A New Class of Steiner Tree Heuristics with Good Performance: the Iterated 1-Steiner Approach

Andrew Kahng and Gabriel Robins

UCLA Department of Computer Science
Los Angeles, California 90024

Abstract

Virtually all previous methods for the rectilinear Steiner tree problem begin with a minimum spanning tree *topology* and rearrange edges to *induce* Steiner points. This paper gives a more direct approach: we iteratively find *optimal* Steiner points to be added to the layout. Our method gives improved average-case performance, and also escapes the worst-case examples of existing approaches. Sophisticated computational geometry techniques allow efficient and practical implementation, and the method is naturally suited to real-world VLSI regimes where, e.g., via costs can be high. Extensive performance results show almost 3 percent wirelength reduction over the best existing methods. We describe a number of variants and extensions, and suggest directions for further research.

1 Introduction

The minimum rectilinear Steiner tree (MRST) problem is as follows: given N points in the plane, find a minimum-length tree of rectilinear edges which connects the points. This problem has been extensively studied, and important applications arise in such circuit design phases as wirability analysis and global routing. The problem is NP-complete [3], and a number of heuristics have been proposed [14] which resemble classic minimum spanning tree (MST) construction methods. Hwang [9] showed that the rectilinear MST itself is an approximation to the MRST with worst-case ratio $length(MST)/length(MRST) \leq 3/2$, and a fundamental open question over the years has been whether there exists an MRST heuristic with worst-case performance ratio less than 3/2.

The MRST heuristics proposed thus far have very similar performance on random instances (i.e., average heuristic RST length being 7-9 % smaller than MST length), and have tight worst-case bounds of 3/2, the same as for the simple rectilinear MST. This effective similarity has been noted in [14] and [19], and is confirmed by our own empirical studies. Two recent examples of MST-improvement methods are due to Ho, Vijayan and Wong [8], and Hasan, Vijayan, and Wong [7]. These have recently joined the list of methods which have worst-possible worst-case performance bound 3/2 (i.e., as large as that of the simple MST [11] (see Figure 1 and Figure 2), and it seems unlikely that any MST-inspired heuristic will have a performance ratio less than 3/2. Steiner routing for VLSI is surveyed in [10], [14] and [19].

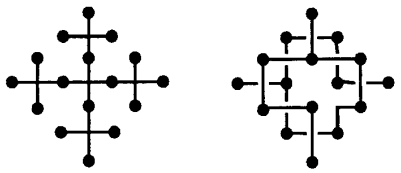

Figure 1: An example where the cost ratio MST-derived-RST / MRST = 3/2.

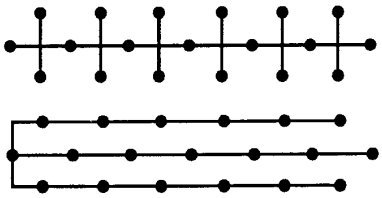

Figure 2: An example where the cost ratio of a *separable* (in the terminology of [8]) MST-derived-RST to the MRST is arbitrarily close to 3/2.

It has been shown [1][5] that the optimal Steiner tree, as well as heuristic MST-based RSTs, will have a linear number of Steiner points. With preferred-direction wiring planes, each Steiner point usually corresponds to an additional via. In certain (board) wiring technologies, or for chip reliability considerations, having so many Steiner points may not be desirable. Ideally, one would like to *prescribe* the relative incidence of Steiner points as a routing parameter which might depend on technology or estimated layout congestion, but this is not a natural concept with an MST-based method.

In the extreme case where extra vias are very expensive, it is natural to ask the following: if *exactly one* Steiner point can be introduced into a net, where should it be placed? This is the motivation for our *Iterated 1-Steiner* heuristic, which repeatedly adds the best possible Steiner point to the layout until no further improvement is possible. The advantages of this method are several: (1) we can limit the algorithm so that it introduces only k Steiner

CH2924-9/90/0000/0428$01.00 © 1990 IEEE

points (e.g., in a layout regime where vias are expensive), (2) the method can be efficiently implemented by applying recent computational geometry results, (3) the method performs significantly better than all previous MST-based methods, yielding an average improvement of 10 to 11 % over MST lengths, and (4) the method is amenable to a number of extensions, such as randomization, partial amortization of computations, parallel implementation, and application to higher-dimensional or alternate-metric geometries. The method, its variants, and a number of formal and empirical results are fully described in [12].

2 A New Approach.

Definition: Given a set $P = \{p_1, \ldots, p_N\}$ of points in the plane, the *1-Steiner point* is the point x such that the length of MST($P \cup \{x\}$) is minimized, and $MST(P \cup \{x\}) < MST(P)$. (Where unambiguous, $MST(P)$ denotes the length of the MST on point set P.) The *1-Steiner tree* is the minimum spanning tree on $P \cup \{x\}$.

Our approach is to iteratively calculate 1-Steiner points and add them to the set of Steiner points S. With each added point, the length of the MST on $P \cup S$ will decrease. If there is no x such that $MST(P \cup S \cup \{x\}) < MST(P \cup S)$, we terminate the construction.

If there is no x such that $MST(P \cup S \cup \{x\}) < MST(P \cup S)$, we terminate the construction.

For a net with N pin locations in the grid, the *Steiner candidate set* consists of all points whose x and y coordinates are both shared with two of the N pins. Hanan [6] showed that an optimal MRST exists whose Steiner points are taken from this set. Thus, the 1-Steiner tree is found by examining N^2 Steiner candidates, constructing a new MST for each set of $N+1$ points, then picking the best candidate.

By using the Voronoi diagram [13], the MST computation can be done in $O(NlogN)$ time, requiring $O(N^3 logN)$ time per added Steiner point. Georgakopoulos and Papadimitriou [4] give an $O(N^2)$ method for computing the 1-Steiner *tree* in the Euclidean plane, and their method is readily adapted to the Manhattan norm [12], yielding an efficient implementation.

2.1 The Algorithm.

We now state the **Iterated 1-Steiner** heuristic:

```
while the number of Steiner points is less than N
    and a 1-Steiner point x exists,
        add x to the point set.
```

There are at most N iterations, each requiring $O(N^2)$ computation [4], and therefore the time complexity of this method is $O(N^3)$. Empirical results indicate that Iterated 1-Steiner significantly outperforms all existing heuristics.

A variant of this heuristic consists of adding a *random* improving Steiner point (rather than the best one) at each step. The number of Steiner points produced may be kept low by removing degree-1 or degree-2 points in subsequent MSTs; by the triangle inequality the latter can be removed without increasing the MST length, and the former can trivially be removed from the layout:

```
while 1-Steiner points exist
    add a randomly selected 1-Steiner point.
    delete Steiner points of degree 1 or 2 in MST.
```

Iterated Random 1-Steiner clearly terminates (because the MST length decreases monotonically), and has a simple, compact implementation. Performance is worse than Iterated 1-Steiner, but remains slightly better than MST-derived methods for typical instances.

The Iterated 1-Steiner heuristic and its variants have as their closest conceptual relative a method of Smith and Liebman [16][17]. Our method is preferable on several grounds: (1) *performance*: the method in [16] gives less than 8% average improvement over MST length for random point sets [10], while our method gives almost 11% average improvement; (2) *efficiency*: [16] gives an $O(N^4)$ method, while the Iterated 1-Steiner algorithm is $O(N^3)$; (3) *simplicity*: the algorithm in [16] requires seven pages to describe while our method is simply described and coded.

2.2 A Batched Variant.

In order to reduce the running time of our heuristic, we still use the approach of [4] to compute the optimal 1-Steiner point and its associated MST cost savings, but instead of selecting only the Steiner candidate with highest cost savings, we select a maximal "independent" set of SPs, similar to the approach of [7]. The criterion for "independence" here is that candidate Steiner points x and y may be added in the same round only if $\Delta MST(P, \{x\}) + \Delta MST(P, \{y\}) \leq \Delta MST(P, \{x, y\})$, where $\Delta MST(P, S) = max(0, MST(P) - MST(P \cup S))$. A candidate SP x satisfies $\Delta MST(P, \{x\}) > 0$.

The running time is kept low, using such techniques as planar subdivision search [13] and dynamic MST maintenance [2]. The total time required for an entire round is $O(N^2 logN)$:

```
while S = {x | ΔMST(P, {x}) > 0} ≠ ∅ do
    P' = P
    for x in {S sorted by descending ΔMST} do
        if ΔMST(P, {x}) ≤ ΔMST(P', {x})
            then P' = P' ∪ {x}
    P = P'
```

Empirical data indicates that the number of rounds required grows much more slowly than the number of Steiner points produced. For example, for pointsets of size 40, where the average number of SPs produced is about 17 (with a max of 22), the average number of rounds for Batched 1-Steiner is only 2.05 (with a max of 4). We conjecture that the number of rounds grows only sub-linearly as a function of $|P|$.

3 Analysis of the Approach.

3.1 Performance Ratio.

It is not difficult to show that the 1-Steiner heuristic is optimal for four or fewer points. On the other hand, we have found a 9-point example where the 1-Steiner heuristic performs as badly as 13/11 times optimal (Figure 3).

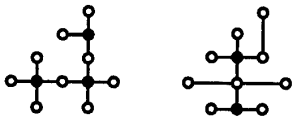

Figure 3: A 9-point example where the Iterated 1-Steiner performance ratio is 13/11.

It is encouraging to note that for every other MRST heuristic in the literature, there are five- or six-point examples which *force* a performance ratio of 3/2; in contrast, the worst-case performance ratio of Iterated 1-Steiner for a five-point example seems to be 7/6. We conjecture that the Iterated 1-Steiner method has performance ratio strictly less than 3/2.

3.2 The Number of Steiner Points.

Trivially, there cannot be more than $O(N^2)$ Steiner points produced by our algorithm, since there are only N^2 candidate points. In practice, our method requires far fewer Steiner points. We have found a four-point example for which three Steiner points are produced by Iterated 1-Steiner, although we can easily enforce the $N-2$ bound by removing degree-2 and degree-1 points as we did in Iterated Random 1-Steiner. The average number of Steiner points seems to grow as approximately $N/3$, and it is interesting to note that most of the "win" (as a percentage improvement over the MST length) occurs in the first several 1-Steiner iterations/rounds: for 40-point instances, about 95% of the total improvement comes in the *first* round, and over 99% of the improvement comes in the first two rounds of Batched 1-Steiner.

3.3 Computational Results.

We coded these heuristics, along with several existing methods, using ANSI C (the code is available from the authors upon request). For typical values of N, 5000 N-point instances were solved using all methods. The instances were generated randomly from a uniform distribution in a fixed-size grid; this is the standard testbed for Steiner tree heuristics [14]. The results are depicted graphically in Figure 4, and are summarized more comprehensively in [12].

3.4 Extensions.

Since multiple-layer wiring, two-sided PCB design, and three-dimensional VLSI technologies are proliferating, we briefly mention several advantages of our approach in higher dimensions. It is not difficult to see that Hanan's theorem still holds in all higher dimensions [18], and we have conjectured [11] that $MST \leq \frac{2d-1}{d} \cdot MRST$ holds in d-space, the obvious generalization of Hwang's theorem. Figure 5 illustrates (in three dimensions) an infinite family of pointsets for which our 1-Steiner scheme performs *optimally* yet all other MST-based heuristics perform as badly as $\frac{2d-1}{d}$ times optimal, which is no better than the MST length.

The previous standard approach, i.e., improving an initial MST solution, becomes much harder in three dimensions since there are more orientations for each edge; this suggests that the benefit of using a constructive 1-Steiner strategy increases in this higher dimension. We are currently studying the complexity of the 1-Steiner approach in higher dimensional geometries. The 1-Steiner approach also succeeds in the presence of non-orthogonal wiring directions [15], and Hanan's result also generalizes to such geometries.

3.5 On Meta-Heuristics.

Given an instance of problem P and n different heuristics (algorithms) H_1, H_2, ...,H_n, we define a meta-heuristic $H(H_1,\ldots,H_n)$ as follows: execute the n heuristics separately, and output the best among the n outputs. The total running time is asymptotically dominated by the slowest H_i.

The approximation performance is better than the best performance of any single method, since the various methods can trade off in their "areas of expertise". To illustrate this phenomenon, the Steiner tree heuristics Corner (similar to the method of [8]) and Prim (a simple analog of Prim's MST heuristic), when used together, give an average performance of about half a percent better than Corner alone, although the average performance of Prim is about two percent *worse* than that of Corner.

The meta-heuristic H(Prim, Corner, 1-Steiner) gives essentially the same performance as 1-Steiner alone, implying that 1-Steiner *strictly dominates* the other methods. This is a very important aspect: it suggests that the 1-Steiner method will *universally* give "reasonably good" solutions.

The meta-heuristic is a general algorithmic phenomenon that applies to numerous other problems and subareas of computer science. There is very little evidence in the literature to indicate that this phenomenon, especially for *heuristics*, has received the attention it deserves. Particularly in light of advances in parallel computation and hardware implementation of algorithms, such composite methods should become a highly fertile avenue of research in (practical) optimization.

4 Conclusion.

In this paper, we have presented a fast new approach to the rectilinear Steiner problem. The algorithm is

practical, parallelizes readily, and yields results that reduce wirelength by several percent over the best previous methods. Randomized and "batched" variants have also proved successful, and the approach extends to non-orthogonal wiring and three-dimensional layout technologies.

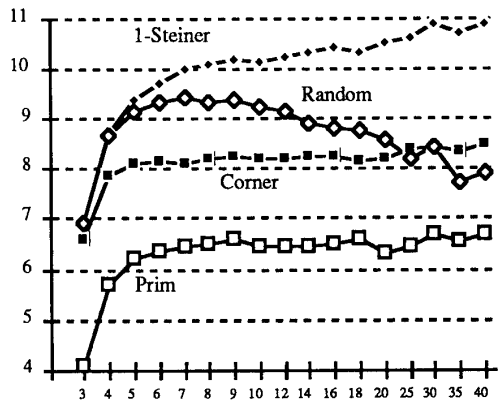

Figure 4: Performance comparison of the heuristics. Horizontal axis represents number of points per set; vertical axis represents performance improvement over MST in percent.

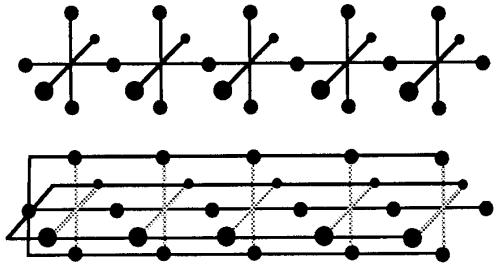

Figure 5: An example for $D = 3$ where the ratio optimal-MST-derived-RST / MRST is arbitrarily close to 5/3.

References

[1] M. W. Bern, "Two Probabilistic Results on Rectilinear Steiner Trees", *Algorithmica* 3 (1988), pp. 191-204.

[2] D. Eppstein, G. Italiano, R. Tamassia, R. E. Tarjan, J. Westbrook and M. Yung, "Maintenance of a Minimum Spanning Forest in a Dynamic Planar Graph", *Proc. ACM-SIAM Symp. on Discrete Algs.*, 1990, pp. 1-11.

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

[4] G. Georgakopoulos and C. H. Papadimitriou, "The 1-Steiner Tree Problem", *J. Algorithms* 8 (1987), pp. 122-130.

[5] E. N. Gilbert and H. O. Pollak, "Steiner Minimal Trees", *SIAM J. App. Math.* 16 (1968), pp. 1-29.

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

[7] N. Hasan, G. Vijayan and C. K. Wong, "A Neighborhood Improvement Algorithm for Rectilinear Steiner Trees", *Proc. of ISCAS*, 1990.

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

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

[10] F. K. Hwang, "The Rectilinear Steiner Problem", *J. Design Automation and Fault-Tolerant Computing* (1978), pp. 303-310.

[11] A. Kahng and G. Robins, "On Performance Bounds for Two Rectilinear Steiner Tree Heuristics in Arbitrary Dimension", 1990, submitted to *IEEE Trans. on Computer-Assisted Design.*

[12] A. Kahng and G. Robins, "A New Family of Steiner Tree Heuristics With Good Performance: The Iterated 1-Steiner Approach", *Technical Report*, UCLA CS Department, #CSD-900014., April 1990.

[13] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, New York, Springer-Verlag, 1985.

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

[15] M. Sarrafzadeh and C. K. Wong, "Hierarchical Steiner Tree Construction in Uniform Orientations", *draft*, 1990.

[16] J. M. Smith and J. S. Liebman, "Steiner Trees, Steiner Circuits and the Interference Problem in Building Design", *Engineering Optimization* 4 (1979), pp. 15-36.

[17] J. M. Smith, D. T. Lee and J. S. Liebman, "An $O(N \log N)$ Heuristic Algorithm for the Rectilinear Steiner Minimal Tree Problem", *Engineering Optimization* 4 (1980), pp. 179-192.

[18] T. L. Snyder, "A Simple and Faster Algorithm for the Rectilinear Steiner Problem in General Dimension", *Proc. ACM Symp. on Comp. Geometry*, 1990, *to appear.*

[19] P. Winter, "Steiner Problem in Networks: A Survey", *Networks* 17 (1987), pp. 129-167.