A New Class of Iterative Steiner Tree Heuristics with Good Performance

Andrew B. Kahng, *Associate Member, IEEE*, and Gabriel Robins, *Student Member, IEEE*

Abstract—The minimum rectilinear Steiner tree (MRST) problem is very important for such aspects of physical layout as global routing and wiring estimation. Virtually all previous heuristics for computing rectilinear Steiner trees begin with a minimum spanning tree (MST) *topology* and rearrange edges to *induce* Steiner points. This paper gives a more direct approach which makes a significant departure from such spanning tree-based strategies: we iteratively find *optimal* Steiner points to be added to the layout. Our method not only gives improved average-case performance, but also escapes the worst-case examples of existing approaches. We show that the performance ratio of our method can *never* be as bad as $3/2$, and is in fact bounded by $4/3$ on the entire class of instances where the $c(MST)/c(MRST)$ cost ratio is exactly $3/2$. Sophisticated computational geometry techniques allow efficient and practical implementation, and the method is naturally suited to technological regimes where, e.g., via costs can be high and the number of Steiner points should be limited. Extensive performance results show a 2% to 3% wire length reduction over the best previous heuristics. We describe a number of variants and extensions, and suggest directions for further research.

I. 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 Steiner trees connecting the pins of a signal net.

Several results have greatly influenced the progress of research on the MRST problem. First, Hanan showed in 1966 that if one draws horizontal and vertical grid lines through each of the points in P, there is an MRST whose Steiner points S are all chosen from among the intersection points (the *Steiner candidate set*) in the resulting grid [8]. Second, Garey and Johnson showed that despite this restriction on the solution space, the MRST problem is NP-complete [5]. Thus, a number of heuristics have been proposed, as surveyed recently in [16] and [23]. When

attacking intractable problems, a basic goal is to devise *provably good* heuristics, typically in the sense of having bounded worst-case error from optimal. Therefore, a third fundamental result is that of Hwang [11], who showed that the rectilinear *minimum spanning tree* (MST) over P is a fairly good approximation to the MRST, with a worst-case performance ratio of $3/2$; i.e., if $c(T)$ denotes the total cost of a tree T, $c(MST)/c(MRST) \leq 3/2$. Fig. 1 shows an MST and an MRST for the same four-pin net. The result of Hwang implies that any *MST-based* strategy which improves upon an initial MST topology will also have performance ratio of at most $3/2$. Thus, a number of Steiner tree heuristics resemble classic MST construction methods.

Examples of this approach include two recent MRST heuristics from Ho *et al.* [10] and Hasan *et al.* [9]. The first gives a linear-time construction for the *optimal* rectilinear Steiner tree (RST) derivable from a given MST, i.e., lying within the union of the bounding boxes of the MST edges. The second heuristic also begins with an MST topology, and iteratively adds as many "locally independent" Steiner points as possible. Because the output of these heuristics will not have greater cost than the MST, they are indeed provably good, with a worst-case performance ratio of $3/2$ by the result of Hwang [11]. In practice, existing MRST heuristics exhibit very similar performance on random instances (n points generated from a uniform distribution in the unit square); the heuristic Steiner tree cost is 7% to 9% smaller than MST cost on average [16], [23]. A fundamental open issue has been to find a heuristic method with performance ratio strictly less than $3/2$.

The worst-case bound given by Hwang and such average-case bounds as that given by Steele[1] provide compel-

[1]A more theoretical, retrospective justification for MST-based approaches is based on asymptotics of subadditive functionals [1], [22] in the L_p plane (i.e., where the distance function in the plane is given by $\Delta = \sqrt[p]{(\Delta x)^p + (\Delta y)^p}$; thus $p = 1$, $p = 2$, and $p = \infty$ define the Manhattan, Euclidean, and Chebyshev norms, respectively). Such functionals include the MST cost and the MRST cost. Steele [22] has shown that optimal solutions to random n-point instances of these problems have expected cost $\beta\sqrt{n}$, where the constant β depends on both the problem, e.g., MRST versus MST, and the underlying L_p norm. The theory of subadditive functionals has many implications for VLSI CAD optimization. For example, several VLSI global routers (e.g., TimberWolfSC [18]) use the semiperimeter of a signal net bounding box as a computationally efficient MRST approximation. The growth function above immediately implies that this estimate can be refined by using an $O(\sqrt{n})$ scaling factor, with negligible CPU cost.

Fig. 1. MST (*left*) and MRST (*right*) for the same four-point set.

Fig. 2. An example where the cost ratio of a *separable* (in the terminology of Ho, Vijayan, and Wong) MST-derived-RST to the MRST is arbitrarily close to $3/2$. The MRST shown (*top*) has cost $(4/3) \cdot (n-1)$, while any RST derivable from the accompanying MST (*bottom*) has cost $2(n-2)$.
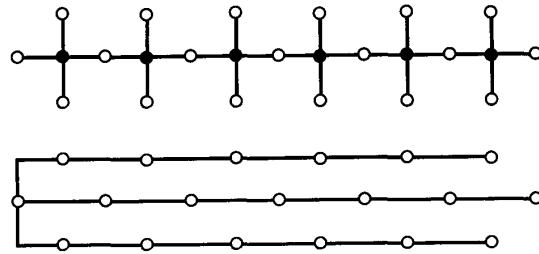
ling justification for MST-based MRST approximations. However, there are motivations for considering alternative approaches. Kahng and Robins [13] recently gave a class of examples which shows that the $3/2$ bound is tight for a large number of MST-based methods; i.e., the MST can be "unimprovable" (see Fig. 2). Thus, it seems unlikely that an MST-inspired heuristic will have performance ratio strictly less than $3/2$.

Furthermore, although the MST and MRST may have "similar" growth rates, an MST-derived solution method may not be appropriate for VLSI routing applications. It has been shown [2], [7] that the optimal Steiner tree, as well as heuristic MST-based RST's, will have a linear expected number of Steiner points. However, in certain board wiring technologies or for IC manufacturing and reliability considerations, having so many Steiner points may not be desirable. In fact, we would like to *prescribe* the relative incidence of Steiner points as a routing parameter that is a function of technology, performance, or estimated layout congestion, but this is not a natural concept when we use an MST-based method.

When we consider the extreme case where extra vias are very expensive, it is natural to ask the following: If we are allowed to introduce *exactly one Steiner point* into a net, where should it be placed? This is the motivation for the *iterated 1-Steiner* heuristic, which repeatedly finds the best possible Steiner point and adds it to the point set until no further improvement is possible. The purpose of this paper is to introduce the iterated 1-Steiner method along with several variations. We find that this new approach has a number of advantages:

- The performance ratio of the method is *never* as bad as $3/2$, we prove that it is not greater than $4/3$ on the entire class of instances for which $c(MST)/c(MRST) = 3/2$, while other known methods have performance ratio arbitrarily close to $3/2$.
- The average performance of the method is significantly better than all previous MST-based methods, yielding an average improvement of 10% to 11% over MST costs.
- We can limit the algorithm so that it introduces only k Steiner points (e.g., in a layout regime where vias are expensive).
- The method can be efficiently implemented by applying elegant computational geometry results, in-

cluding those of Georgakopoulos and Papadimitriou [6].

- There are many useful extensions, including randomized, batched, and parallel variations, as well as applications to alternate routing geometries.

In the following section, we review several important attributes of Steiner trees and develop the new method in detail. In Section III, we discuss variants and extensions of the method, while Section IV presents a large body of empirical results. The paper concludes by listing directions for further research.

II. THE ITERATED 1-STEINER APPROACH

We denote the minimum spanning tree over a point set P by $MST(P)$, and use $c(MST(P))$ to denote the cost of the MST on point set P. Given a point set $P = \{p_1, \cdots, p_n\}$, a *1-Steiner point* is any point x such that $c(MST(P \cup \{x\}))$ is minimized, with $c(MST(P \cup \{x\})) < c(MST(P))$. A *1-Steiner tree* is the minimum spanning tree over $P \cup \{x\}$.

Recall that a Steiner tree is an MST on the union of the original point set P and a set of Steiner points S. Our approach is to iteratively calculate optimum 1-Steiner points and include them into S. The length of the MST over $P \cup S$ will decrease with each additional point, and we terminate the construction if there is no x such that $c(MST(P \cup S \cup \{x\})) < c(MST(P \cup S))$. Fig. 4 illustrates the execution of iterated 1-Steiner on a four-point example.

The iterated 1-Steiner algorithm is thus stated as follows:

$S = \emptyset$
While $|S| < n$ and $\exists$ 1-Steiner point x **Do** $S = S \cup \{x\}$

By the result of Hanan, we can find a 1-Steiner point by constructing a new MST on $n+1$ points for each element in the Steiner candidate set, then picking the candidate which results in the shortest MST. Each MST com-
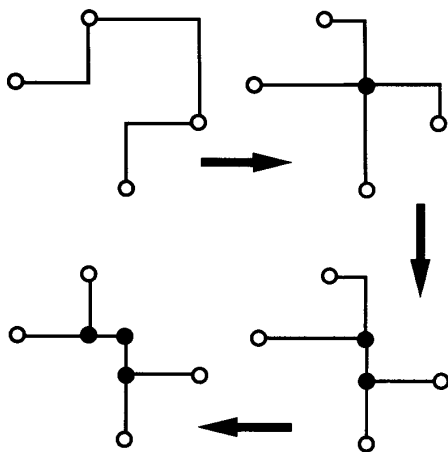
Fig. 3. Execution of iterated 1-Steiner on a four-point example.

putation can be performed in $O(n \log n)$ time [15], yielding an $O(n^3 \log n)$ time bound. Note that this is the time required to find just *one* 1-Steiner point, and that the Steiner tree may contain up to $n - 2$ Steiner points [7].

As it turns out, a new 1-Steiner point may be added in $O(n^2)$ time, as described below. A linear number of Steiner points can thus be found efficiently with a total of $O(n^3)$ effort, and finding heuristic solutions with $\leq k$ Steiner points, in a regime with high via costs, can be accomplished in $O(kn^2)$ time.

There are numerous extensions to the iterated 1-Steiner approach, including randomized variants and a very useful amortization of the 1-Steiner point computation which adds an entire set of "independent" Steiner points in a single iteration. Before we discuss these and other versions of this iterative construction, we review the $O(n^2)$ method for finding an 1-Steiner point and discuss the performance ratio of the 1-Steiner approach.

A. Finding 1-Steiner Points Efficiently

Georgakopoulos and Papadimitriou [6] give an $O(n^2)$ method for computing a 1-Steiner *tree* (not just a 1-Steiner point) for n points in the Euclidean plane. We use a direct adaptation of their method for the Manhattan norm. This idea is summarized as follows:

- A point p cannot have two neighbors in the MST which lie in the same octant of the plane with respect to p. Thus we can fix eight "orientations" at 45° intervals, each of which induces a Voronoi-like partition (the *oriented Dirichlet cells*) of the plane.
- These eight partitions can be overlaid into a "coarsest common partition" within $O(n^2)$ time. The resulting $O(n^2)$ regions of this partition are *isodendral:* introducing any point from within a given region will result in a constant MST *topology.*

- The minimum spanning tree on the n points is constructed, and we perform preprocessing in $O(n^2)$ time such that whenever a new point is added to the point set, updating the MST to include the new point requires *constant* time.
- We then iterate through the $O(n^2)$ regions of the overlaid partitions and determine, in constant time per region, the optimal Steiner point in each region. Each such point will induce an MST on $n + 1$ points that can be computed in constant time using the information obtained from the preprocessing. Comparing the costs of these trees and selecting the smallest one will give the minimum-length MST on $n + 1$ points. The total time for all phases is $O(n^2)$.

There are at most n iterations, each requiring $O(n^2)$ computation; therefore the time complexity of this method is $O(n^3)$. Empirical results in Section III show that iterated 1-Steiner significantly outperforms all existing heuristics (see Table I). We observe that the actual number of iterations the algorithm performs for random pointsets is less than $n/2$ on average.[2]

In surveying the vast Steiner tree literature, it seems that the closest conceptual relative of the iterated 1-Steiner heuristic is a method proposed by Smith and Liebman [19], [20], which involves a highly ad hoc examination of a linear-size subset of the candidate Steiner set. Our method seems preferable for several reasons: (1) *performance:* the method in [19] gives less than 8% average improvement over MST length for random point sets [12] and thus seems to fall in with the other methods in the literature, while our method gives almost 11% average improvement[3]; (2) *efficiency:* [19] gives an $O(n^4)$ method, while the iterated 1-Steiner algorithm is $O(n^3)$; (3) *simplicity:* the algorithm in [19] requires seven pages to describe while our method is simply described and coded.

B. Performance Ratio

Several results can be proved which bound the error of the iterated 1-Steiner method. A main result is that the output of iterated 1-Steiner can never be as bad as $3/2$ times optimal. We prove our bound as follows. First, we completely characterize the class of instances (the "unions of pluses") for which the ratio $c(MST)/c(MRST) = 3/2$, which is a result of independent interest. We then show that the iterated 1-Steiner algorithm will always find a 1-Steiner point on such instances, whereas previous methods may fail to find any improvement over the MST. Finally, we show that on this class of "difficult" instances,

---

[2]There are examples where as many as $n - 1$ iterations are performed. Thus, our method can generate more Steiner points than would exist in the optimal MRST, although we can easily enforce the $n - 2$ bound by removing degree-2 and degree-1 Steiner points without increasing the tree cost.

[3]Recently, the method of [3] has been reported to also yield about 11% average improvement over MST.

TABLE I
STEINER TREE HEURISTIC STATISTICS

# Pts	Corner Flipping			Prim			Iterated 1-Steiner					
	Min Perf.	Ave Perf.	Max Perf.	Min Perf.	Ave Perf.	Max Perf.	Min Perf.	Ave Perf.	Max Perf.	Min #SP	Ave #SP	Max #SP
3	0.00	6.60	24.93	0.00	4.10	24.93	0.00	6.94	24.93	0	0.66	1
4	0.00	7.85	29.11	0.00	5.71	29.11	0.00	8.73	29.11	0	1.09	3
5	0.00	8.09	25.07	0.00	6.20	25.07	0.00	9.41	25.07	0	1.59	4
6	0.00	8.16	27.00	0.00	6.34	22.56	0.00	9.74	27.00	0	2.03	5
7	0.00	8.12	23.15	0.00	6.45	23.24	0.00	9.99	23.80	0	2.52	5
8	0.00	8.20	22.57	0.00	6.52	19.97	0.31	10.10	25.46	1	2.96	6
9	0.37	8.27	20.94	0.00	6.57	19.16	0.80	10.19	22.57	1	3.42	7
10	0.39	8.20	19.29	0.00	6.45	17.34	0.39	10.15	22.62	1	3.83	7
12	1.02	8.20	19.27	0.00	6.44	18.40	1.50	10.26	19.93	2	4.73	8
14	1.93	8.25	18.04	0.12	6.46	16.09	2.45	10.34	21.77	2	5.60	9
16	2.46	8.24	16.65	0.41	6.48	14.34	2.95	10.43	19.90	3	6.61	10
18	3.07	8.15	16.12	0.72	6.60	13.24	3.61	10.35	17.19	4	7.42	11
20	3.26	8.23	14.21	0.84	6.31	11.89	5.40	10.52	16.25	5	8.35	13
25	3.10	8.37	13.70	1.81	6.47	13.01	4.07	10.64	15.62	6	10.59	15
30	3.51	8.44	12.70	2.19	6.67	11.20	5.00	10.90	15.60	9	12.80	17
35	4.98	8.35	13.18	2.85	6.53	11.32	6.46	10.74	15.32	10	15.04	20
40	4.35	8.50	12.78	2.89	6.68	11.40	5.78	10.93	14.76	12	17.37	22

# Pts	Iterated Random 1-Steiner						Meta Heuristic					
	Min Perf.	Ave Perf.	Max Perf.	Min #SP	Ave #SP	Max #SP	Min Perf.	Ave Perf.	Max Perf.	Min #SP	Ave #SP	Max #SP
3	0.00	6.94	24.93	0	0.66	1	0.00	6.94	24.93	0	0.66	1
4	0.00	8.70	29.11	0	1.20	3	0.00	8.73	29.11	0	1.09	3
5	0.00	9.15	25.07	0	1.72	4	0.00	9.45	25.07	0	1.60	4
6	0.00	9.35	27.00	0	2.23	4	0.00	9.80	27.00	0	2.07	5
7	0.00	9.42	23.80	0	2.76	5	0.00	10.07	23.80	0	2.57	5
8	0.00	9.36	25.46	1	3.22	6	0.31	10.19	25.46	1	3.02	6
9	0.00	9.38	22.14	1	3.73	7	0.80	10.28	22.57	1	3.49	7
10	3.30	9.24	19.99	1	4.18	7	0.39	10.25	22.62	1	3.94	7
12	1.26	9.16	19.93	1	5.11	9	1.50	10.35	19.93	2	4.86	9
14	0.00	8.93	21.73	1	5.98	11	2.45	10.42	21.77	3	5.76	10
16	0.00	8.85	19.35	1	6.81	11	3.01	10.51	19.90	3	6.78	11
18	0.09	8.78	15.99	1	7.71	12	4.09	10.43	17.19	4	7.63	11
20	0.04	8.61	15.98	1	8.50	14	5.40	10.60	16.25	5	8.58	14
25	0.42	8.21	14.39	1	10.34	17	4.07	10.67	15.62	6	10.83	16
30	0.16	8.44	15.00	1	12.50	19	5.00	10.90	15.60	10	13.11	18
35	0.44	7.75	14.82	1	13.56	24	6.46	10.78	15.32	12	15.31	22
40	0.22	7.90	13.64	1	16.00	26	5.78	10.97	14.76	12	17.78	26

the iterated 1-Steiner method actually has performance bound $\leq 4/3$, significantly better than previous methods.

Lemma 1: Any point set P with $|P| \leq 3$ has $c(MST)/c(MRST) \leq 4/3$.

Proof: For $|P| = 2$, $c(MST)/c(MRST) = 1$. For $|P| = 3$ we have $c(MRST) = R/2$, where R is the perimeter of the bounding box of P. On the other hand, we observe that $c(MST) \leq 2R/3$. It follows that $c(MST)/c(MRST) \leq (2R/3)/(R/2) = 4/3$. □

Definition: A *plus* is a Steiner tree over four points having coordinates of the form $\{(x - r, y), (x + r, y), (x, y - r), (x, y + r)\}$; a plus has exactly one Steiner point at (x, y), the *midpoint* of the plus.

Lemma 2: A plus is the only configuration of four points that achieves a ratio $c(MST)/c(MRST)$ of *exactly* $3/2$ using exactly one Steiner point.

Proof: If a four-point configuration has exactly one

Steiner point in its MRST, its topology is the unique one depicted in [11, Fig. 5(a)] (i.e., that of a plus); thus the point set must have coordinates of form $P = \{(x - h_1, y), (x + h_2, y), (x, y - v_1), (x, y + v_2)\}$. Again, let R be the perimeter of the bounding box of P. The MRST for P has cost exactly equal to $R/2$, while the MST has cost at most $R - R/4$ since we can obtain a spanning tree by deleting the largest of the four edges which make up the bounding box, and this edge must have cost at least $R/4$. This implies that $c(MST)/c(MRST) \leq 3/2$, with equality holding only when the largest edge around the bounding box is not greater than $R/4$, i.e., when all four edges around the bounding box are of equal length. Therefore, $h_1 = h_2$ and $v_1 = v_2$. We write $h = h_1 = h_2$ and $v = v_1 = v_2$, and assume without loss of generality that $h \leq v$. We now have

$$\frac{c(MST)}{c(MRST)} = \frac{2(v + h) + 2h}{2(v + h)} = 1 + \frac{h}{v + h} \leq \frac{3}{2}$$

with equality holding when $h = v$, implying that the configuration is indeed a plus. □

Definition: A *union of pluses* is a Steiner tree with k Steiner points over a point set P with $|P| = 3k + 1$, where each Steiner point has degree 4 and all four edges incident to any Steiner point are of equal length.

Theorem 1: Any point set having $c(MST)/c(MRST) = 3/2$ has an MRST which is a union of pluses.

Proof: Following the proof of the result in Hwang [11], note that for any point set P there is an optimal Steiner tree composed of connected components, each of which has all of its Steiner points forming a chain. Without loss of generality, all of the Steiner points on such a chain are collinear, with the possible exception of the Steiner point at the end of the chain. Using the same upper bound for MST cost and the exact expression for MRST cost as in [11], we can equate expressions for $(2/3) \cdot c(MST)$ and $c(MRST)$ for the points of any chain:

$$R \cdot (\tfrac{1}{2} + \tfrac{2}{3} \cdot \Theta) = R \cdot (\tfrac{1}{2} + \Theta),$$

where R is again the length of the bounding box of the points in the chain, and Θ is defined so that $R \cdot \Theta$ is equal to the sum of the distances from all (except the last) of the original points to their adjacent Steiner points in the chain. The equality above implies that $\Theta = 0$ and thus all but one of the original points have the same coordinates as their adjacent Steiner points, a contradiction unless there is only one Steiner point (i.e., the last) in this chain. We already know from Lemma 2 that any chain which has only one Steiner point and which exactly achieves the $3/2$ ratio must be a plus. It follows that any optimal Steiner tree which exactly achieves the $3/2$ ratio must be decomposable into pluses; i.e., it is a union of pluses. □

Theorem 1 completely characterizes the point sets for which $c(MST)/c(MRST)$ is *exactly* equal to $3/2$. Using this, we show the following:

Theorem 2: The performance ratio of iterated 1-Steiner is always less than $3/2$.

Proof: If a point set has $c(MST)/c(MRST) < 3/2$ then even if iterated 1-Steiner does not find *any* Steiner points, its performance ratio will be less than $3/2$. From Theorem 1, we know that any point set for which $c(MST)/c(MRST) = 3/2$ will have an MRST that is a union of pluses; in this case iterated 1-Steiner will certainly select and add the midpoint of some plus at the first iteration; hence the performance ratio will be less than $3/2$ overall. To see this, note that a spanning tree with cost $3/2$ times $c(MRST)$ is found by simply replacing every plus in the MRST by an arbitrary tree on the four endpoints of the plus, as shown in Fig. 4. Adding the midpoint of the plus as a Steiner point will reduce the cost of connecting these four endpoints, and the midpoint is
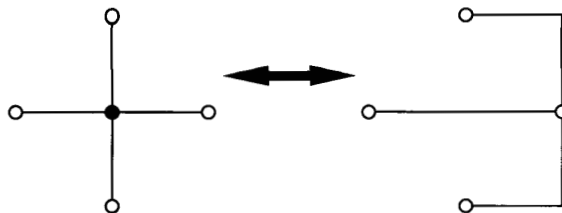

Fig. 4. Locally replacing each plus (*left*) with an MST (*right*).

indeed one of the candidates considered during the first iteration of the iterated 1-Steiner algorithm.[4] □

Notice that existing MST-based methods will have performance ratio arbitrarily close to $3/2$ on unions of pluses, as shown by the example of Fig. 2. In contrast, we may show the following good performance bound for the 1-Steiner method:

Theorem 3: The performance ratio of iterated 1-Steiner on instances whose MRST's are unions of pluses is always $\leq 4/3$.

Proof: When iterated 1-Steiner selects a midpoint of a plus, at most three midpoints of other pluses may be excluded from future selection. By the selection rule of the algorithm, the three midpoints that are possibly excluded cannot belong to pluses larger than the one selected. Thus if iterated 1-Steiner selects a plus that is not in the optimal MRST, the savings will be at least as great as the savings that would have been realized by selecting the largest of the (up to three) pluses that are now excluded by topological constraints, as shown in Fig. 5.

Each plus represents a savings of $1/3$ of the MST cost over the endpoints of the plus, so even if we use simple MST edges to connect the remaining affected points to the selected plus, the total heuristic cost is no more than $c(MST) - (1/3) \cdot (1/3) \cdot c(MST) = (8/9) \cdot c(MST)$. Therefore, the performance ratio of iterated 1-Steiner is no greater than $(\tfrac{8}{9} \cdot c(MST))/(\tfrac{2}{3} \cdot c(MST))/ = 4/3$. □

We note that this bound can probably be tightened by more exhaustive case analysis. Since most nets of practical size have fewer than six terminals, we now briefly discuss performance bounds for small nets.

Theorem 4: The iterated 1-Steiner heuristic is optimal for four or fewer points.

Proof: For three points, there can be at most one Steiner point, and since our heuristic examines all candidates, it is optimal. For a set of four points, the MRST can have zero, one, or two Steiner points, and our method is trivially optimal when this number is less than 2. When the MRST has two Steiner points, it must have one of the two topologies shown in Fig. 6 [11]. A simple case anal-

[4]Even if there are other candidates within the convex hull of the four points of the plus, the midpoint trivially gives the greatest possible savings since it achieves a cost improvement of exactly one third.
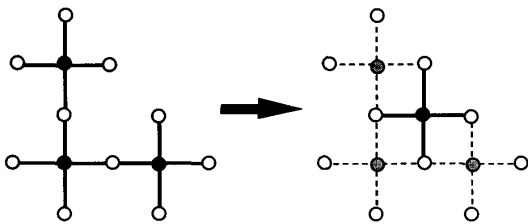
Fig. 5. Each selected 1-Steiner point may exclude at most three potential 1-Steiner points from future selection; thus at least one third of the maximum possible savings is achieved.
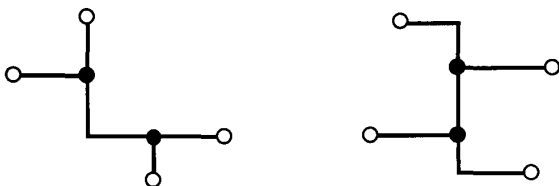

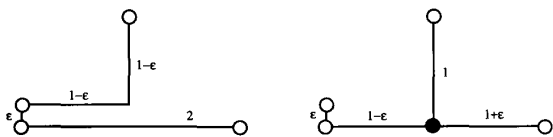Fig. 6. The two possible Steiner tree topologies on four points.

Fig. 7. A four-point example where MST-improvement algorithms perform arbitrarily close to 4/3 times optimal (*left*); in contrast, iterated 1-Steiner performs optimally on all point sets of size 4 or less (*right*).
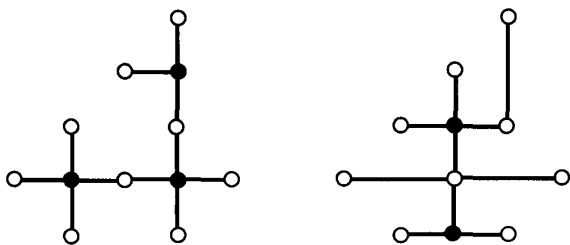

Fig. 8. A nine-point example where the iterated 1-Steiner performance ratio is 13/11; the optimal MRST (*left*) has cost 11, while the (possible) heuristic output (*right*) has cost 13.

ysis shows that our heuristic always selects both Steiner points, with order of selection irrelevant.   □

In contrast, MST-based methods are generally not optimal even for four-point nets, as shown by the example in Fig. 7. We have found a nine-point example where the 1-Steiner heuristic performs as badly as 13/11 times optimal (Fig. 8). After considerable effort we have not found any instance for which iterated 1-Steiner has a performance ratio worse than 13/11.

It is encouraging that while five- or six-point examples exist which *force* a performance ratio of 3/2 for other MRST heuristics in the literature, the worst-case performance ratio of iterated 1-Steiner for a five-point example seems to be only 7/6 (Fig. 9). We conjecture that the iterated 1-Steiner method has a performance ratio uniformly bounded away from 3/2; i.e., there exists a pos-

Fig. 9. A five-point example where the iterated 1-Steiner performance ratio is 7/6. The optimal MRST (*left*) has cost 6, while the (possible) heuristic output (*right*) has cost 7.

itive constant ϵ such that iterated 1-Steiner will never have performance ratio greater than $3/2 - \epsilon$ on any instance.[5]

III. 1-STEINER VARIANTS

A. A Random Variant

An important variant of iterated 1-Steiner is motivated by observing that it may not be necessary to find the *best* candidate Steiner point at each iteration. In particular, the quality of the final tree might be acceptable even if each step simply chooses a *random* improving point. For both this method and the original heuristic, we may simplify the output by removing Steiner points that become degree-1 or degree-2 points in subsequent MST's; by the triangle inequality the latter can be removed without increasing the MST cost, and the former can trivially be removed. The advantage of this refinement is that performance is not affected, while the final layout is guaranteed to have at most $n - 2$ Steiner points. The *iterated random 1-Steiner* heuristic is:

$$S = \emptyset$$
While ∃ 1-Steiner points **Do**
$\quad S = S \cup \{$ a random improving Steiner point $\}$
$\quad$ Remove elements of S with degree ≤ 2 in $MST(P \cup S)$

Iterated random 1-Steiner lends itself well to a simple, compact implementation. Performance is worse than iterated 1-Steiner, but remains slightly better than MST-derived solutions for typical instances. Iterated random 1-Steiner will clearly terminate because the MST length decreases monotonically, but the cost can take on any of an exponential number of distinct values for certain instances (intuition suggests that there is a polynomial expected upper bound on the number of iterations). A variant which requires that a point cannot return to the layout after it has been deleted will have a trivial $O(n^2)$ bound on the number of iterations, and we can construct a family of instances for which the randomized method actually produces this quadratic number of Steiner points.

B. A Batched Variant

Perhaps the most promising variant involves amortization of the computational expense, as follows. We use the approach of [6] to compute an optimal 1-Steiner point and its associated MST cost savings within each isodendral region, but instead of selecting only a Steiner candidate which has highest cost savings, we select a maximal ''in-

dependent'' set of Steiner points, similar to the approach of [9]. The criterion for independence is that no candidate Steiner point is allowed to interfere with, i.e., reduce, the potential MST cost savings of any other candidate in the added set. In particular, for a set of points P, 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 $\Delta MST(P, S) = max(0, MST(P) - MST(P \cup S))$ and where $\Delta MST(P, \{x\}) > 0$ for x to be a candidate Steiner point. A round of this method is formally described as follows:

- Compute the MST over P in $O(n \log n)$ time using a Voronoi diagram-based method [15]. Also construct the weighted undirected graph $G = (P, E)$, where $E = \{(x, y) \mid (x, y)$ is an edge in the Delaunay triangulation over $P\}$ and the weight of each edge in E is the rectilinear distance between its two endpoints.
- Compute the $O(n^2)$ isodendral regions over P, and for each region determine the $O(1)$ potential neighboring points in the MST as in [6]. This requires a total of $O(n^2)$ time.
- Preprocess the $O(n^2)$ isodendral regions, now treated as a planar subdivision, so that future planar subdivision searches (i.e., determining the planar region in which a given point lies) may be performed in $O(\log n)$ time [15]. This preprocessing requires $O(n^2 \log n)$ time, using $O(n^2 \log n)$ space.
- For each candidate Steiner point x, compute the cost savings $\Delta MST(P, \{x\})$ associated with x. We determine the isodendral region to which x belongs in $O(\log n)$ time via planar subdivision search, and let X be the set of potential MST neighbors of x. For each subset $Y \subseteq X$ we add the weighted edge set $\{(x, y) \mid y \in Y\}$ to the graph G (recall that the weight of each edge is the rectilinear distance between x and y). The MST of a planar weighted graph can be maintained using $O(\log n)$ time per addition/insertion of a point or edge [4]. Since $|X| = O(1)$ and therefore $|Y| = O(1)$, we can determine in $O(\log n)$ time the MST cost savings for each candidate Steiner point. By Hanan's theorem there are at most n^2 candidate Steiner points; therefore the time for this entire phase is $O(n^2 \log n)$.
- Next, sort the $O(n^2)$ Hanan candidates in order of decreasing MST cost savings; this requires $O(n^2 \log n)$ time using any efficient sorting algorithm.
- Determine a maximal set of independent candidate Steiner points to be added during this round, by successively adding candidates in order of decreasing MST cost savings, as long as each added Steiner point is independent of all Steiner points previously added during the round. In other words, for an original point set P, a set of already added candidate points S, and a new candidate x, add x to S if and only if $\Delta MST(P, \{x\}) \leq \Delta MST(P \cup S, \{x\})$. Again, MST cost savings arising from the addition

or deletion or a single point can be determined in time $O(\log n)$ [4], bringing the total time for this entire step to $O(n^2 \log n)$.

We iterate such rounds with $P = P \cup S$ until we reach a round which fails to add at least one Steiner point to P. The total time required for each round is $O(n^2 \log n)$. Given a point set P, the *batched 1-Steiner algorithm* is summarized as follows:

While $\exists$ set $S = \{x \mid \Delta MST(P, \{x\}) > 0\} \neq \emptyset$ **Do**
  For $x \in S$ in order of decreasing ΔMST **Do**
    If $\Delta MST(P - S, \{x\}) \leq \Delta MST(P, \{x\})$
      Then $P = P \cup \{x\}$

Empirical data indicate that the number of rounds required grows much more slowly than the number of Steiner points produced. For example, experimental results on point sets of size 40 show an average number of about 17 Steiner points produced (with a maximum of 22), while the average number of rounds for batched 1-Steiner is only 2.05 (with a maximum of 4). We conjecture that the number of rounds grows only sublinearly as a function of $|P|$.

IV. COMPUTATIONAL RESULTS

We coded the iterated 1-Steiner, the iterated random 1-Steiner, and batched 1-Steiner heuristics, along with several existing methods, using ANSI C in both Sun-4 and Apple Macintosh environments. The code is available from the authors upon request.

Extensive performance comparisons contrasted iterated 1-Steiner and random 1-Steiner with the standard Corner and Prim methods described below. 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 1000 × 1000 grid; such instances are statistically indistinguishable from the pin locations of actual cell-based layouts, and they are in fact the standard test-bed for Steiner tree heuristics [16]. The results are summarized in Table I, and are depicted graphically in Fig. 10. Table II gives results demonstrating that batched 1-Steiner is as effective as iterated 1-Steiner.

A. Incremental Calculations

These tables also show that even when restricted to a k-point or k-round solution, both the iterated and the batched 1-Steiner algorithms still perform well, with a large portion of the wire-length savings (as a percentage improvement over the MST cost) occurring in the first several iterations/rounds. Because of this, it seems reasonable for a layout system to use our method for "k-Steiner point routing"; this will be accomplished in $O(kn^2)$ time and the parameter k can reflect via costs, routing congestion, performance, and other manufacturability or reliability attributes. Similar arguments can be made for a k-round implementation of the batched 1-Steiner variant, which will take $O(kn^2 \log n)$ time. For batched

TABLE II
AVERAGE IMPROVEMENT PERCENTAGES PER ROUND FOR BATCHED 1-STEINER

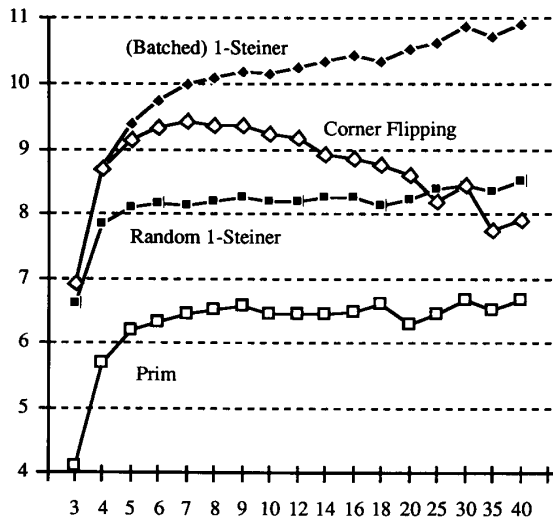| # pts	# of Rounds			improvement per round				improvement per Steiner point									
	min	ave	max	1	2	3	4	1	2	3	4	5	6	7	8	9	10
3	0	0.67	1	7.66	0.00	0.00	0.00	7.66	0.00								
4	0	0.96	2	8.83	0.18	0.00	0.00	8.39	0.62	0.00							
5	0	1.07	4	9.30	0.30	0.00	0.00	7.78	1.72	0.10	0.00						
6	0	1.15	4	9.48	0.38	0.01	0.00	7.06	2.37	0.42	0.03	0.00					
7	0	1.19	5	9.61	0.40	0.01	0.00	6.34	2.73	0.82	0.11	0.01	0.00				
8	0	1.24	4	9.75	0.45	0.01	0.00	5.90	2.83	1.15	0.30	0.04	0.00				
9	1	1.28	5	9.75	0.47	0.02	0.00	5.38	2.87	1.36	0.50	0.12	0.01	0.00			
10	1	1.33	6	9.82	0.49	0.02	0.00	5.01	2.83	1.51	0.69	0.24	0.04	0.00			
12	1	1.40	4	9.79	0.48	0.02	0.00	4.33	2.71	1.65	0.91	0.46	0.20	0.04	0.00		
14	1	1.48	5	9.87	0.53	0.02	0.00	3.87	2.56	1.70	1.07	0.63	0.37	0.17	0.05	0.01	0.00
16	1	1.56	4	9.90	0.54	0.02	0.00	3.44	2.39	1.71	1.17	0.77	0.50	0.31	0.15	0.04	0.01
18	1	1.61	4	9.85	0.54	0.03	0.00	3.16	2.23	1.65	1.20	0.83	0.56	0.39	0.26	0.11	0.03
20	1	1.65	4	9.81	0.58	0.03	0.00	2.88	2.09	1.59	1.22	0.89	0.62	0.43	0.25	0.21	0.12
25	1	1.77	5	9.97	0.52	0.03	0.00	2.45	1.84	1.46	1.18	0.94	0.74	0.57	0.41	0.30	0.24
30	1	1.93	4	10.14	0.69	0.03	0.01	2.17	1.68	1.36	1.12	0.93	0.78	0.62	0.51	0.39	0.31
35	1	2.00	4	10.09	0.58	0.02	0.00	1.82	1.44	1.23	1.07	0.92	0.79	0.67	0.54	0.45	0.38
40	1	2.05	4	9.80	0.55	0.04	0.01	1.74	1.12	0.94	0.81	0.71	0.62	0.56	0.48	0.43	0.37

Fig. 10. Performance comparison of the heuristics; the horizontal axis represents the number of points per set, while the vertical axis represents percentage cost improvement over MST.

TABLE III
META(CORNER, PRIM) OUTPERFORMS ITS COMPONENT HEURISTICS

# Pts	Corner Ave %	Prim Ave %	Meta Ave %
5	8.022	6.162	8.580
10	8.155	6.455	8.584
15	8.352	6.548	8.613
20	8.240	6.392	8.424

TABLE IV
1-STEINER DOMINATES BOTH CORNER AND PRIM

# Pts	Corner Ave %	Prim Ave %	1-Steiner Ave %	Meta Ave %
10	8.18	6.54	10.23	10.26
12	8.16	6.30	10.25	10.28
15	8.19	6.54	10.33	10.35
17	8.16	6.43	10.38	10.39
18	8.25	6.48	10.51	10.52
22	8.29	6.49	10.45	10.46
25	8.38	6.53	10.65	10.66

1-Steiner, the advantages of incremental calculation are dramatic: on 40-point instances, about 95% of the total improvement occurs in the first round, and over 99% of the improvement occurs in the first two rounds. Results detailing the nature of the incremental improvements are given in Table II.

B. On Metaheuristics

For a number of combinatorial problems, the following concept of a *metaheuristic* is natural. Given an instance of a problem and m different heuristics (algorithms) H_1, H_2, $\cdots$, H_m, the metaheuristic $H_{meta}(H_1, H_2, \cdots, H_m)$ will output the best among the m outputs of heuristics H_1, H_2, $\cdots$, H_m. Intuitively, several methods can trade off in their "areas of expertise", so while the metaheuristic is asymptotically of the same time complexity as the slowest component heuristic, the approximation perfor-

mance is typically better than the performance of any single method.

To illustrate this phenomenon, we give computational results from implementations of *Corner* (from "corner-flipping"; this method gives results similar to the method of [10]) and *Prim*, a simple analog of Prim's MST heuristic construction that is similar to MRST heuristics analyzed in [16]. Table III shows that Corner and Prim, when used together, give an average performance of about half a percent better than Corner alone, although the average performance of Prim is about 2% worse than that of Corner. In contrast, the metaheuristic H(Prim, Corner, 1-Steiner) gives essentially the same performance as 1-Steiner alone, implying that 1-Steiner strictly dominates the other methods (Table IV). This is a very important aspect: it suggest that the 1-Steiner method will universally give "reasonably good" solutions.

The metaheuristic 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 the 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.

V. EXTENSIONS AND CONCLUSIONS

In this paper, we have presented a fast new approach to the rectilinear Steiner problem. The method yields results that reduce wire length by up to 2% to 3% over the best previous methods. Furthermore, it is the first heuristic which has been shown to have a performance ratio less than 3/2; in fact, the performance ratio is less than or equal to 4/3 on the entire class of instances where the ratio $c(MST)/c(MRST)$ is exactly equal to 3/2. The algorithm has practical asymptotic complexity owing to an elegant implementation which uses methods from computational geometry and which parallelizes readily. A randomized variant of the algorithm, along with a batched variant, has also proved successful.

We further note that the approach extends easily to routing formulations for newer technologies. We observe that several technological trends lead to three-dimensional global routing formulations. The previous standard heuristic approach, i.e., improving an initial MST solution, becomes much more difficult 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. To this end, we briefly mention several advantages of our approach in higher dimension. It is not difficult to see that Hanan's theorem still holds in all higher dimensions [21], and we have conjectured [13] that the obvious generalization of Hwang's theorem holds in d-space, i.e.,

$$c(MST) \leq \frac{2d - 1}{d} \cdot c(MRST).$$

There is an infinite family of higher-dimensional point sets for which our iterated 1-Steiner scheme performs optimally yet all other MST-based heuristics can perform as badly as $(2d - 1)/d$ times optimal in d dimensions, which is no better than the MST length for the same point sets [13]. Furthermore, Theorem 3 can be generalized to arbitrary dimension d, where the performance of iterated 1-Steiner would be no worse than $(4d^2 - 5d + 2)/(d(2d - 1))$, e.g., for $d = 3$ we obtain a worst-case bound of 23/15 for "difficult" point sets which have an MRST that is a union of "pluses." Empirical results for three-dimensional problem instances seem quite favorable. The 1-Steiner approach also succeeds in the presence of nonorthogonal wiring directions [17], and Hanan's result also generalizes to such geometries.

We conclude by noting two avenues for future research: (i) the concept of a metaheuristic, introduced in Section IV, which may be effective in addressing other optimizations, and (ii) the unknown performance ratio of the iterated 1-Steiner method, which we conjecture to be uniformly bounded away from 3/2 (see Footnote 5).

REFERENCES

[1] J. Beardwood, H. J. Halton, and J. M. Hammersley, "The shortest path through many points," in Proc. Cambridge Philos. Soc., vol. 55, pp. 299–327, 1959.
[2] M. W. Bern, "Two probabilistic results on rectilinear Steiner trees," Algorithmica, vol. 3, pp. 191–204, 1988.
[3] T. H. Chao and Y. C. Hsu, "Rectilinear Steiner tree construction by local and global refinement," in Proc. IEEE Int. Conf. Computer-Aided Design (Santa Clara), Nov. 1990, pp. 432–435.
[4] D. Eppstein et al., "Maintenance of a minimum spanning forest in a dynamic planar graph," in Proc. ACM-SIAM Symp. Discrete Algorithms, 1990, pp. 1–11.
[5] M. Garey and D. S. Johnson, "The rectilinear Steiner problem is NP-complete," SIAM J. Appl. Math., vol. 32, no. 4, pp. 826–834, 1977.
[6] G. Georgakopoulos and C. H. Papadimitriou, "The 1-Steiner tree problem," J. Algorithms, vol. 8, pp. 122–130, 1987.
[7] E. N. Gilbert and H. O. Pollak, "Steiner minimal trees," SIAM J. Appl. Math., vol. 16, pp. 1–29, 1968.
[8] M. Hanan, "On Steiner's problem with rectilinear distance," SIAM J. of Appl. Math., vol. 14, pp. 255–265, 1966.
[9] N. Hasan, G. Vijayan, and C. K. Wong, "A neighborhood improvement algorithm for rectilinear Steiner trees," in Proc. ISCAS, 1990.
[10] 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.
[11] F. K. Hwang, "On Steiner minimal trees with rectilinear distance," SIAM J. Appl. Math., vol. 30, no. 1, pp. 104–114, 1976.
[12] F. K. Hwang, "The rectilinear Steiner problem," J. Design Automation and Fault-Tolerant Computing, pp. 303–310, 1978.
[13] A. Kahng and G. Robins, "On performance bounds for two rectilinear Steiner tree heuristics in arbitrary dimension," Tech. Rep. UCLA CSD-TR-900015, Apr. 1990.
[14] A. Kahng and G. Robins, "A new family of Steiner tree heuristics with good performance: The iterated 1-Steiner approach," in Proc. IEEE Int. Conf. Computer-Aided Design (Santa Clara), Nov. 1990, pp. 428–431.
[15] F. P. Preparata and M. 1. Shamos, Computational Geometry: An Introduction. New York: Springer-Verlag, 1985.
[16] D. Richards, "Fast heuristic algorithms for rectilinear Steiner trees," Algorithmica, vol. 4, pp. 191–207, 1989.
[17] M. Sarrafzadeh and C. K. Wong, "Hierarchical Steiner tree construction in uniform orientations," to be published.
[18] C. Sechen, VLSI Placement and Global Routing Using Simulated Annealing. Boston, MA: Kluwer Academic Publishers, 1988.
[19] J. M. Smith and J. S. Liebman, "Steiner trees, Steiner circuits and the interference problem in building design," Engineering Optimization, vol. 4, pp. 15–36, 1979.
[20] 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, vol. 4, pp. 179–192, 1980.
[21] T. L. Snyder, "A simple and faster algorithm for the rectilinear Steiner problem in general dimension," in Proc. ACM Symp. Computational Geometry, 1990.
[22] J. M. Steele, "Growth rates of Euclidean minimal spanning trees with power weighted edges," Annals of Probability, vol. 16, no. 4, pp. 1767–1787, 1988.
[23] P. Winter, "Steiner problem in networks: A survey," Networks, vol. 17, pp. 129–167, 1987.
[24] P. Berman and V. Ramaiyer, "Improved approximations for the Steiner tree problem," in Proc. ACM/SIAM Symp Discrete Algorithms, 1992.
[25] A. Z. Zelikovsky, "The 11/6 approximation algorithm for the Steiner problem on networks," Information and Computation, to be published.
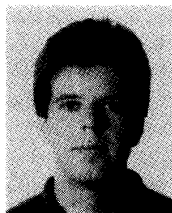
Andrew B. Kahng (A'89) was born in October 1963 in San Diego, CA. He holds the A.B. degree in applied mathematics and physics from Harvard College and the M.S. and Ph.D. degrees in computer science from the University of California at San Diego.

He has been an Assistant Professor in the Computer Science Department at UCLA since July 1989. His research interests include computer-aided design of VLSI circuits, combinational and parallel algorithms, global optimization theory, and computational geometry.

Dr. Kahng is a member of ACM and SIAM.

Gabriel Robins (S'90) is currently working toward the Ph.D degree in the Computer Science Department at UCLA, where he has won a Distinguished Teaching Award and currently holds an IBM Graduate Fellowship. He holds an M.S. from Princeton University and a B.S. from UCLA, also in computer science. His primary areas of research are VLSI CAD and geometric algorithms, with recent work focusing on performance-driven routing and heuristic Steiner tree constructions, as well as robot motion in planning and pattern recognition.

Mr. Robins is a member of ACM.