Neural Self-Organization Based 3D Rectilinear Steiner Minimal Tree Generation

Rushabh Agrawal, Rishabh Mehrotra
Birla Institute of Technology and Science
Pilani – 333 031, Rajasthan, India
Email : rushabh0803@gmail.com
erishabh@gmail.com

Atanendu Sekhar Mandal
IC Design Group,
CSIR-CEERI
Pilani – 333 031, Rajasthan, India
Email : atanendusekhar.mandal@gmail.com

***Abstract*—Given N points in a plane, generation of Rectilinear Steiner Minimal Tree (RSMT) is always a challenging problem (NP hard) with numerous applications. As the number of points increases, the complexity of the problem increases exponentially. A neural self organization based method with linear complexity and linear memory requirements has been used for generation of Rectilinear Steiner Minimal Tree in 3D space. The system is initialized by constructing an open curve around the given set of points and each given point is connected to the nearest point generated on the open curve. An energy equation is framed reflecting the length of the system and the total energy of the system is subsequently minimized iteratively using Neural Networks. The nature of the open curve and its other parameters are determined experimentally. The methodology will have significant applications in multilayer VLSI/ULSI interconnection design and for resource connections in any plant design**.

Keywords-RSTM; self-organizing; neural network; NP hard ;circuit design

I. INTRODUCTION

The Steiner Minimal Tree (SMT) problem is one of the most important combinatorial optimization problems. The geometric version of the SMT problem is how to interconnect a set of given points in a metric space such that the total length of segments used between points is minimal. The optimal solution must have a tree structure, which is called a Steiner tree. A classic intractable problem, it has many applications, especially in the physical design of VLSI circuits, mail routing, telephone line networks [5] to even computer vision problems like object detection [6]. It is a well-known NP-hard problem [7] ie an exact optimal solution cannot be constructed in optimal time and approximate solutions need to be looked for. A large no. of applications especially those of VLSI design, are based on a more restrictive version of the SMT problem known as the Rectilinear SMT or RSTM problem. This too is a NP-hard problem[8].

Many approaches have been proposed for solving the RSTM problem. One set of methods make use of an initial Spanning Tree on the set of points, followed by iteratively improving it, to come up with a solution [9][10]. Many heuristics have also been developed for the rectilinear Steiner problem. An early example by Smith, Lee and Liebman [11] divided the given points into components of three and four points. It returned RStTs that were approximately 92% as long as minimal rectilinear spanning trees on instances of up to 40 points. Kahng [12] described a “shrinking bubble" heuristic that identified RStTs that were about 91% as long as the corresponding MRStTs on instances of up to 100 points. Among the heuristics developed for the rectilinear Steiner problem are several evolutionary algorithms. An approach using Genetic Algorithms has also been proposed in [13]. Wakabayashi [14] too describs genetic algorithm for a multi objective problem on rectilinear Steiner trees. [15] describes an ant colony optimization[16] approach for solving a RSMT. Approach of using Neural Network based self-organizing has also been used in [17][18] to solve a general SMT problem. One of the authors involved in this work has also been involved in fine-tuning the NN-based approaches for the 2D steiner tree problem [3].

In our paper, we use a NN-based self-organizing approach to solve a 3 dimensional RSMT problem. The section II of the paper formulates the problem describing terms which will be used in the rest of the paper. Section III caters to the mathematical modeling of the RSMT problem and describes the equations to solve it and Section IV lists the experimental observations and results based on them. Section V discusses the results obtained and Section VI ends with conclusion and scope of future work.

II. THE RSMT PROBLEM

The 3D Rectilinear Steiner Minimal Tree Problem (RSMTP) is to find the shortest tree connecting N given site points $\{p_1, p_2, \ldots\ldots, p_n\}$ lying in 3D space, where the tree may contain vertices other than the site points, called the Steiner points. The Steiner points are denoted by $\{s_1, s_2, \ldots\ldots s_k\}$, where $K \le (N-1)$. The RSMTP is specified by the set of site points p_i $(i = 1,2 \ldots.N)$. Each site point p_i is specified by its coordinates $(p_{i1},p_{i2},\ldots,p_{iD})$, where D is the dimension. In 3D rectilinear space, $D=3$. The site points are encircled by an open curve in spiral form. A number of extra points, called curve points are lying on this open spiral curve C. These curve points will converge to the future Steiner points. Fig 1 suggests that one can consider the Steiner tree to be a piece wise linear open curve C

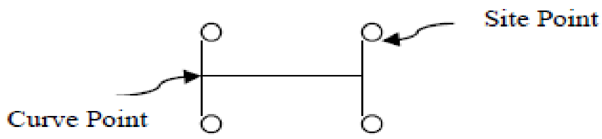

Figure 1 : Rectilinear Steiner Minimal Tree in 2D space

with some vertices of C at the site points and the other site points connected by a rectilinear path to the closest vertex on C.

The Steiner points or the location of the vertices on the curve C are not determinable apriori. However, one may use RSMT network to find a Steiner tree by essentially solving an optimization problem to determine the location of the curve points. These correspond to the activities of the neurons in the network.

The Rectilinear Steiner Minimal Tree in 3D can be used in multilayer VLSI/ULSI interconnection applications.

III. The Model

Let us consider a total of *M* curve points, each having a coordinate (x_{i1}, x_{i1}, x_{i1}) (i = 1, 2,….,M) and let each site point be connected to he nearest curve point. Now considering the network for solving the 3D RSMTP to be consisting of M neurons with activity vector $x_i = (x_{i1}, x_{i1}, x_{i1})^T$, (i = 1, 2,….M). The energy function associated with this network is given by [1]

$$E = \sum_1^M dist(1, x_{i+1}, x_i) + \sum_1^N \sum_1^M d_{ij} h(d_{ij}) \ ; \quad (1)$$

where

$$dist(1, z_i, z_j) = \left((z_{i1} - z_{j1}) + (z_{i2} - z_{j2}) + (z_{i3} - z_{j3}) \right) \ ; \quad (2)$$

$$d_{ij} = dist(1, p_i, x_j) \ ; \quad (3)$$

and $$h_{ij} = \begin{cases} 1, & d_{ij} = min_k\, d_{ik} \\ 0, & otherwise \end{cases} \quad (4)$$

The dynamics of the network corresponds to a steepest descent of the energy function given by (1). The differential equation governing the activities of the neurons is given by [1]. Also,

$$\dot{x}_{ij} = -\frac{\partial E}{\partial x_{ij}}$$

$$= sgn\left(x_{i+1,j} - x_{i,j}\right) + sgn\left(x_{i-1,j} - x_{i,j}\right) + \sum_{k=1}^N h(d_{ki})\ sgn\left(p_{k,j} - x_{i,j}\right) \quad (5)$$

where
$$\begin{aligned} sgn(z) &= 1, && \text{if } z > 0 \\ sgn(z) &= -1, && \text{if } z < 0 \\ sgn(z) &= 0, && \text{if } z = 0 \end{aligned} \quad (6)$$

The eqn. 1 represents the sum of the\length of C, and the lengths of the connection from site points to the curve point nearest to them. Since the network finds the minimum of the energy function we obtain the corresponding Steiner tree.

When the number of site points exceed 3, there may be more than one Steiner points and it becomes difficult to visualize the energy and tree length landscape. To get over this problem, in practice $h(d_{ij})$ is approximated by

$$h(d_{ij}) = \varphi(d_{ij}, \beta_t) \Big/ \sum_{k=1}^M \varphi(d_{ik}, \beta_t) \quad (7)$$

where β_t decreases to 0 as $\lim_{t \to \infty}$. Now, φ should be a decreasing function of d_{ij} and an increasing function of β_t. One possible choice is

$$\varphi(d_{ij}, \beta_t) = \exp(-d_{ij}^2 / \beta_t^2) \quad (8)$$

which we have used in the course of our experiments as well.

IV. Experimental Observations and Results

A. Simulation Tools

The above equations were implemented in Java (the language and platform are no issues and more appropriate ones can be chosen depending on the user system). This section describes the algorithmic approach and the subsequent observations from our experiments.

B. Initial Configuration

The first step is the generation of curve points in an open curve around the site points. As shown in Fig 2, the curve point are generated in a spiral around the site points. From a practical point of view, the RSMT problem would involve the *site points*(in black) to be in a specific set of planes (Fig 2 shows 10 points in 3 different planes) and as a part of initial setup, a set of *curve points* (in green) are generated on an open curve (in blue). The red lines connect each site point to the nearest curve point.

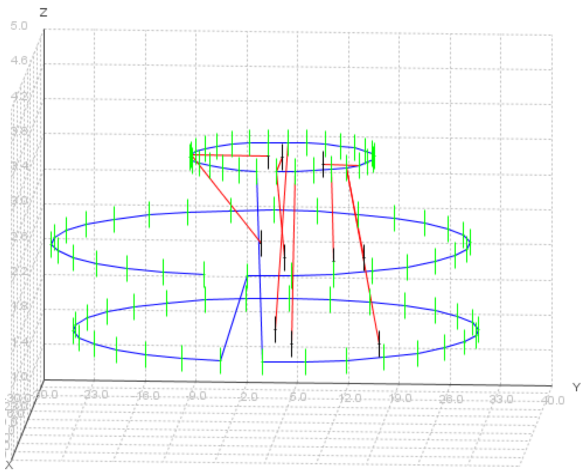

Fig. 2

- The curve points are generated in only those layers(the XY-planes in Fig 2) where the site points reside .
- The curve points are generated in each layer in the form of a circle.
- The radius of the circle is *factor* times the maximum distance between any two site points in that layer. The center for the circle is the mean of all site points in the layer.
- The no. of curve points in the layer are *theta* times the no. of site points in that layer.

Experimentally, its observed that the an optimum solution is obtained with *theta* being 4 or 5 and *factor* between 1 to 5. The results enumerated in this paper take *theta* to be 4 while the value of *factor* is allowed to iterate from 1.1 to 4.9 (The maximum limit of 5 was chosen because, beyond that, there was not much difference in final solution).

Following this initial setup, the equations enumerated in Section II of this paper are simulated and results for a 5 set of points have been given in Table 1. Table 1 compares the lengths of the RSMT with the corresponding Minimal Spanning Tree (MST) for each of the point set.

Table 1 : Results

Point Set	MST	factor	stage1	final
1 (10 pts)	53	1.5	53.10 (90) *	46 .00(9)
2 (10 pts)	43	4.9	54.72(90)	42 .00(8)
3 (12 pts)	55	3.9	54.91(90)	52 .00(12)
4 (14 pts)	63	1.5	84.49(120)	62.00 (12)
5 (20 pts)	65	3.9	73.78 (90)	68 .00(19)

* 53.10(90) indicates that length of Steiner Tree is 53.10 units with 90 Steiner Points. Similarly for all other values.

The values in Table 1 are explained below:

- Using the initial set of curve points as Steiner points, the equations are simulated until there is no further reduction in length of the Steiner Tree. The value in RSMT (*stage-1*) is this final length. Following this, the extra or redundant curve points (those with degree 2 or less) are eliminated [2].
- Another round of iterations take place which further optimize the distance on the reduced point set. When the length gets stabilized again, another round of elimination is done.
- If elimination results in further reduction of points, previous step is repeated. If not, final solution is obtained (*final* in Table 1).

The point sets are enumerated below :

Pt set 1: (3,5,4), (1,4,2), (2,2,3), (2,3,4), (12,19,2),
(14,17,3), (17,13,3), (9,11,4), (13,7,2), (14,6,3)

Pt set 2: (11,5,4), (9,4,2), (2,2,4), (2,3,4), (11,9,2), (14,17,3),
(17,13,3), (9,11,4), (3,7,2), (3,6,3)

Pt set 3: (13,5,4), (9,4,2), (2,2,3), (2,3,4), (12,9,2), (14,17,3),
(7,13,3), (9,11,4), (3,7,2), (14,6,3),(7,16,4),(11,8,2)

Pt set 4 : Pt set1,(7,16,4), (11,8,2),   (12,6,5), (2,9,4)

Pt set 5 : Pt set1, (13,6,4), (1,6,2), (2,6,3), (2,7,4), (4,19,2),
(14,7,3), (7,1,3), (9,1,4), (1,7,2), (14,9,3)

The results for point set 1 are represented in Figs 3(a) and 3(b) and the results for point set 2 are represented in Figs 4(a) and 4(b).

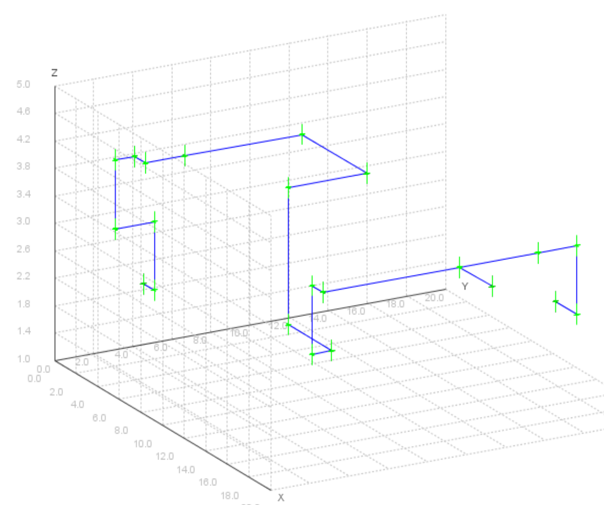

Figure 3(a) : MST for point set 1

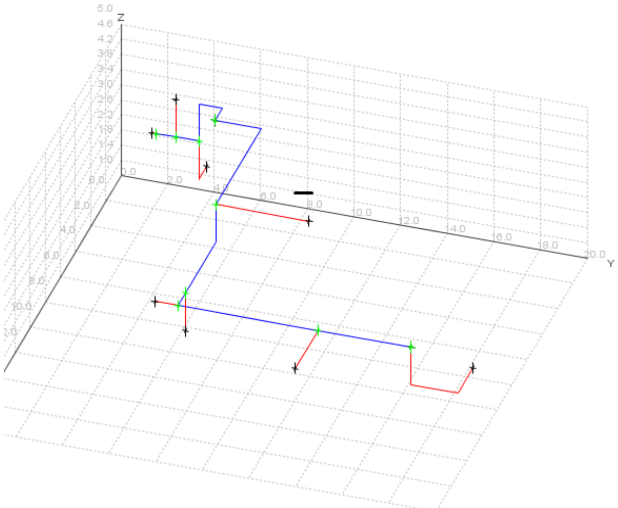

Figure 3(b) : RSMT for point set 1

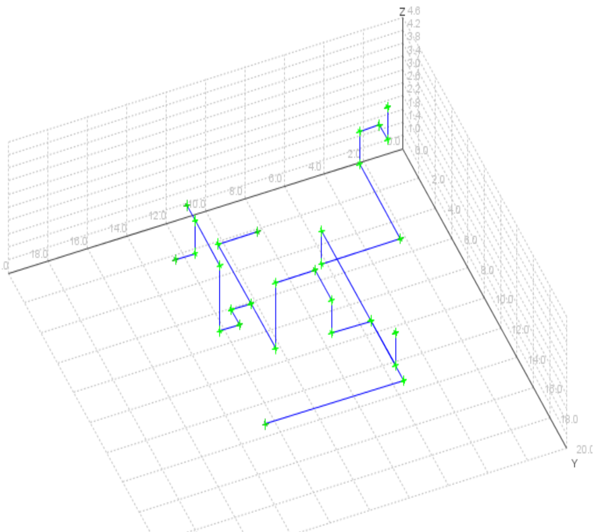

Figure 4(a) : MST for point set 1

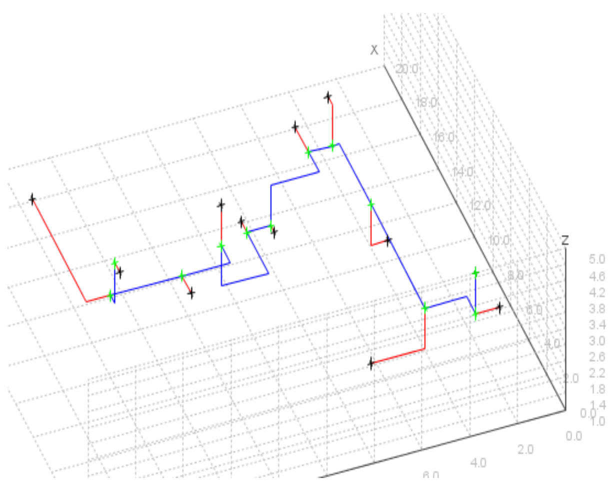

Figure 4(b) : RSMT for point set 1

V. Discussion of Results

A. *Generation of Curve Points*

Just like for the 2D Euclidean SMT (ESMT) and 2D RSMT problem in which final results depended upon the initial shape and orientation of the open curve [2][3], similar was the case with 3D. Among the various shapes of the open curve experimented with, 2 methods stood out. One approach was generation of curve points on a sphere around the site points which gave good results for ESMT 3D problem (Method 1) but for the RSMT problem, the approach of generating curve points in specific planes (as explained in Section III) gave better results(Method 2). For comparison purposes, see Table 2 and Fig 5.

Table 2

Point set	MST	Method 1	Method 2
1(10 points)	53	50.00(9)	46.00(9)
2(14 points)	63	65.00(13)	62.00(12)

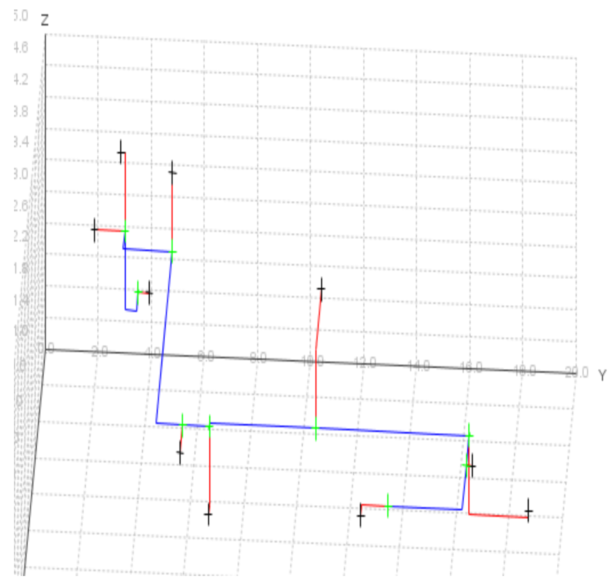

Figure 5 : RSMT using Method 1 for point set 1 (compare with Fig 3 (b))

B. *Factor values*

As mentioned in Section III, *factor* values are varied from 1.1 to 4.9 and the most optimum solution is obtained in this range. As enumerated in Table 3, equally optimum solutions (in terms of length of Steiner Tree and no. of Steiner points) can be obtained for multiple values of *factor*.

As seen from Table 3, the RSTM length is 46 units for 3 values of factor - 1.3, 1.5 and 4.1.

Table 3: RSTM lengths for various Factor values (pt set 1)

Factor	Final Length of RSMT
1.1	53.00
1.3	46.00
1.5	46.00
1.7	61.00
1.9	57.00
2.1	58.00
2.3	62.00
2.5	52.00
2.7	51.00
2.9	53.00
3.1	58.00
3.3	57.00
3.5	48.00
3.7	58.00
3.9	58.00
4.1	46.00
4.3	53.00
4.5	49.21
4.7	51.00
4.9	66.00

The Fig 6 and Fig 7 show the final RSMT for *factor* values 1.3 and 4.1 (for comparison purposes ref. Fig 2(b)).

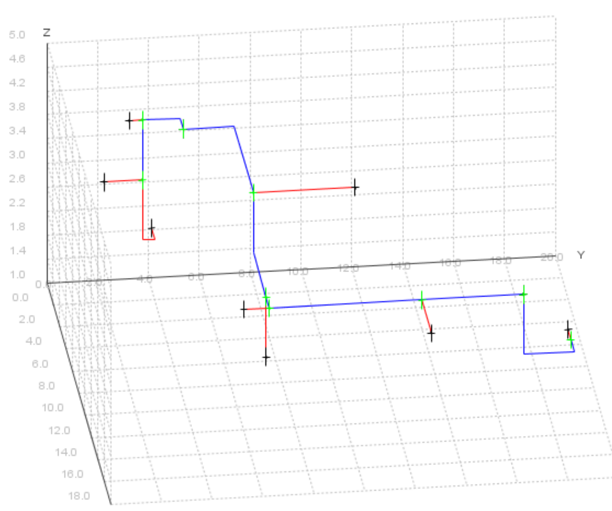

Fig 6 with factor= 1.3, pt set 1

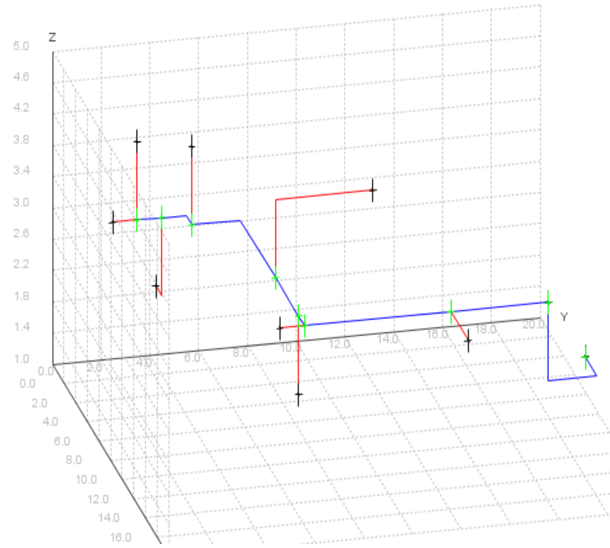

Fig 7 with factor= 4.1, pt set 1

C. The Algorithm

Some of the noteworthy points of observation are as follows

- The algorithm simply connects the site point to the nearest curve point in each iteration. In case another site point is very near to it, the algorithm takes care of this by placing a curve point very close to it. This shows the efficiency of the algorithm. This property can be further used to decrease the no. of Steiner points in the final solution.
- In case of RSTM, a connections can be made not only to the curve/site points directly but to a connection between some other pair of points which is nearer. This additional computation makes RSTM algorithm slightly slower than the ESTM one.
- The spanning tree used for comparison of length has been generated using the Prim's Algorithm [4]
- Using beta approximation (5) in case of ESTM leads to faster termination of the algorithm with lesser no. of Steiner points but longer final lengths.
- The RSMT lengths are less than the corresponding MST in cases when number points are small. For larger no. of points, the answer seems to deviate. This can be rectified by clustering the site points into clusters of 10-12 points and connecting the individual solutions.
- The no. of iterations done to obtain the final length of RSMT depends on the value of the *threshold*. In the above runs, the iterations stopped when the change in length for 1000 iterations was less than 0.1 units. A higher starting step size (1 unit was taken here) or higher *threshold* value could be taken for faster execution at the cost of better results.

D. Algorithmic Complexity

The space complexity is O(n) as "4xn" curve points are generated for "n" site points. The time taken for the algorithm to run depends on the value of threshold set for the iterations. If its smaller, it takes a longer time. (The difference in length in 2 consecutive iterations is taken and if the changes are below the threshold value, iterations stop).

VI. CONCLUSION AND FUTURE WORK

In this paper we have presented generation of rectilinear Steiner minimal tree in 3D using neural self-organization. This can be used for VLSI/ULSI interconnection problem. For making the methodology robust, we need to partition the point set into smaller sets. The solutions for the smaller sets can be concatenated to generate the overall solution for the whole point set. We also need to find solutions with obstacles and with connectivity constraints on the point sets.

REFERENCES

[1] Jayadeva, Optimization with Neural Networks, Phd Dissertation, IIT Delhi, June,1993

[2] Jayadeva and Bhaumik B, A Neural Network for the Steiner Minimal Tree Problem, Biological Cybernetics, 1994, 70: 485-494

[3] Atanendu Sekhar Mandal and Basabi Bhaumik,, Techniques for improving the neural optimization method, Proceedings Progress in VLSI Design & Test (VDAT-03),2003, Bangalore, ed. By C.P. Ravikumar, pp.

[4] R..C. Prim. Shortest connection networks and some generalizations. Bell System Technical Journal 36:1389-1401! 1957.

[5] Xiuzhen Cheng, Ding-Zhu Du and Hung Q. Ngo, Steiner Trees in industries – Handbook of Combinatorial Optimization Vol5, Kluwer Academic Publishers, 2002

[6] Olga Russakovsky and Andrew Y. Ng. A Steiner tree approach to efficient object detection. Poster at CVPR-2010

[7] Ihler, E. , The Complexity of Approximating the Class Steiner Tree Problem. WG'91, 1992.

[8] M. R. Garey and D. *S.* .Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal of Applied Mathematics:* 32(4):826-834: 1977.

[9] Qi Zhu, Hai Zhou, Tong Jing, Xian-Long Hong and Yang Yang. Spanning Graph-Based Nonrectilinear Steiner Tree Algorithms :IEEE Transactions on CAD of Integrated Circuit Systems, Vol. 24, No. 7, July 2005

[10] M. Sarrafzadeh and C. K.Wong. "Hierarchical Steiner tree construction in uniform orientations," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst*, vol. 11, no. 9, pp. 1095–1103, Sep. 1992.

[11]J. MacGregor Smith, D. T. Lee, and Judith S. Liebman. An O(n log n) heuristic algorithm for the rectilinear Steiner minimal tree problem. *Engineering Optimization*, 4:179-192,1980.

[12] Andrew B. Kahng. A steiner tree construction for vlsi routing. In *Proceedings of the International Joint Conference on Neural Networks*, volume I: pages 133-139, Piscataway, NJ, 1991 IEEE press

[13] Bryant A. Julstrom. Encoding rectilinear Steiner trees as lists of edges. In Gary B. Lamont: Evalgelos A. Yfantis, Hisham Haddad, George A. Papadopoulos, and Jan Carroll, editors, *Proceedings of the 16th ACM Symposium on Applied Computing*, pages 356-360: New York, 2001. ACM Press.

[14] Shin'ichi Wakabayashi. A genetic algorithm for generating a set of rectilinear Steiner trees in VLSI interconnect layout. In Proceedings of the International Conference on Chip Design Automation, Beijing, August 2000.

[15] Yu Hu, Tong Jing, Zhe Feng, Xian-Long Hong, Xiao-Dong Hu, Gui-Ying Yan. Ant Colony optimization based Rectilinear Steiner Minimal Tree Algorithm. Journal of Comp Sci and Technology. Jan 2006 Vol21

[16]M. Dorigo, V. Maniezzo, et A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics--Part B, volume 26, numéro 1, pages 29-41, 1996.

[17] Jin Zhongqi, Shu Huaying, Guo Hong ming. Using Neural Networks to solve Steiner Tree Problem. IEEE TENCON-93/ Beijing

[18]Pornavalai, C., Chakraborty, G., Shiratori, N. Neural Networks for Solving Constrained Steiner Tree Problem. Proceedings., IEEE International Conference on Neural Networks. 1995