shrink-MRST.

Several results can be proved which bound the performance of our algorithm. The algorithm is easily shown *correct* in the sense that a tree topology will always result. Furthermore, by using a result of Logan and Shepp [12], we can also show that *on average*, the heuristic Shrink-MRST output will be a constant factor away from optimal:

Claim 2: The expected cost of the Shrink-MRST output for n random points in the unit square is within a constant factor of the expected optimal MRST cost. □

This average-case performance bound is reflected in the computational results presented below. However, a construction of n points equally spaced around the unit circle shows that the worst-case performance ratio of our heuristic is not bounded by any constant. Thus Shrink-MRST is, in the worst-case sense, inferior to existing MST-based methods, which have worst-case performance ratio of $\frac{3}{2}$. In practice, unfavorable problem instances are rare: of the 90,000 test cases reported below, only 29 resulted in Shrink-MRST cost greater than the MST cost.

3 Simulation Results

Algorithm Shrink-MRST was simulated via a sequential implementation using ANSI C on Sun-4 hardware. The structure of the implementation follows the template of Figure 5, and output is exemplified by Figure ??. Experimental results on pointsets with cardinality ranging from 4 to 100 (5000 random instances of each cardinality) are summarized in Table 1. Table 1 also provides comparisons with the recent work of Ho, Vijayan and Wong [7], which is the most effective and efficient MST-based heuristic in the literature. It can be seen that for problems of practical size, Shrink-MRST outperforms the other methods. It is also clear that the performance of our method worsens slightly as n becomes very large, since Shrink-MRST in effect forces a "star-like" tree topology that may not be appropriate. In general, our results confirm that Shrink-MRST is not only an effective MRST construction, but also a rare example of an "analog" neural-style heuristic for a hard problem that is competitive with combinatorial heuristics.

# pts	$\frac{MST-Shrink}{MST}$	$\frac{MST-LMST}{MST}$	$\frac{MST-SMST}{MST}$
4	0.0845	NA	NA
5	0.0942	0.082	0.088
6	0.0977	NA	NA
7	0.0989	NA	NA
8	0.0997	NA	NA
9	0.0998	NA	NA
10	0.0992	0.085	0.090
12	0.0983	NA	NA
14	0.0979	NA	NA
16	0.0974	NA	NA
18	0.0974	NA	NA
20	0.0970	0.090	0.095
25	0.0960	0.090	0.095
30	0.0951	0.088	0.094
40	0.0943	NA	NA
50	0.0933	0.093	0.098
100	0.0916	0.097	0.102

Table 1: Performance of Shrink-MRST (% improvement over MST) versus reported results for the two methods in [8]. Results shown are averages for 5000 test cases of each cardinality.

4 Neural Implementations

Our current sequential implementation has $O(n^2 \log n)$ time complexity, since pathological examples can force a quadratic number of shrinking operations, while a heap data structure allows $O(\log n)$ retrieval of longest valid segments. However, the main advantage of our construction is its amenability to parallel implementation. In this section, we briefly propose two practical implementations of the Shrink-MRST method: one is a "physical" neural-style method, and the other exploits the underlying rectilinear grid to derive a cellular automata-like approach. The essential ideas for each are as follows.

(i) The **Physical Analog** (PA) algorithm models the effect of an attractive force that exists only between parallel valid segments of the boundary. As parallel edges are attracted to each other, their rate of motion is given by a high-degree polynomial in the length of each moving edge to enforce the longest-first ordering of the shrink operations in Figure 5 above). When two edges meet and fuse, they are immobilized and removed from the list of valid segments.

We propose a neural-style implementation that is similar to the elastic net construction of Durbin and Willshaw [2] [4]. Here, the boundary of the bubble is discretized and assigned to many distinct processing elements. We also construct an energy functional that is minimized when the bubble encloses zero area. One component of the functional maximizes the difference between total length of fixed tree segments and total length of active boundary segments; this enforces the progressive overlapping of segments in the bubble. A second component gives a "potential" term that is polynomial in the length of and separation between parallel *active* boundary segments: this enforces the proper inward direction of the shrinking. Notice that the second component of the energy functional can be constructed so that the original Shrink-MRST algorithm is very accurately modeled.

An actual implementation will use a fixed number (proportional to the grid perimeter) of processing elements, each carrying position and orientation information for a small part of the bubble. To increase accuracy of the implementation, it is reasonable to dynamically reallocate processing elements from fixed tree segments to the remaining active boundary segments.

(ii) A second, **"Cellular Automata"** (CA) method exploits availability of local and neighborhood information, assuming a two-dimensional grid architecture. Note that the shrink operation alone actually suffices to implement Shrink-MRST: in order to always shrink the ***longest*** valid segment, information regarding a boundary segment's incidence to current Steiner points and points of P must be passed along entire gridlines, but this only requires time proportional to the grid dimension. Thus, even a simple CA implementation offers obvious speedups over our current implementation.

5 Conclusions

We have presented a new parallel approach to minimum rectilinear Steiner tree construction for VLSI routing. The theoretical complexity is good and the approach scales well with problem size. Extensive simulations indicate that the method outperforms virtually all existing combinatorial MRST heuristics, and is thus of practical interest. Several extensions are being pursued, including direct implementation of physical-analog and cellular-automata versions of the basic algorithm.

6 Acknowledgements

Jerry Waldorf provided many interesting discussions and implemented the ShrinkMRST algorithm.

References

[1] C. V. Boys, "The Soap-Bubble", reprinted in *The World of Mathematics*, Vol. II, J. R. Newman, ed., Tempus Press, 1988, pp. 877-886.

[2] D. J. Burr, "An Improved Elastic Net Method for the Traveling Salesman Problem", *Proc. IEEE Intl. Conf. on Neural Networks*, 1988, pp. I-69 - I-76.

[3] R. Courant and H. Robbins, "Plateau's Problem", reprinted in *The World of Mathematics*, Vol. II, J. R. Newman, ed., Tempus Press, 1988, pp. 887-895.

[4] R. Durbin and D. Willshaw, "An Analogue Approach to the Travelling Salesman Problem Using an Elastic Net Method", *Nature* 326(6114) (1987), pp. 689-691.

[5] M. Garey and D. S. Johnson, "The Rectilinear Steiner Problem is NP-Complete", *SIAM J. of Applied Math.* 32(4) (1977), pp. 826-834.

[6] M. Hanan, "On Steiner's Problem With Rectilinear Distance", *SIAM J. of Applied Math.* 14 (1966), pp. 255-265.

[7] J.-M. Ho, G. Vijayan and C. K. Wong, "New Algorithms for the Rectilinear Steiner Tree Problem", *IEEE Transactions on Computer-Aided Design* 9(2) (1990), pp. 185-193.

[8] G. Hueter, "Solution of the Traveling Salesman Problem with an Adaptive Ring", *Proc. IEEE Intl. Conf. on Neural Networks*, 1988, pp. I-85 - I-92.

[9] F. K. Hwang, "On Steiner Minimal Trees with Rectilinear Distance", *SIAM J. Applied Math.* 30(1) (1976), pp. 104-114.

[10] F. K. Hwang, "An $O(n \log n)$ Algorithm for Rectilinear Minimal Spanning Trees", *J. Assoc. for Computing Machinery* 26(2) (1979), pp. 177-182.

[11] T. Kohonen, *Self-Organization and Associative Memory*, New York, Springer-Verlag, 1984.

[12] B. F. Logan and L. A. Shepp, "A Variational Problem for Random Young Tableaux", *Adv. in Math.* 26 (1977), pp. 206-222.

[13] D. Richards, "Fast Heuristic Algorithms for Rectilinear Steiner Trees", *Algorithmica* 4 (1989), pp. 191-207.

[14] S. Wolfram, *Theory and Applications of Cellular Automata*, Singapore, World Scientific, 1986.

[15] W. S. Wong and C. A. Funka-Lea, "An Elastic Net Solution to Obstacle Avoidance Tour Planning", *Proc. IEEE/INNS Intl. Joint Conf. on Neural Networks*, 1989, pp. III-799 - III-804.

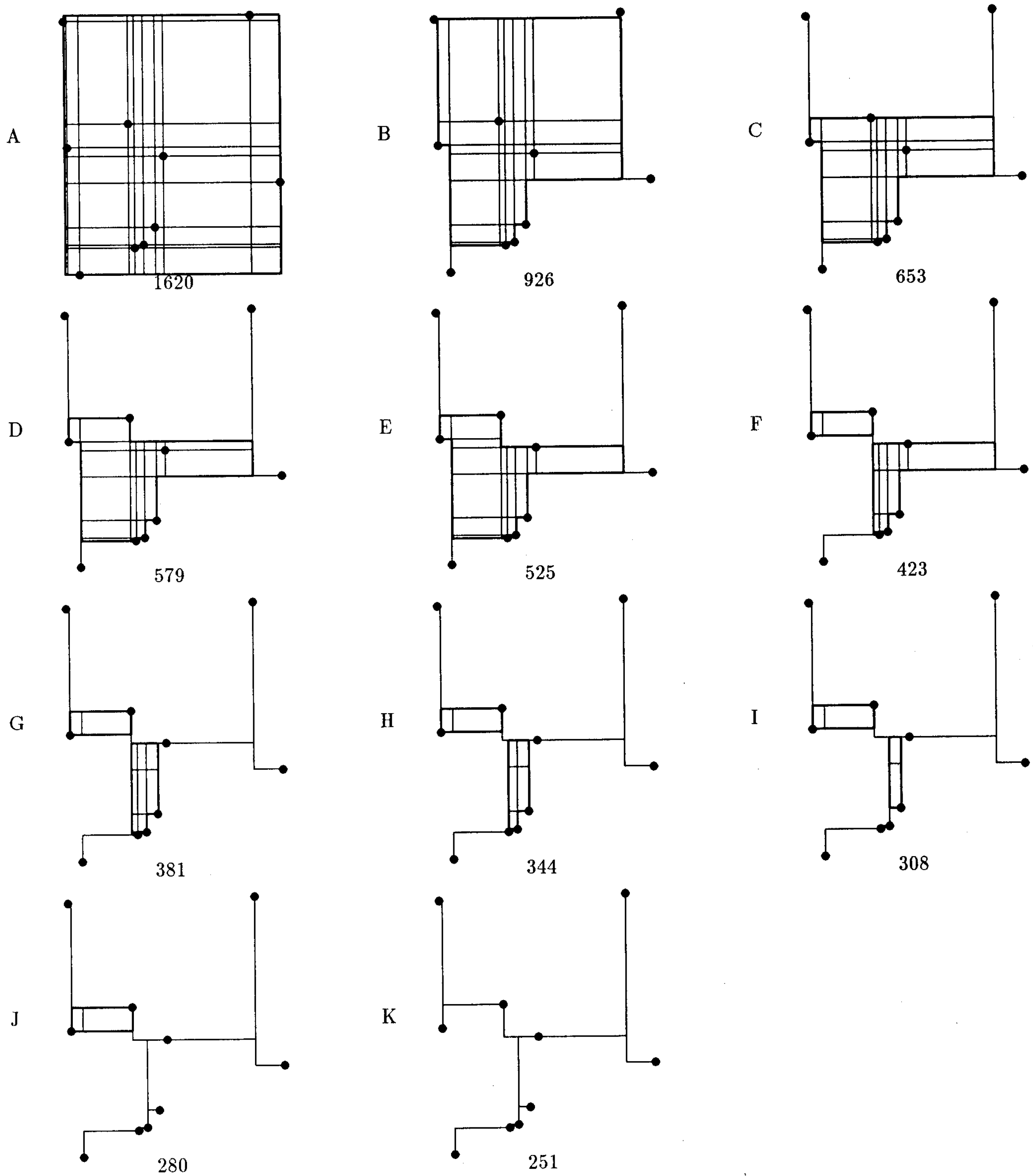

Figure 6: Execution of Shrink-MRST on a 10-point example. Flips occur between (A) and (B); shrinks occur, e.g., between (E) and (F). Boundary edges are shown in bold. Numbers shown are sums C of costs of all remaining segments in the underlying Hanan grid. For this example the heuristic Steiner tree, with cost = 251, is optimal.