An Effective Analog Approach to Steiner Routing

Andrew Kahng

Dept. of Computer Science, UCLA, Los Angeles, CA 90024-1596

Abstract

Construction of a minimum rectilinear Steiner tree (MRST) is a fundamental problem in the physical design of VLSI circuits. The problem is NP-complete, and numerous heuristics have been proposed. We propose a new "analog" approach which intuitively shrinks a bubble around the pins of the signal net until a Steiner tree topology is induced. The method easily maps to parallel neural-style architectures, as well as to fairly generic two-dimensional processor arrays. Extensive simulation results show better performance than virtually all existing MRST approaches. The result is a rare instance where an "analog" heuristic for an NP-complete problem outperforms existing combinatorial methods, both in time complexity and in 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 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 Steiner trees connecting the terminals of a signal net. The MRST problem is NP-complete [6], and over the years a number of heuristics have been developed [13].

A large subset of the MRST literature involves refining an initial *minimum spanning tree* (MST) topology to yield a heuristic Steiner tree. Hwang [9] showed that the worst-case ratio of MST cost to MRST cost is $\frac{3}{2}$, hence MST-improvement methods are attractive in that they also enjoy the $\frac{3}{2}$ worst-case bound. Variations of the MST-based approach have improved the computation and the subsequent refinement of the initial MST, culminating with the recent work of Ho, Vijayan and Wong [8] which efficiently constructs the *optimal* Steiner tree derivable from a given MST.

The standard testbed for MRST heuristics consists of random sets of n points from a uniform distribution in the unit square; this testbed reflects the statistical properties of placed netlists [13]. For such inputs, both the expected minimum spanning tree (MST) cost and the expected MRST cost grow as $\Theta(\sqrt{n})$ [14]. Therefore, MRST heuristics are usually evaluated by their average cost improvement over the MST routing. Virtually all algorithms yield from 7% to 9% improvement over MST cost on random inputs (e.g., the optimal MST-based method of [8] averages just over 9% improvement for typical problem sizes). It has been conjectured that no more than 12% average improvement over MST cost is possible [1].

In addition to good performance as measured by average improvement over MST cost, desirable properties of an MRST heuristic include:

- *Speed*: current methods have high complexity or large constant-factor overheads in computational geometry subroutines.
- *Tree topology output*: Rough tree cost estimates based on probabilistic wiring models or net bounding box dimensions are less useful for, e.g., congestion analysis.
- *Parallelizability*: current incremental constructions do not map well to parallel machines.

With this in mind, we propose a new "analog" approach to Steiner tree computation. The method is based on the following idea. Represent the points P of a net by fixed pegs on a flat surface, and suppose that we have an elastic band which initially bounds a region containing all of the pegs. If the air is "sucked out" of this region (or imagine a strong force imploding the band inward from all sides), the band will collapse in on itself until it encloses zero area: *this induces a heuristic Steiner tree topology.* Such a picture recalls early ideas in the calculus of variations, particularly for Plateau's problem of minimum surfaces. As early as 1931, C. V. Boys mentioned similarities between soap-bubble formation and the Steiner problem [2], and Courant and Robbins discussed related physical experiments in [4]. There are also resemblances between our idea and the "elastic-net" [5] mappings proposed in the neural network literature for the traveling salesman problem.

In this paper, we formalize the "bubble-shrinking" heuristic for Steiner tree construction. Our approach is naturally implemented in parallel, e.g., on connectionist, neural-style architectures or on two-dimensional arrays of processors. The generic algorithm described below has been simulated with very promising results: heuristic tree cost is smaller than almost all existing methods, and runtimes are very fast. The remainder of this paper is organized as follows. Section 2 describes a basic sequential algorithm,

Bubble-MRST, and gives theoretical performance bounds. Section 3 presents extensive simulation results. In Section 4, we propose two simple parallel implementations, and Section 5 concludes with directions for future work.

2 A Practical Heuristic

We motivated our algorithm with the picture of n points in the plane surrounded by an imaginary "bubble". Any "analog" algorithm will necessarily lose some of its flavor when implemented on digital hardware, and our method is no exception. Because of the rectilinear metric, the bubble is composed of alternating horizontal and vertical *segments* in the underlying grid. Thus, our rectilinear bubble is not as smooth as a real bubble. However, this allows us to only consider interactions between parallel segments when we model attractive forces. When the bubble collapses upon itself, parallel segments will attract each other, meet, and fuse together to form fixed edges in the final tree topology. (Note that the bubble cannot shrink *through* a point, or else an incomplete topology may result.)

We may also use a result of Hanan [7], which states that all Steiner points in the MRST will be intersection points of the *Hanan grid* formed by drawing horizontal and vertical lines through each point of P (Figure 1).

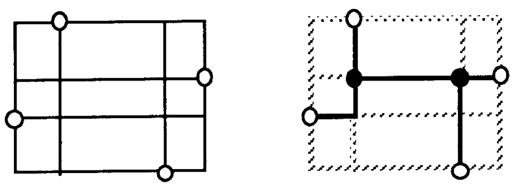

Figure 1: The optimal MRST is a subtree of the Hanan grid.

Thus, at every time step we force all bubble segments to lie on the Hanan grid, and bubble-shrinking is equivalent to deleting extremal edges in the grid. In fact, our construction is a simple iterative process of removing *boundary edges* to induce *fixed edges* and new boundary edges until a final tree topology is determined. In what follows, we use the following terms.

A *Hanan edge* is a line segment between adjacent intersection points of the Hanan grid. A *boundary edge* is a Hanan edge that is in the current boundary, i.e., it is eligible for the shrink operations defined below. A *fixed (tree) edge* is a Hanan edge that is a permanent part of the solution, i.e., where boundary edges have already met and fused. The *current solution* is the connected union of fixed tree edges and 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 lies within the segment.

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 in order to induce the optimal Steiner tree topology. In an "analog" algorithm that models physical forces, this choice can be enforced by an attraction between parallel bubble segments that is superlinear in the segment lengths.

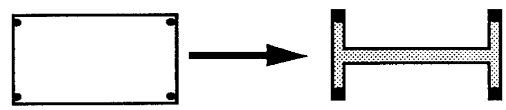

Figure 2: Long parallel segments have greater attraction to induce the proper Steiner tree.

With these observations, the bubble-shrinking is accomplished via a single repeated operation:

Definition: A *shrink* operation deletes the longest valid segment, along with all edges that are perpendicular to the segment at gridpoints strictly interior to the segment.

The high-level outline of the **Bubble-MRST** heuristic is given in Figure 3, and a typical execution of the Bubble-MRST construction is shown in Figure 4.

Algorithm Bubble-MRST
  Build-Grid
    mark points on all grid lines which pass through some point of P;
    (grid has total total segment cost C)
  Initialize-Bubble
    set active boundary edges to be perimeter of grid;
  Shrink-Bubble
    while C still decreasing, *shrink* longest valid segment in grid;

Figure 3: Algorithm Bubble-MRST.

Several results can be proved which bound the performance of our algorithm. First, since the area enclosed by the bubble goes to zero, the construction is correct:

Theorem 1: Heuristic Bubble-MRST always outputs a tree topology. □

We can establish a tight bound on the expected Bubble-MRST heuristic tree cost by tracing the evolution of boundary edges and using a result of Logan and Shepp [11]:

Theorem 2: The expected cost of the Bubble-MRST output for n points randomly chosen in the unit square is $\Theta\sqrt{n}$. □

The result of Steele [14], that subadditive functionals (e.g., MRST cost) grow as $\Theta\sqrt{n}$ over random pointsets in the unit square, implies:

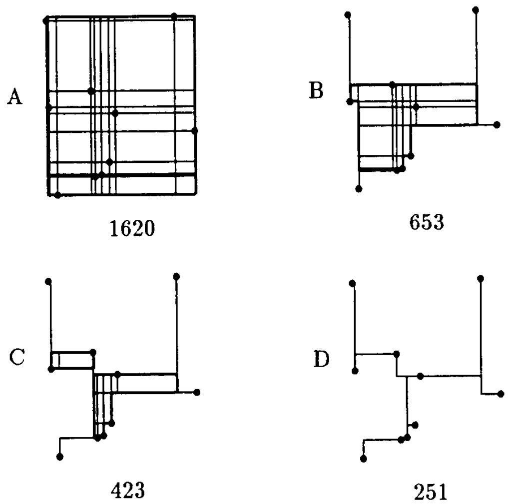

Figure 4: Stages from a sample run of the Bubble-MRST algorithm on a 10-point example, showing decrease in total edge cost of the remaining grid.

Corollary: On average, the Bubble-MRST output is bounded by a constant factor from the expected optimal MRST cost. □

This average-case performance bound is reflected in the computational results presented below. However, the absolute performance ratio of our heuristic is easily seen to be unbounded. For example, an input of n points equally spaced on the unit circle will result in a heuristic tree that passes through the center of the circle, a highly suboptimal solution. We thus have:

Theorem 3: The performance ratio of Bubble-MRST is not bounded by any constant. □

Therefore, Bubble-MRST is in the worst-case sense inferior to existing MST-based methods which have a performance bound of $\frac{3}{2}$. However, note that unfavorable instances are very rare: of the 180,000 test cases reported below, only 51 resulted in Bubble-MRST cost greater than the MST cost.

3 Computational Results

The Bubble-MRST algorithm was implemented in a UNIX/C environment on Sun-4 hardware. The structure of the implementation follows the template of Figure 3, and output is exemplified by Figure 4. Code is available from the author.

Experimental results on pointsets with cardinalities ranging from 4 to 100 (10000 random instances for each cardinality) are summarized in Figure 5. Figure 5 also provides comparisons with the MST improvement values reported for the "S-MST" algorithm of Ho, Vijayan and Wong [8]. For problems of practical size, Bubble-MRST clearly outperforms this *optimal* MST-derived heuristic construction. As n becomes very large (over 40 pins), the performance of our method worsen while the methods in [8] improve. This is because Bubble-MRST essentially forces a star-like tree topology that is suboptimal for cases such as points along the unit circle. However, we note that this may turn out to be a useful side-effect: as feature sizes decrease, diameter of the global routing gives an increasingly significant bound on the circuit timing [3], and Bubble-MRST naturally yields good heuristic solutions for minimum-diameter (i.e., minimum signal delay) global routing formulations. In general, the simulation results indicate that our method not only provides a new and effective MRST construction, but is also a rare example of an "analog" heuristic for a hard problem that is competitive with combinatorial heuristics.

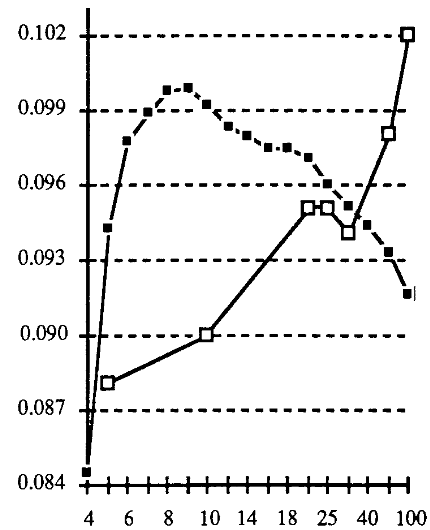

Figure 5: Comparison between Bubble-MRST (square dots) and the optimal MST-based (S-MST) method of Ho, Vijayan and Wong (boxes). The horizontal axis gives net size, and the vertical axis gives improvement over MST cost averaged over 10000 random instances.

4 Parallel Implementations

Our current 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 the method is its amenability to parallel implementation. In this section, we briefly propose two practical implementations of the Bubble-MRST method: one is a "physical" method with an attractive force model, and the other exploits the underlying rectilinear grid to derive a cellular automata style approach. The essential ideas for each are as follows.

(I) The *attractive force* (AF) method posits an attraction that exists only between parallel valid segments of the boundary. When parallel edges are attracted to each other, their rate of motion is given by a high-order polynomial in the length of each moving edge; this enforces the proper longest-first ordering of the shrink operations in Figure 3 above. When two edges meet and fuse, they are immobilized and removed from the list of valid segments. A neural-style implementation of the AF method is via the elastic net construction of Durbin and Willshaw [5], where the boundary of the bubble is discretized and assigned to many distinct processing elements. The shrinking is accomplished by constructing an energy functional that is minimized when the bubble encloses zero area. For example, we want to maximize the difference between total length of fixed tree segments and total length of active boundary segments; we also want to minimize a "potential" term that is polynomial in the length of and separation between parallel active boundary segments. 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 the accuracy with which the shrinking process is modeled, we dynamically reallocate processing elements from fixed tree segments to the remaining active boundary segments.

(II) The *2-D array* (2DA) method assumes a two-dimensional grid architecture and uses local and neighborhood information to implement the "shrink" operation. 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 constant time (proportional to the fixed grid dimension). The straightforward heuristic computation thus requires quadratic time in a fixed-size grid, and we suspect that more careful analysis/implementation will further improve the 2DA time complexity. Note that in practice, implementations on a fixed grid will simply yield a tree topology which has coarser resolution; the result is still useful for global routing.

A more natural "cellular-automata"-like implementation results if all boundary segments are allowed to shrink simultaneously. Among other things, this would yield in a *constant-time* heuristic Steiner tree construction, regardless of the size of P. We have simulated this variant but, as expected, performance is noticeably worse than that of the original Bubble-MRST since highly star-like topologies result.

5 Conclusions

We have presented a new parallel approach to MRST construction for VLSI routing. The theoretical complexity of the Bubble-MRST algorithm is good, and extensive simulations indicate that the method outperforms virtually all existing combinatorial MRST heuristics. Furthermore, parallel speedup seems straightforward via neural-network or two-dimensional grid architectures. Our basic algorithm and the implementation variants extend to other routing geometries. Extensions to weighted gridgraphs for routing in congested regions do not seem easy, as our method seems to rely on a geometric embedding. A number of other extensions are being pursued, particularly the direct implementation of the AF and 2DA versions of the basic algorithm.

6 Acknowledgements

Jerry Waldorf provided many interesting discussions and implemented the Bubble-MRST algorithm.

References

[1] M. W. Bern and M. Carvalho, "A Greedy Heuristic for the Rectilinear Steiner Tree Problem", *technical report* UCB-CSD 87-306, 1987.

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

[3] J. Cong, A. Kahng, G. Robins, M. Sarrafzadeh and C. K. Wong, "Performance-Driven Global Routing for Cell Based IC's", *Proc. IEEE ICCD*, Cambridge, October 1991.

[4] 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.

[5] 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.

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

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

[8] 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.

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

[10] A. Kahng and G. Robins, "A New Family of Steiner Tree Heuristics With Good Performance: The Iterated 1-Steiner Approach", *Proc. IEEE Intl. Conf. on Computer-Aided Design*, Santa Clara, November 1990, pp. 428-431.

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

[12] B. Preas and M. Lorenzetti, eds., *Physical Design Automation of VLSI Systems*, Benjamin Cummings, 1988.

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

[14] J. M. Steele, "Growth Rates of Euclidean Minimal Spanning Trees With Power Weighted Edges", *The Annals of Probability* 16(4) (1988), pp. 1767-1787.

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