Constructing Exact Octagonal Steiner Minimal Trees

Chris S. Coulston
Penn State Erie, The Behrend College
Erie, PA 16563
coulston@psu.edu

ABSTRACT

Octagonal Steiner Minimal Trees (OSMTs) are used in the global routing phase of pervasive octagonal VLSI layout. The OSMT problem seeks a minimal length spanning structure using edges composed of line segments having one of four equally spaced orientations. The concept of a canonical form is introduced providing a strong framework for the structure and characteristics of OSMTs. An exact algorithm and a variety of pruning techniques are introduced. Random and OR Library instances are solved and compared against rectilinear and Euclidean SMTs. These experiments demonstrate the utility of pervasive octagonal routing, showing that octagonal SMTs are consistently 10% smaller than rectilinear SMTs.

Categories and Subject Descriptors

F.2.2 [**Nonnumerical Algorithms and Problems**]: Geometrical problems and computations; J.6 [**Computer-Aided Engineering**]: Computer-aided design

General Terms

Algorithms, Design, Theory

Keywords

Steiner Trees, Octagonal, Routing, Computer-Aided Design

1. INTRODUCTION

Octagonal edges are composed of line segments having one of four equally spaced orientations, see Figure 1A. The octagonal Steiner minimal tree (OSMT) problem seeks a minimal length set of octagonal edges spanning a set of N *given points*. In general, Steiner trees contain additional *Steiner points*, which are not explicitly specified, but whose inclusion reduces the length of the tree, see Figure 1B.

Octagonal routing in VLSI has two significant advantages. First, the amount of wire required to interconnect nets is reduced by "cutting corners". Given two points on opposite corners of a square, an octagonal router can use the diagonal while a rectilinear router must use two of the squares edges. Second, all things being equal, an octagonal wire of length 1 will cover 1.41 times more area than a rectilinear wire of the same length; giving the placer more flexibility in a parts position. The need for octagonal routing design tools is witnessed by the X-Initiative [14]; a consortium of companies pursuing the idea of pervasive octagonal routing. There are difficulties associated with octagonal routing. Foremost of these is the fact that gridded alignment is not feasible because it induces suboptimal spacing between adjacent wires in the diagonal directions [11], see Figure 2. This deficiency is minimized by removing the preferred direction constraint, allowing wires within a metal layer to travel in more than 1 direction. Using this so called "liquid routing" [11] technique, a Toshiba microprocessor core implemented with pervasive octagonal routing had a 20% reduction in wire length, 40% reduction in via count, and a 11% decrease in die size [11].

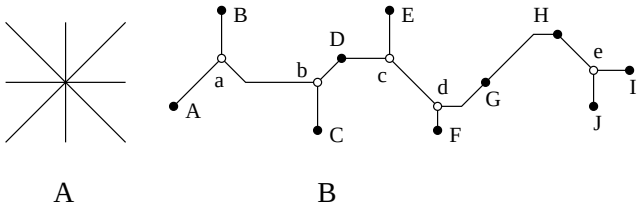

Figure 1: A) Legal octagonal line segment orientations. B) An octagonal Steiner tree on a set of 10 given points. Given points are black and Steiner points white.

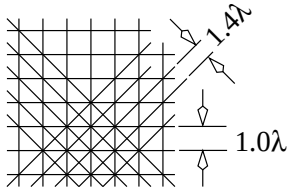

Figure 2: Gridded alignment induces suboptimal spacing in either the regular or diagonal routing layers.

There are two main contributions of this work. First, the space of feasible Steiner points is formalized in the definition of a canonical form. This definition provides a strong frame-

GLSVLSI'03, April 28–29, 2003, Washington, DC, USA.

work for the structural characteristics of OSMTs. Second, an exact OSMT algorithm is presented along with pruning techniques. Experiments are presented comparing the length and structure of rectilinear, octagonal, and Euclidean SMTs.

In the rest of this paper all angular measurements are represented in radians. In figures, given points are black, Steiner points white and points which can be either Steiner or given are shaded grey. Unless stated otherwise, all SMT references are assumed to refer to OSMTs.

2. PRELIMINARIES

The number of edges incident to a point is its degree. In a full SMT each given point has degree one and each Steiner point has degree at least three. A SMT can be considered as the edge disjoint union of full SMTs, called the full components [7]. For example, the OSMT in Figure 1B is composed of four full components, $\{A, B, C, D, a, b\}$, $\{D, E, F, G, c, d\}$, $\{G, H\}$, $\{H, I, J, e\}$.

An *edge* is a shortest collection of straight line segments which connects two points. A *crooked edge* is an edge consisting of two straight line segments whose orientations differ by $\pi/4$ [10] (edge ab in Figure 1B). In Euclidean geometry the edges incident to a Steiner point must be separated by $2\pi/3$; they divide a circle centered on the Steiner point into three equal slices. In an octagonal geometry this is not possible because $\pi/4$ (the smallest angular difference between edges) does not divide $2\pi/3$ evenly. The *angle condition* [8, 4] states that the edges incident to a Steiner point form angles of $\pi/2$ or $3\pi/4$ with respect to each other. If a Steiner point has degree three then the angle condition forces the edges to form angles of $3\pi/4, 3\pi/4$ and $\pi/2$. Degree four Steiner points cannot exist in full components containing more than four given points. Furthermore, [4, 9] a full component may have at most 1 crooked edge.

The location of the Steiner points in a SMT are computed recursively using a generalization of Hanna's grid [6, 8, 9] as follows. The given points are define to be on the 0^{th} level grid, denoted $\mathcal{G}^0$. Eight rays along the valid line-segment orientations are extended from points i and j in $\mathcal{G}^i$ and $\mathcal{G}^j$ respectively. The intersections of the rays define candidate Steiner points in $\mathcal{G}^{\max\{i,j\}+1}$. For example, in Figure 1 points D, E, F, G are in $\mathcal{G}^0$, point c is in $\mathcal{G}^1$ and d is in $\mathcal{G}^2$. Not all ray intersections constitute legal Steiner points. Of the 12 ray intersections generated by points i and j in Figure 3A only the 6 numbered in Figure 3B can exist in a SMT due to the angle condition. The arrows in Figure 3B indicate the direction of the third edge incident to the Steiner point so that the angle condition is not violated. The two arrows emanating from points 2 and 5 mean that the third edge incident to these Steiner points may assume either orientation, or both in the case of a crooked edge.

The depth of the Steiner points in the generalized Hanna's grid strongly influences the time complexity of the OSMT algorithm. Clearly, decreasing the depth of the grid decreases the running time of the algorithm. Unfortunately, the results in [9] appear to be incorrect; there are given point sets that force the Steiner points of an OSMT to reside in $\mathcal{G}^{N-2}$.

The most successful SMT solver to date for the rectilinear and Euclidean metrics is GeoSteiner [13, 15]. Both of these problems are known to be NP-complete [5] and yet GeoSteiner is capable of solving instances with hundreds of given points in minutes. The success of GeoSteiner can be attributed, in large part, to the fact that the full components of a SMT are usually small and the number of full components can be pruned extensively. The exact algorithm presented in section 4 borrows from the ideas used in GeoSteiner.

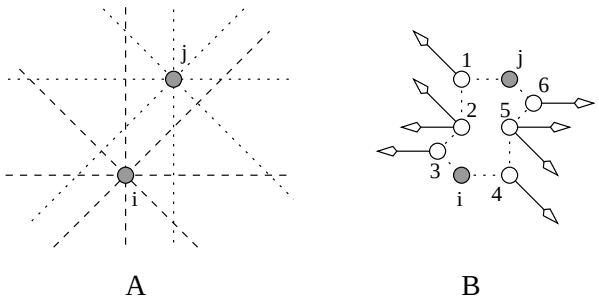

Figure 3: A) The intersections of rays emanating from two points define candidate Steiner points. B) Only the 6 identified points can exist in an OSMT.

3. CANONICAL FORM

The graph topology of a SMT represents the identities of the points (given and Steiner) and their interconnections. If the given point locations and graph topology are fixed then the SMT is characterized by a vector of Steiner point coordinates, $z \in R^{2k}$, where k is the number of Steiner points.

The length of an edge is a piecewise-linear convex function of its coordinate endpoints. Since the length of a SMT is the sum of its edge lengths then it is a convex function of the Steiner point locations. The minimum values of a convex function form a convex polytope. Hence, the set z forms a $2k$-dimensional convex polytope. A SMT is in *canonical form* iff z is an extremal point on this convex polytope. By definition, if a full SMT T is not in a canonical form then:

1) A smaller SMT exists (a contradiction), or

2) there exists two edge disjoint full SMTs T_1, T_2 such that $T_1 \cup T_2$ contains all the given points in T and $length(T_1 \cup T_2) = length(T)$ (a break), or

3) z is not an extremal point on the polytope of Steiner point vectors (a wiggle).

In Figure 4, the Steiner point pair aa' can slide in unison up to cc' without changing the length of the SMT. The convex polytope of possible Steiner point locations can be represented by the straight line ac. Thus, a SMT whose Steiner points are located at either aa' or cc' is in canonical form.

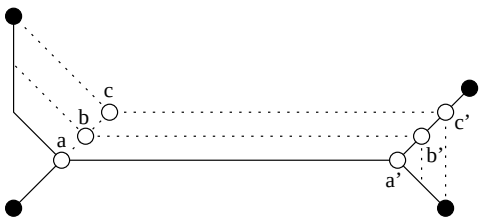

Figure 4: The set of four given points whose Steiner points can move.

The definition of a canonical form naturally quantifies flexibility in a SMT [2]. The dimension of the polytope represents the number of degrees of freedom in the Steiner point locations. For example, in Figure 4 the polytope is one dimensional corresponding to one degree of freedom of the Steiner point along the straight line ac. The volume of the polytope describes the amount of flexibility in the SMT, *i.e.* the length of the line ac.

4. EXACT ALGORITHM

The exact algorithm forms an OSMT in two phases; generation of all possible full components for the set of given points and merging the full components to form an OSMT. The set of full components is formed by "growing" partial component as described below.

A partial component is a minimum spanning tree on n given points and $n-1$ Steiner points, where all the given points have degree 1. All $2n-2$ edges of the partial component are defined to be straight. For now, assume that the OSMT does not have degree 4 Steiner points. Hence, $n-2$ of the Steiner points will have degree 3 and 1 Steiner point, the *available* Steiner point, has degree 2. A single given point is considered a *singleton* partial component whose available Steiner point is the given point. For example, in Figure 1 the point set $\{D, E, c\}$ with edges Dc, Ec is a partial component with available Steiner point c. Point $\{F\}$ by itself is a (singleton) partial component with available Steiner point F. Partial components are built by connecting two partial components to a new Steiner point by two straight edges. For example, in Figure 1 combining partial components $\{D, E, c\}$ and $\{F\}$ yields a partial component $\{D, E, F, c, d\}$ with available Steiner points d.

Full components are then formed by connecting the available Steiner points of two partial components with an edge. The advantage of this technique is that, in general, this edge will be crooked and the location of the crook is implicit in the coordinate end points of the edge. The pseudo-code to generate full components is described in Algorithm 1.

Algorithm 1 Generating full components.

```
for all given points i do
  partial += i;
end for
while depth++ < N − 2 do
  for all i, j ∈ partial do
    partial += HANNA(i, j);
  end for
end while
for all i, j ∈ partial do
  full += JOIN(i, j);
end for
```

The function HANNA takes as input two partial components with available Steiner points i and j. If both of i and j are singletons then 6 new partial components are created as shown in Figure 3B. Otherwise the number of partial components can be reduced from the 6 shown in Figure 3B by invoking the angle condition. If the edges of the non-singleton partial component, incident to the available Steiner point, form an angle of $\pi/2, 3\pi/4$ then the available Steiner point is said to have an even/odd normal respectively. Figure 5 shows an available Steiner point with an even normal being combined to a partial component (potentially a singleton). The angle condition requires that the orientation of the edge incident to the available Steiner point i be horizontal, restricting the number of potential intersections to the two shown. Obviously, this construction is invariant under translation and rotation.

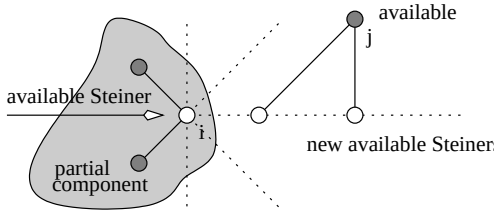

Figure 5: Attaching a given point to a partial component with an even normal.

The *orientation* of an edge is the smallest clockwise angle that the edge makes with respect to the x-axis. Figure 6 shows an available Steiner point with an odd normal being combined with a partial component. The angle condition forces the third edge incident to i to have an orientation of 0 or $\pi/4$. To accommodate this fact, three subcases are required, see j in Figure 6.

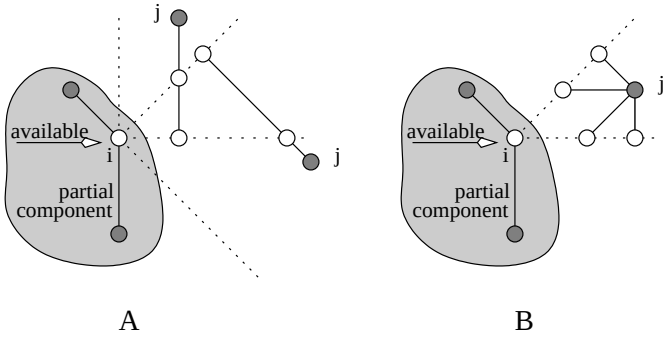

Figure 6: Attaching a given point to a partial component with an odd normal.

Implicit in Figures 5 and 6 is the fact that the newly created Steiner point is connected to the available Steiner point by straight edges. Each newly created Steiner point, p, becomes the available Steiner point of a new partial component formed by the union of, p, copies of the components associated with points i and j, and the two edges connecting p to i and j.

Because each partial component generated is used to generate more partial component, it is desirable to curtail the generation of unnecessary (non-optimal) partial components. To combat this growth, pruning techniques are used to discard any partial component which can be shown to be non-optimal. Four such tests, listed in the order in which they are employed by the OSMT algorithm, are described below.

4.1 Bottleneck Steiner Distance Property

In a minimum spanning tree the longest edge on a path between given points a and b is the Bottleneck Steiner Distance (BSD) of a, b, denoted BSD(a, b). It is straightforward to show [7] that every SMT edge on a path between given points a and b must be shorter than BSD(a, b). The BSDs

are precomputed for all pairs of given points in the octagonal metric. Before two partial components A and B are merged the two new edges are compared against BSD(a, b) $\forall a \in A$ and $\forall b \in B$. If there exists a pair of points for which BSD(a, b) is less than the length of either new edge then the union of A, B is not formed.

4.2 Normal Property

The normal of a Steiner point is the orientation of the edge opposite the $\pi/2$ angle, *e.g* Steiner points c and e in Figure 1 have the same normal. The *normal condition* [4, 3] states that the Steiner points of a full component may contain at most two different normal.

Before two partial components are combined, the union of their normals and the normal of the new Steiner point is checked to make sure that the normal condition is satisfied. If it is not then the union is not formed.

4.3 Lune Property

Denote the distance between two points i and j by $|ij|$. The lune of i and j is the locus of points whose maximum distance from i or j is $|ij|$. It is well known [7] that the area enclosed by the lune formed by the end-points of an edge in a SMT must be free of given or Steiner points. In the octagonal geometry the structure of the lune depends on the angle, θ, formed by the line between the two points i and j with respect to the coordinate axis. If $(\theta+\psi) \bmod \pi/4 < 2\psi$ then the lune is composed of 8 edges as shown in Figure 7B else the lune is composed of 6 edges as shown in Figure 7C, where $\psi = \arctan(\frac{3-\sqrt{2}}{7})$. ψ is determined by computing where two critical vertices of the lunes defined by i and j intersect, see Figure 7, with respect to the point i. If there exists a point (either Steiner or given) in the pair of lunes associated with the two new edges connecting the two partials to their new Steiner point then the union of the partials is not formed.

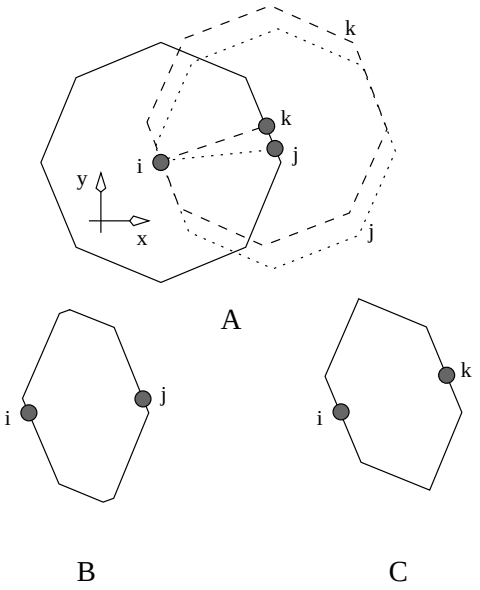

Figure 7: Lunes for two point pairs i, j and i, k.

4.4 Wedge Property

The *wedge property* states that in order for a partial component to become a full component it must be able to connect to some given point without violating the angle condition. With respect to Figures 8A and B, define p1 to be the proposition, "there is at least one given point in region p1." m3,m1, and p3 are similarly defined propositions. Let p2 be the proposition, "there is at least one given point on the line p2." z and m2 are similarly defined.

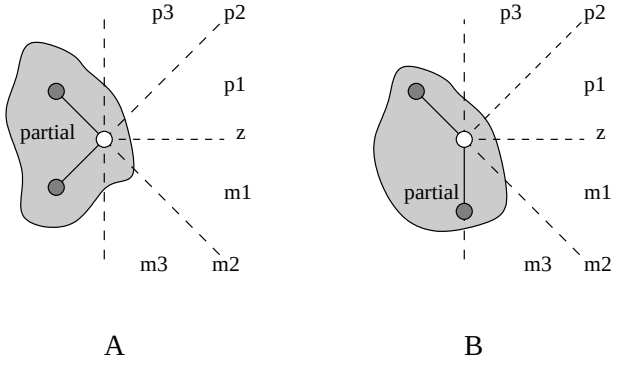

Figure 8: Regions around an available Steiner point.

Assume that a newly formed partial component has an available Steiner point with an even normal, see Figure 8A. If z, p1,p2 and p3 are all false then there is no way any given point can be attached to this newly formed available Steiner point without violating the angle condition. Hence, such a partial component should not be formed. A candidate Steiner point with an even normal satisfies the wedge property if the sentence

$$z \vee (m1 \wedge (p1 \vee p2 \vee p3)) \vee (p1 \wedge (m1 \vee m2 \vee m3)) \vee (p2 \vee m2)$$

is true. Each of the clauses represents a case where a partial component could be transformed into a full component and thus must be preserved for future use. If the first clause is true then the partial component is transformed into a full component by directly connecting the given point on z to the available Steiner point. If the second or third clauses are true then the two given points, together, could form a partial component whose available Steiner point is on z as shown in Figure 5. If the fourth clause is true then the two given points could each be connected to the available Steiner point forming a degree four Steiner point.

A candidate Steiner point with an odd normal, see Figure 8B satisfies the wedge property if the sentence

$$z \vee p2 \vee (p1 \wedge (m1 \vee m2 \vee m3))(m1 \wedge (p1 \vee p2 \vee p3))$$

is true. If the either the first or second clause is true then the given point can be connected directly to the available Steiner point. If the third or fourth clause is true then the two given points could form a full component whose available Steiner point could be connected to the available Steiner point. The two sentences for even and odd normals are sufficient, but not necessary, conditions to guarantee the utility of the newly formed partial component to the formation of a full component.

After all the partial components have been formed they are transformed into full components by JOIN. This function takes as input two partial components with available Steiner

points i and j and attempts to insert an edge (crooked or straight) between them. This new edge is checked against the BSD and lune tests. If these tests are passed then JOIN returns a full component which is the union of the two partial components and the edge joining their available Steiner points.

The number of full trees is reduced before the merging phase in order to reduce the search space. Utilizing the fact [7] that a SMT must have the suboptimality property (any full component of a SMT must be a SMT on the set of given points it spans) the set of fulls is pruned by retaining only the smallest full tree which covers a particular set of given points.

In order to speed up the search for a union of fulls which spans the given points the notion of *compatible* full trees is introduced. Two full trees are compatible if they have exactly one given point in common and the two edges incident to the shared given point do not form an angle less than $\pi/2$. If two full trees do not share any given points they are said to be *independent*, else if they share two or more given points or violate the angle condition at the shared given point they are said to be *incompatible*. Incompatible fulls cannot both be present in a SMT because in the length of the tree could be reduced.

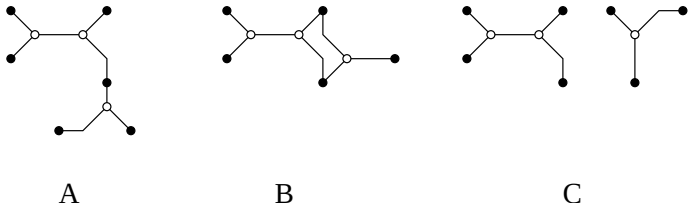

Figure 9: The three cases of compatibility. A) Compatible fulls. B) Incompatible fulls. C) Independent fulls.

Compatible fulls are merged one at a time in a "Prim-like" fashion. That is, full trees are added to a partial OSMT one at a time, maintaining the invariant that the covering forms a connected spanning tree. The merging process is recursive and a search path is terminated whenever it is determined that the OSMT is non-minimal or the set of given points cannot be covered by the remaining full components.

5. EXPERIMENTAL RESULTS

Let the length of a SMT in the rectilinear, octagonal and Euclidean metrics be denoted L_2, L_4, L_∞ respectively. Let the percentage reduction of L_b with respect to L_a be denoted $R_{a,b}$. In the following experiments the rectilinear and Euclidean SMTs were constructed using GeoSteiner [15], while the OSMTs were constructed using the algorithm presented in this paper.

In the first experiment SMTs were constructed in each of the three metrics on a fixed set of 100 instances. Each instance was created using *rand_points* [15] and contained 15 given points over a 1x1 unit square. The average length of a tree was $L_2 = 2.8925$, $L_4 = 2.6375$, and $L_\infty = 2.5371$. The reductions were computed to be $R_{2,4} = 9.75 \pm 2.29$, $R_{2,\infty} = 14.03 \pm 2.60$, and $R_{4,\infty} = 3.90 \pm 0.64$. Clearly, OSMTs are consistently smaller than their rectilinear counterparts.

To gain further insight into the influence that the metric has on the SMT, information about the size and composition of the full components was extracted from the solutions to the instances above. Specifically, the number of full components in a SMT and the number of given point in each full component was recorded. This data is presented in Table 1. The i^{th} row of Table 1 provides the average number of full components with i given points required to build a SMT with 15 given points. The last row of Table 1 gives the total number of full components required to build a SMT with 15 givens points.

Number of given points	Rectilinear	Octagonal	Euclidean
2	3.05	4.44	4.39
3	3.04	2.71	2.85
4	1.05	0.94	1.00
5	0.31	0.21	0.22
6	0.07	0.06	0.03
>6	0.02	0.03	0.00
Total	7.54	8.39	8.46

Table 1: The average number of full components with i given points required to build a SMT with 15 given points.

The OSMT algorithm required, on average, 29.6 seconds to construct each SMT. Compared to the average of 0.01 and 0.22 seconds required by GeoSteiner to construct the rectilinear and Euclidean SMTs, it is clear that the current implementation has a way to go before it approaches the state-of-the-art in SMT construction. In order to establish a baseline, Table 2 presents the efficiency of each pruning test in terms of the rejection rate; the number of full components passed by a test over the total number of full components shown to the test. The order in which the pruning strategies are employed is implicit in the top to bottom ordering of the data.

The majority of the OSMTs algorithms time is spent merging full components. Table 3 gives the percentage of compatible, incompatible and independent full components pairs used to construct the 100 SMTs in the first experiment.

From this table it is clear that only a small portion of the full components are compatible and hence its the sheer number of full components which is creating work for the merging phase.

In the second set of experiments two problem sets in the OR-Library [1] were solved. The OR-Library consists of 15 problem sets, each containing 15 instances. The number of given points in each problem set is the same and varies from 10 to 10,000. Since the current OSMT algorithm experiences a dramatic slow-down at 25 given points, only the first two problem sets were solved. In Table 4 problem instances are

	Number Fulls	Rejection Rate
Created	243843	
Pass BSD test	102058	58.15%
Pass normal test	214401	78.99%
Pass lune test	7994	62.71%
Pass wedge test	5903	26.16%

Table 2: The rejection rate of the various pruning tests.

Compatiable fulls	17.70%
Independent fulls	37.88%
Incompatible fulls	44.43%

Table 3: The distribution of full component pairs.

identified by the number of given points and the instance number in parenthesis. The lengths of hexagonal SMTs are cited in [12].

Instance	Rect	Hex	Oct	Euc
10 (1)	2.29	2.20	2.11	2.02
10 (2)	1.91	1.79	1.67	1.60
10 (3)	2.60	2.39	2.33	2.22
10 (4)	2.04	1.99	1.88	1.79
10 (5)	1.88	1.85	1.76	1.69
10 (6)	2.65	2.41	2.37	2.30
10 (7)	2.60	2.43	2.32	2.23
10 (8)	2.50	2.31	2.25	2.17
10 (9)	2.20	2.07	2.04	1.96
10 (10)	2.39	2.26	2.14	2.05
10 (11)	2.22	2.12	2.02	1.94
10 (12)	1.96	1.96	1.82	1.75
10 (13)	1.94	1.86	1.79	1.71
10 (14)	2.18	2.11	2.02	1.94
10 (15)	1.86	1.83	1.73	1.67
20 (1)	3.37	3.36	3.19	3.07
20 (2)	3.26	3.05	2.97	2.85
20 (3)	2.78	2.70	2.56	2.45
20 (4)	2.75	2.70	2.55	2.46
20 (5)	3.39	3.19	3.13	2.95
20 (6)	3.60	3.42	3.25	3.13
20 (7)	3.49	3.28	3.18	3.05
20 (8)	3.78	3.63	3.42	3.31
20 (9)	3.67	3.51	3.27	3.13
20 (10)	3.40	3.22	3.10	3.01
20 (11)	2.71	2.53	2.42	2.31
20 (12)	3.04	2.87	2.76	2.65
20 (13)	3.44	3.31	3.10	3.02
20 (14)	3.40	3.20	3.05	2.93
20 (15)	3.23	2.96	2.92	2.79

Table 4: The length of the first 30 OR library instances in four different metrics.

The average percentage reductions are $R_{2,4} = 9.80 \pm 1.96$, $R_{2,\infty} = 14.35 \pm 2.13$, $R_{3,\infty} = 8.88 \pm 1.81$, and $R_{4,\infty} = 4.14 \pm 0.67$.

6. CONCLUSIONS

The merging of the full components is currently the limiting factor in the size of the problem instances solved. Clearly, the running time of the merging phase is impacted by the number of full components. Future work will focus on developing stronger pruning techniques for the full components. In addition, the techniques used in [13] to merge full components could be employed to improve the performance of the OSMT algorithm.

The complexity of the OSMT problem is open. However, it should be fairly straight forward to extend the NP-completeness proofs of the Euclidean and rectilinear SMT problems given by Garey *et. al.* [5] to cover the SMT problem in the octagonal and other oriented metrics.

An octagonal approach to VLSI layout holds intriguing promise. Constructing exact OSMTs for a moderate number of point is feasible and reduces wire length to within 4% of the theoretical minimum.

7. REFERENCES

[1] J.E. Beasley, "OR-Library: Distributing Test Problems by Electronic Mail", *Journal of the Operational Research Society*, **41**(11), pp. 1069-1072, 1990.

[2] E. Bozorgzadeh, R. Kastner and M. Sarrafzadeh, "Creating and Exploiting Flexibility in Steiner Trees", *Design Automation Conf.*, pp. 195-198, 2001.

[3] M. Brazil, D.A. Thomas, and J.F. Weng, "Forbidden Subpaths for Steiner Minimum Networks in Uniform Orientation Metrics", *Networks*, **39**(4), pp. 186-202, 2002.

[4] C. S. Coulston, *Steiner Tree Algorithms*, Ph.D. dissertation, Pennsylvania State University, 1999.

[5] M. Garey, R. Graham and D. Johnson, "Some NP-Complete Geometric Problems", *Symposium on Theory of Computing*, pp. 10-22, 1976.

[6] M. Hanna, "On Steiner's Problem with Rectilinear Distance", *SIAM J. Appl. Math.*, **14**(2), pp. 255-265, 1966.

[7] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner Tree Problem*, Springer Verlag, North Holland, 1994.

[8] D.T. Lee and C.F. Shen, "The Steiner Minimal Tree Problem in the λ-geometry Plane", *Int'l Symp. on Algorithms and Computation*, pp. 247-255, 1996.

[9] G Lin and G. Xue, "Reducing the Steiner Problem in Four Uniform Orientations", *Networks*, **35**(4), pp. 287-301, 2000.

[10] M. Sarrafzadeh and C. K. Wong, "Hierarchical Steiner Tree Construction in Uniform Orientations", *IEEE Trans. Computer-Aided Design*, **11**(9), pp. 1095-1103, 1992.

[11] S. Teig, "The X Architecture: Not Your Father's Diagonal Wiring", *ACM International Workshop on System-Level Interconnect Prediction*, pp. 33-37, 2002.

[12] A. Thurber and G. Xue, "Computing Hexagonal Steiner Trees using PCX", *IEEE International Conference on Electronics, Circuits and Systems*, pp. 381-384, 1999.

[13] D.M. Warme, P. Winter, and M. Zachariasen, "Exact Algorithms for Plane Steiner Tree Problems: A Computational Study", in D.Z. Du, J.M. Smith and J.H. Rubinstein (Eds.), *Advances in Steiner Trees*, Kluwer Academic Publishers, Boston, pp. 81-116, 2000.

[14] X Initiative Homepage, www.xinitiative.org, March 2003.

[15] M. Zachariasen, GeoSteiner Homepage, www.diku.dk/geosteiner, March 2003.