Wirelength Reduction by Using Diagonal Wire

Charles Chiang & Qing Su
Synopsys, Inc.
700 E. Middlefield Road
Mountain View, CA 94043
{clc,qsu}@synopsys.com

Ching-Shoei Chiang
Dept. of Comp. and Info. Sci.
Soochow University
Taiwan, ROC
chiang@cis.scu.edu.tw

ABSTRACT

We study the octilinear Steiner tree to evaluate the rectilinear Steiner tree based router. First, we give the worst and average case wirelength for rectilinear routing and octilinear routing for two terminals net. Next, we show the octilinear Steiner trees have smaller wirelength reduction for multiterminal net than that of rectilinear Steiner tree. Then, we propose an $O(|V|+|E|)$ algorithm to construct an isomorphic octilinear Steiner tree from a rectilinear Steiner tree $G = (V, E)$ and prove the wirelength of the isomorphic octilinear Steiner tree is the lower bound. In the end, we show two types of experiment of wirelength reduction results by using diagonal wire The octilinear Steiner tree reduces 9.201% and 6.63% of the wirelength over rectilinear Steiner tree on a set of nets generated at random and on 15 VLSI designs, respectively.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids—*Layout*

General Terms

Algorithms, Experimentation

Keywords

Steiner tree, routing, 45^0 routing, diagonal routing

1. INTRODUCTION

In two-dimensional λ-geometry, only orientations with angles $(i\pi/\lambda)$, for all i, are allowed, where $(\lambda \geq 2)$ is an integer. Either λ divides 360 or 360 divides λ. $\lambda = 2$ corresponds to rectilinear geometry, $\lambda = 4$ corresponds to octilinear geometry, and $\lambda = \infty$ corresponds to Euclidean geometry[8]. Rectilinear geometry, or 2-geometry, has been given much attention because it is practical important on VLSI designs [9].

To improve the routability of 2-geometry, researchers have been extended its studied from 2-geometry (rectilinear) to 4-geometry (octilinear). Compare to 3-geometry, 4-geometry is a natural extension of 2-geometry - 2-geometry is a subset of 4-geometry (see Figure 1). Moreover, empirical results reveal that whereas 2-geometry is 27% less efficient than ∞-geometry, 4-geometry is only 5% less efficient than ∞-geometry.

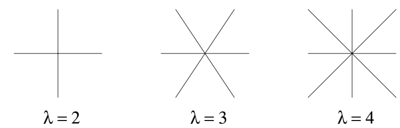

Figure 1: Example of λ-geometry

In this paper, we show the average 17% wirelength reduction from 2-geometry routing to 4-geometry routing on two terminals net in Section 2. Next, we state that the wirelength reduction on multiterminal net for 4-geometry is less than 2-geometry in Section 3. In Section 4 we first describe the difficulties of constructing a 4-geometry Steiner tree. Then we propose an algorithm to construct an isomorphic 4-geometry Steiner tree from a 2-geometry Steiner tree. We also prove total wirelength of the isomorphic 4-geometry Steiner tree obtained by our algorithm is the lower bound. In the end we show the wirelength reduction using our proposed algorithm. For randomly generated nets, we have 9.201% wirelength reduction. On 15 VLSI designs, the wirelength reduction are 7.86% for two terminal nets, 6.16% for multiterminal nets and 6.63% on total wirelength in Section 5.

2. TWO TERMINAL NET

Without considering any obstacles, let ℓ_∞, ℓ_4 and ℓ_2 denote the wirelength, or distance, between two arbitrary points ρ_1 at (x_1, y_1) and ρ_2 at (x_2, y_2) in ∞-geometry, 4-geometry and 2-geometry, respectively. Moreover, $\triangle(\rho_1, \rho_2) = \text{Max}(\triangle x, \triangle y)$ - $\text{Min}(\triangle x, \triangle y)$ and $\triangle x = |x_1 - x_2|$, $\triangle y = |y_1 - y_2|$. Hence,

$$\ell_\infty = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

$$\ell_4 = \triangle(\rho_1, \rho_2) + \sqrt{2} \times Min(\triangle x, \triangle y) \quad (2)$$

$$\ell_2 = \triangle x + \triangle y \quad (3)$$

Let ℓ_λ^{worst} and ℓ_λ^{avg} denote the worst case wirelength and average case wirelength for λ-geometry. Without loss of generality we considering only angles in the first octant. The worst case wirelength for 2-geometry and 4-geometry are

$$\ell_2^{worst} = \ell_\infty \sin\theta + \ell_\infty \cos\theta \approx 1.41\ell_\infty, \ where\ \theta = \pi/4$$
$$\ell_4^{worst} = \ell_\infty / \cos\theta \approx 1.08\ell_\infty, \ where\ \theta = \pi/8$$

Note that, the ℓ_∞ is the shortest route between any two given points (Euclidean distance). The average wirelength of 2-geometry and 4-geometry can be obtained by integrating ℓ_2 or ℓ_4 from 0 to $\pi/4$ and dividing by $\pi/4$.

$$\ell_2^{avg} = \frac{4}{\pi}\ell_\infty \approx 1.27\ell_\infty \quad (4)$$
$$\ell_4^{avg} = \frac{8(\sqrt{2}-1)}{\pi}\ell_\infty \approx 1.055\ell_\infty \quad (5)$$

LEMMA 1. *For two terminal net, the ℓ_4^{avg} is 17% shorter than ℓ_2^{avg} and about 5% longer than ℓ_∞.*

3. MULTITERMINAL NET

In 2-geometry, multiterminal net has been routed as Steiner tree to reduce the wire length. For example, the wirelength of a three terminal net, as in Figure 2(a), can be reduced dramatically by adding one Steiner point, as in Figure 2(b).

Let's define both minimum spanning tree (MST) and Steiner minimal tree (SMT) first. Given a weighted graph $G = (V, E)$, select a subset of edges $E' \subseteq E$ such that E' induce a tree T and the total weight of T is minimum over all such trees. Hence, T is the Minimum Spanning Tree (MST) of G. Note that, the weight of each edge in E is obtained from Equation(1), Equation(2), and Equation(3) for ∞-geometry, 4-geometry and 2-geometry, respectively. A Steiner tree with minimal total weight (i.e., length) is called a Steiner Minimal Tree (SMT). A SMT problem can be defined as following: given a weighted graph $G = (V, E)$, a subset $D \subseteq V$ as demand vertices, the other subset $S \subseteq V$ as Steiner point candidates, $D \cap S = \emptyset$ and $D \cup S = V$. Select a subset $V' \subseteq V$, for $D \subseteq V'$ and $V' - D \subseteq S$, and V' induces a tree of minimal cost over all such tree. In routing, D is referred as terminals or pins and $V' - D$ is referred as Steiner points. Please see Figure 2 for MST and SMT for a graph G with $|D| = 3$ and $|V'| = 4$ in 2-geometry.

The problem of constructing an SMT is known to be NP-hard [3]. Therefore, many heuristic algorithms use MST as starting point and apply local modification to get a Steiner tree [5]. This is because the upper bound ratio, γ, of the length of any MST to that of an SMT has been found. Let γ_λ represent the γ for λ-geometry. Then $\gamma_\lambda = \max \frac{\omega(MST_\lambda)}{\omega(SMT_\lambda)}$ for all MST and SMT pairs in λ-geometry, where ω is the sum of weight of all edges, MST_λ and SMT_λ are the MST and SMT in λ-geometry, respectively. The γ on several λ-geometry can be found in [2] [6] [10] for the following results

$$\gamma_2 = \frac{3}{2} = 1.5 \quad (6)$$
$$\gamma_4 = \frac{4}{2+\sqrt{2}} \approx 1.172 \quad (7)$$
$$\gamma_\infty \leq \frac{2}{\sqrt{3}} \approx 1.155 \quad (8)$$

For example, the weight of a MST is at most 1.5 times of the weight of its SMT in 2-geometry according to Equation (6). Since we focus on 2-geometry and 4-geometry, we have the following Lemma from Equation (6) and (7)

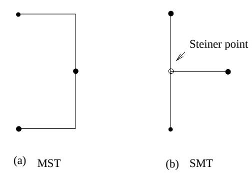

Figure 2: 2-geometry MST and SMT

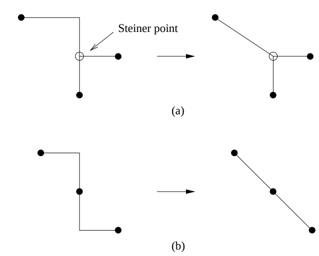

Figure 3: Multiterminal Net Cases

LEMMA 2. *The upper bound wirelength reduction ratio from a MST to its SMT in 4-geometry (γ_4) is less than that of 2-geometry (γ_2).*

4. 4-GEOMETRY STEINER TREE

From Lemma 1 and 2 we conclude that the the average wirelength reduction between 2-geometry and 4-geometry is 17% for two terminals net. For multiterminal net, Steiner tree has been used to connect all terminals. But the 4-geometry Steiner tree wirelength reduction ratio is smaller than 2-geometry's for multiterminal net. We would like find a way to construct a 4-geometry Steiner tree such that we can use the result to compare with the result of 2-geometry Steiner tree.

It is obvious that not every 4-geometry SMT produces wirelength reduction over its 2-geometry SMT. For example, if all terminals of a net lie on a vertical line or a horizontal line then the wirelength of both routings are the same. When all terminals do not lie on a vertical or a horizontal line then the reduction can vary. For example, the 4-geometry SMT does has some wirelength reduction over 2-geometry SMT in Figure 3(a). Next, the 4-geometry SMT in Figure 3(b) achieves the most wirelength reduction over its 2-geometry SMT. However, The problem of constructing an SMT is known to be NP-hard [3]. The 2-geometry and ∞-geometry have been studied on the context of Steiner tree. But the 2-geometry and ∞-geometry were not easily extended to λ-geometry, for arbitrary λ. For 4-geometry SMT, only handful of researchers study its three and four terminals Steiner tree.

To construct 2-geometry Steiner tree, most researchers utilize a nice property on 2-geometry - all Steiner point candidates are vertices of Hanan grid [4]. A Hanan grid for 2-geometry is obtained by passing a vertical line and a horizontal line through each terminal of a net. From Hanan grid, there are mainly two approaches to construct the 2-geometry SMT. One is to construct a MST first and then, based on Hanan grid, apply heuristic to obtained its SMT. The second approach is to utilize the Steiner point location knowledge from Hanan grid and construct the SMT.

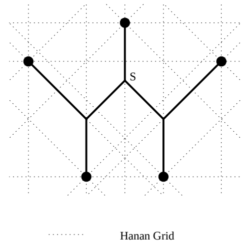

Figure 4: 4-geometry Hanan Grid

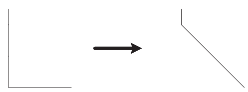

Figure 5: edge_conversion

But 4-geometry Hanan grid does not contain all Steiner point candidates for 4-geometry. A Hanan grid for 4-geometry is obtained by passing vertical, horizontal, 45^0, and 135^0 lines through each terminal. Figure 4 shows a Steiner tree of a five terminals with one Steiner point s not in its Hanan grid. Therefore, our approach to construct a 4-geometry SMT is starting from a 2-geometry SMT.

We convert a 2-geometry Steiner tree, $T_2 = (V_2, E_2)$, into an *isomorphic* 4-geometry Steiner tree, $T_4 = (V_4, E_4)$. That is, T_2 and T_4 are 1-1 correspondences $f : V_2 \rightarrow V_4$ and $g : E_2 \rightarrow E_4$ such that for every edge e from vertex u to vertex v in T_2 there is an edge $g(e)$ from vertex $f(u)$ to vertex $f(v)$ in T_4.

We show how to convert E_2 to E_4 by utilizing the $\pm 45^0$ line in 4-geometry first. Then we consider the Steiner point in T_2 to see how to reduce wirelength by sliding it location.

Considering edge $e, e \in E_2$, e has at most two wire segments, one horizontal with wirelength of $\triangle x$ and the other vertical segment with wirelength of $\triangle y$. For two segments edge, we do *edge_conversion* to replace all or portion of it by a 45^0 or 135^0 degree segment of wirelength (Figure 5). Hence,

Lemma 3. *For each edge $e, e \in E_2$, the wirelength reduction is $(2 - \sqrt{2}) \times Min(\triangle x, \triangle y)$.*

From Section 3, a vertex v in a 2-geometry Steiner tree $T_2 = (V, E)$, $v \in V$, is either a demand point (terminal) or a Steiner point. The degree of a Steiner point in T_2 can only be either three or four. To satisfy the isomorphic between T_2 and T_4, we only move the locations of vertices but not change it connectivity. For terminal, its locations has been predetermined and cannot be changed. Moving degree four Steiner point does not decrease the wirelength. For degree three Steiner point, we can slide the Steiner point, *Steiner_sliding*, by $Min(w_1, w_2, w_3)$ to reduce the wirelength, where w_1, w_2 and w_3 are wirelength of three wire segments connecting to the degree three Steiner point. From Figure 6, the Steiner point s is slide to $s^{'}$ such that the wirelength will be reduce from $3 \times w_2$ to $2\sqrt{2} \times w_2$. Therefore,

Lemma 4. *For every degree three Steiner point v in T_2, sliding the Steiner point can reduce the wirelength by (3 - $2\sqrt{2}$) $\times$ $Min(w_1, w_2, w_3)$, where w_1, w_2, and w_3 are wirelength of three wire segments connect at v.*

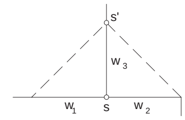

Figure 6: Steiner_sliding

We propose the following algorithm to construct a 4-geometry Steiner tree.

Algorithm Steiner_tree_conversion

1. Give a 2-geometry Steiner tree $T_2 = (V, E)$
2. For each edge in E, a edge_conversion is performed
3. For each degree three Steiner point in V, a Steiner_sliding is executed
4. Output an isomorphic 4-geometry Steiner tree T_4

Theorem 1. *The Steiner_tree_conversion takes $O(|V| + |E|)$ to convert a 2-geometry Steiner tree $T_2 = (V, E)$ to an isomorphic 4-geometry Steiner tree T_4, where $\omega(T_2) - (\Re_e + \Re_v) = \omega(T_4)$, where $\Re_e$ and $\Re_v$ are wirelength reduction from edge_conversion and Steiner_sliding, respectively. Moreover, the $\omega(T_4)$ is the lower bound for $T_2's$ isomorphic 4-geometry Steiner tree T_4.*

Proof: Skip due to page limitation. □

5. EXPERIMENTAL RESULTS

We provide two types of experimental results. The first one is on randomly generated nets and the second one is on 15 VLSI designs from our regression tests. This two different sets of testing data represent two different situations. The locations of terminals from the first set are obtained at random but the locations of terminals from the second set are from a placement tool, therefore, are affected by obstruction, congestion, timing and 2-geometry detailed router.

Table 1: Steiner_tree_conversion Results

# term	(2-MST/ 2-ST)-1	(2-MST/ 4-ST)-1	(2-ST/ 4-ST)-1	(4-MST/ 4-ST)-1
10	9.95	20.35	9.48	3.67
20	9.95	19.74	9.31	3.60
50	9.99	19.77	9.01	3.61
100	9.82	19.92	9.20	3.57
200	9.97	20.11	9.22	3.58
500	10.07	20.07	9.09	3.60
1000	10.12	20.20	9.16	3.64
2000	10.10	20.18	9.17	3.62
5000	10.12	20.23	9.18	3.65
10000	10.10	20.22	9.19	3.64
average	10.019	20.079	9.201	3.618

For first set of data, we randomly generated 100 nets for each terminals set, range from 10 terminals to 10,000 terminals, as shown in Table 1 and Table 2. The terminals are drawn uniformly at random from a 20,000 by 20,000 grid for each net. To build a 2-geometry Steiner tree, we implement linear algorithm from [5] and show the results in column 2 of Table 1. Then, we compare our 4-geometry Steiner tree obtained by our algorithm with 2-geometry MST, 2-geometry Steiner tree and 4-geometry MST in the last three columns

Table 2: Octilinear Steiner Tree Results

#	Edge-Based		Batched		Isomo.	
	%	CPU	%	CPU	%	CPU
100	4.28	0.53	4.43	0.01	3.57	0.0001
500	4.12	13.41	4.29	0.12	3.60	0.0005
1000	4.12	56.64	4.25	0.30	3.64	0.0015
5000	4.17	1466.30	4.31	2.82	3.65	0.0072
10000	4.13	5946.82	4.28	8.36	3.64	0.0124
avg	4.16		4.31		3.62	

of Table 1, with averaging wirelength reduction of 20.079%, 9.201%, and 3.618%, respectively. Moreover, we show experiment result of Lemma 2 on Table 1. As stated in Lemma 2, (γ_4 -1) = 3.618% in the last column is less than (γ_2 -1) = 10.019% in the second column on Table 1.

We also compare our algorithm with two other octilinear Steiner tree algorithms in Table 2. The improving percentage column for each algorithm is comparing a octilinear MST to its octilinear Steiner tree by using the same formula in the last column of Table 1. The two comparing algorithms are $O(n^2)$ edge-based in [1] and the $O(n \log^2 n)$ batched greedy algorithm in [7]. Both edge-based and batched greedy algorithms were run on a dual 1.4 GHz Pentium III Xeon server with 2GB of memory running Red Hat Linux 7.1 [7]. Our algorithm was run on a 4 400 MHz Sun Sparc CPUs machine with 4GB memory. Note that, the run time of algorithm confirms this is a linear algorithm. Our wirelength reduction improving ratio is 13% off edge-based and 16% worse than batched greedy but the run time clearly shows our domination.

We select 15 designs from our regression tests to cover different type of designs. We show the wirelength difference in Table 3. Note that, we keep the same placement for both 2-geometry routing and 4-geometry routing. We route designs with our 2-geometry detailed router first. The result of 2-geometry router is a legal one – have been checked with a third party DRC tool without any violation. Then we apply Algorithm Steiner_tree_conversion for each net without checking the legality of the diagonal wire. Therefore, some diagonal wire might cause DRC. Hence, the wirelength results of 4-geometry routing here are most likely shorter than wirelength results of a legal 4-geometry routing. Table 3 shows an average of 7.86% wirelength reduction from 2-geometry routing to 4-geometry routing for two terminal nets in column 2. This is far below the average 17% from Lemma 1. For the multiterminal nets, we have an average of 6.16% in column 3 of Table 3. The total wirelength reduction is 6.63% from last column of Table 3.

6. CONCLUSIONS

In this paper we give the average wirelength reduction, 17% from 2-geometry routing to 4-geometry routing for two terminal net. For multiterminal net, we show that the improving total wirelength ratio by converting MST to SMT decrease from 1.5 to 1.172 from 2-geometry routing to 4-geometry routing. Then we propose an algorithm to construct an isomorphic 4-geometry Steiner tree for multiterminal net using 2-geometry Steiner tree as starting point. We prove the total wirelength of the isomorphic 4-geometry Steiner tree is the lower bound. Final, we show our experimental results of 2-geometry routing and 4-geometry routing. The wirelength reduction is about 9.201% for nets generated at random. Moreover, the wirelength reduction are 7.86%, 6.16%, and 6.63% for two terminal nets, multiterminal nets and all nets on 15 designs.

Table 3: Wirelength Reduction

	2-term Nets		M-term Nets		Total
	# nets	%Imp.	# nets	%Imp.	%Imp.
d1	2490	7.81	1821	6.19	6.50
d2	3342	8.56	2048	6.58	6.69
d3	4360	8.50	2863	6.34	6.90
d4	5019	7.32	2579	6.06	6.49
d5	9348	8.05	4686	5.99	6.41
d6	16577	7.69	7554	5.74	6.11
d7	24718	8.28	17033	6.65	7.23
d8	29997	7.80	15865	5.91	6.32
d9	33325	6.49	11044	5.93	6.24
d10	65325	8.19	42386	6.39	6.95
d11	66228	8.21	42238	6.61	7.14
d12	85671	8.06	30315	6.29	7.02
d13	97228	7.28	52982	5.67	6.20
d14	98204	7.38	50159	5.66	6.25
d15	112864	8.30	68958	6.33	6.92
avg.		7.86		6.16	6.63

7. REFERENCES

[1] M. Borah, R. M. Owens, and M. J. Irwin. "A Fast and Simple Steiner Routing Heuristic". *Discrete Applied Mathematics*, 90(1-3):51–67, 1999.

[2] D. Z. Du and F. K. Hwang. "A Proof of the Gilbert-Pollak Conjecture on the Steiner Ratio". *Algorithmica*, 7:121–135, 1992.

[3] M. R. Garey and D. S. Johnson. "The Rectilinear Steiner Tree Problem is NP-Complete". *SIAM Journal on Applied Mathematics*, 32(4):826–834, Apr. 1977.

[4] M. Hanan. "On Steiner's Problem with Rectilinear Distance". *SIAM Journal on Applied Mathematics*, 14(2):255–265, Feb. 1966.

[5] J. M. Ho, G. Vijayan, and C. K. Wong. "New Algorithm for the Rectilinear Steiner Tree Problem". *IEEE Trans. Comput.-Aided Design Integrated Circuits*, 9(2):185–193, Feb. 1990.

[6] F. K. Hwang. "On Steiner Minimal Trees with Rectilinear Distance". *SIAM Journal on Applied Mathematics*, 30(1):104–114, Jan. 1976.

[7] A. B. Kahng, I. I. Mandoiu, and A. Z. Zelikovsky. "highly scalable algorithms for rectilinear and octilinear steiner tree". In *Proc. Asia and South Pacific Design Automation Conference*, pages 827–833, Jan. 2003.

[8] M. Sarrafzadeh and C. K. Wong. "Hierarchical Steiner Tree Construction in Uniform Orientations". *IEEE Trans. Comput.-Aided Design Integrated Circuits*, 11(8):1095–1103, 1992.

[9] M. Sarrafzadeh and C. K. Wong. *An Introduction to VLSI Physical Design*. McGraw-Hill, 1996.

[10] C. F. Shen. *"The λ - geometry Steiner Minimal Tree Problem and Visualization"*. PhD thesis, Northwestern University, Evanston, IL, USA, June 1997.