Obstacle-Avoiding Octagonal Steiner Tree Construction Based on Particle Swarm Optimization

Xing Huang[1]. Genggeng Liu[1]. Wenzhong Guo[1]. Guolong Chen[1,2]* .
[1]College of Mathematics and Computer Sciences, Fuzhou University, Fuzhou 350108, China
fzu_hx@163.com, liu_genggeng@126.com, { guowenzhong, cgl } @fzu.edu.cn
[2] Key Laboratory of Discrete Mathematics with Applications of Ministry of Education, Fuzhou University, Fuzhou 3501028, China

Abstract—**The Obstacle-Avoiding Octagonal Steiner Minimal Tree (OAOSMT) problem has been proved to be a NP-hard problem, and it is very important for Very Large Scale Integration (VLSI) Non-Manhattan routing problem. In this paper, we present an efficient algorithm called OA_OST_DPSO for Obstacle-Avoiding Octagonal Steiner Minimal Tree construction. In our algorithm, according to the feature of OAOSMT problem, we construct an appropriate particle coding model and a fitness evaluation function. Meanwhile we integrate the crossover and mutation operator of genetic algorithm (GA) into the Particle Swarm Optimization (PSO) algorithm. Besides, edge transformation is employed to make the particles have the ability to achieve the more optimal solution while Union-Find partition is used to prevent the generation of invalid solution. Finally, our algorithm uses a dynamic parameter strategy, in order to make sure that the search is more efficient. The experimental results show that our algorithm has a good performance, can in the presence of obstacle, greatly shorten the length of the wiring.**

Keywords-VLSI routing; obstacle-avoiding; octagonal Steiner tree; PSO

I. INTRODUCTION

VLSI physical design is a complicated process, routing is one of the most critical steps. In recent years, the octagonal Steiner minimal tree problem has been always a focus problem in the global routing. Many construction methods have been proposed so far. But to our knowledge, there are very few to consider avoiding obstacle in the process of constructing the octagonal Steiner minimal tree.

Obstacle-Avoiding Octagonal Steiner Minimal tree problem is to seek a tree which connects all the given points in the routing plane with additional points (here are called the Steiner point), at the same time bypass all of the obstacles with the minimum cost. In order to solve this problem, the first step is to construct the octagonal Steiner minimal tree. An exact algorithm and a variety of pruning techniques were introduced to construct the octagonal Steiner minimal tree [1]. An algorithm whose time complexity is $O(|V|+|E|)$ was proposed to build an octagonal Steiner minimal tree which must be isomorphic to the given rectilinear Steiner tree [2]. Here $|V|$ and $|E|$ are the vertices and the edges of the given tree. As for obstacle-avoiding, so far, Chang et al. (2008)[3] presented a construction-by-correction approach to solve the Obstacle-Avoiding Rectilinear Steiner Minimal Tree (OARSMT) construction problem, which build an obstacle-weighted spanning tree as a guidance to construct OARSMT on an escape graph. A heuristic maze routing based approach to solve the OARSMT problem was proposed by Li et al. (2008) [4], which based on the searching process as in maze routing and can handle multi-pin nets very well in both solution quality and memory space usage. In 2009, an algorithm based on the Geo Steiner approach, modified and extended to allow rectilinear blockages in the routing region were presented, which could generate an optimal solution in a reasonable amount of time [5]. Unlike previous works usually build a spanning graph and find a spanning tree to deal with this problem, Chuang et al. proposed a novel algorithm which first determine a connection ordering for all pins, and then iteratively connect every two neighboring pins by a greedy heuristic algorithm [6]. Not long ago, Liu et al. applied computational geometry techniques developed an efficient algorithm, which could obtain a solution with excellent quality and has a significant speed-up compared to previously know results [7]. However, the method mentioned above is all based on Manhattan Steiner tree, can't get a good optimization in terms of wiring length.

As a swarm-based evolutionary method, PSO which was introduced by Eberhart and Kennedy in 1995 [8] has been proved to be a global optimization algorithm. PSO algorithm has quick convergence, global search, stable, efficient, and many other advantages. A large number of practices have proved that compared with other methods, PSO can get a cheaper optimal solution on global optimization problem.

Aiming at the aspects concerned above, this paper employs an efficient algorithm based on discrete PSO (DPSO) to solve the obstacle-avoiding octagonal Steiner tree construction problem. In order to solve the problem of the slow convergence rate of PSO used for a high-dimensional space optimization, a self-adapting strategy that can adjust the learning factors, and combine with the crossover and mutation operators of genetic algorithm (GA) is proposed. Moreover, edge transformation is employed to make the particles have the ability to achieve the more optimal solution while Union-Find partition is used to prevent the generation of invalid solution. The experimental results show that the proposed approach is effective and feasible.

The rest of this paper is organized as follows: In Section II, we will give the model of the OAOSMT problem, some basic definitions, and the basic PSO. In Section III, the OA_OST_DPSO algorithm will be described in detail.

This work was supported in part by the National Basic Research Program of China under Grant No. 2011CB808003, the National Natural Science Foundation of China under Grant No. 11141005 and No.11271002, and the Technology Innovation Platform Project of Fujian Province under Grant No.2009J1007.

Section IV gives the experimental results. Section V summarizes our work and proposes possible future work.

II. PRELIMINARIES

A. OAOSMT Problem Model

OAOSMT problem is to connect all the given points in the routing plane with Steiner point, meanwhile bypass all of the obstacles with the minimum cost. In addition to traditional allows only horizontal and vertical line direction, here we introduce the 45 degree and 135 degree line direction.

The OAOSMT problem can be described as follows: Let $P= \{P_1, P_2, P_3 \ldots P_n\}$ is a group of the pins in the net, where each Pi is assigned with its coordinate (x_i, y_i). Let $B= \{ B_1, B_2, B_3 \ldots B_n \}$ is a set of obstacles on the chip, where each Bi associated with two coordinates (X_{i1}, Y_{i1}) and (X_{i2}, Y_{i2}), where (X_{i1}, Y_{i1}) is the lower left corner of the rectangular obstacle's vertex coordinates, (X_{i2}, Y_{i2}) is the top right-hand corner of the rectangular obstacle's vertex coordinates. The input information of pins and obstacles is listed in TABLE I and TABLE II respectively. Each pin has a corresponding coordinate (x_i, y_i). Each obstacle has two corresponding coordinate (X_{i1}, Y_{i1}) and (X_{i2}, Y_{i2}), for example, the pin1 is located at(1, 2), the obstacle1 located at (3, 9) and (5, 14).

TABLE I INPUT INFORMATION OF EACH PIN

Number	1	2	3	4	5	6	7	8	9	10	11	12
X_label	1	2	3	4	3	4	7	8	8	10	11	12
Y_label	2	15	9	5	2	6	8	10	1	8	3	9

TABLE II INPUT INFORMATION OF EACH OBSTACLE

Number	1	2	3	4
Lower_left_X	3	6	10	9
Lower_left_Y	9	4	4	9
Top_right_X	5	9	14	14
Top_right_Y	14	5	5	11

Definition 1: In the λ-geometry, the routing direction is $i\pi/\lambda$, where i is an arbitrary integer and λ is an integer.

1) Manhattan architecture: The value of λ is set to 2. The routing direction is $i\pi/2$, which includes 0° and 90°.

2) Octagonal interconnect architecture: The value of λ is set to 4. The routing direction is iπ/4, which includes 0°, 45°, 90° and 135°.

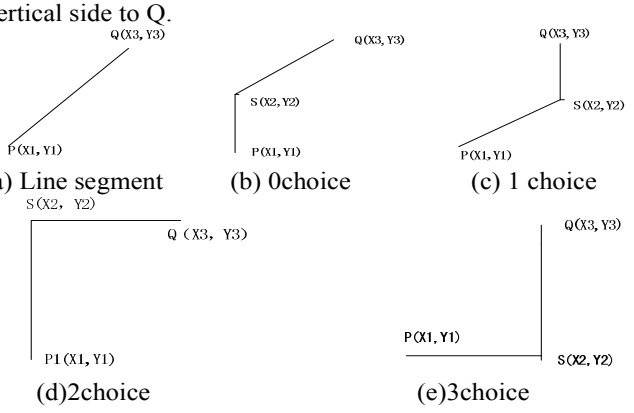
Definition 2 (Pseudo-Steiner): For convenience, we assume that the endpoints except for the pins collectively referred to as pseudo-Steiner points.

Definition 3 (0 Choice): In Figure 1 (a), let $P=(x1, y1)$ and $Q=(x3, y3)$ be the two endpoints of a line segment L, $x1<x2$. The 0 choice of pseudo-Steiner point corresponding to edge L is given in Figure 1(b), which first from P leads rectilinear side to pseudo-Steiner point then leads non-rectilinear side to Q.

Definition 4 (1 Choice): The 1 choice of pseudo-Steiner point corresponding to edge L, as shown in Figure 1 (c), which first from P leads non-rectilinear side to pseudo-Steiner point then leads rectilinear side to Q.

Definition 5 (2 Choice): The 2 choice of pseudo-Steiner point corresponding to edge L is given in Figure 1 (d), which first from P leads vertical side to pseudo-Steiner point then leads horizontal side to Q.

Definition 6 (3 Choice): The 3 choice of pseudo-Steiner point corresponding to edge L, as shown in Figure 1 (e), which first from P leads horizontal side to pseudo-Steiner point then leads vertical side to Q.



(a) Line segment (b) 0choice (c) 1 choice

(d)2choice (e)3choice

Figure 1. Four options of Steiner point

Definition 7(Connectivity model): In routing model, the existence of the obstacles may inhibit some points from selecting as Steiner point, but each pin can link with the other through no more than one time of 0 or 1, or 2, or 3 choice, as shown in Figure 2, we call it connectivity model, where A (X_1, Y_1) and B (X_2, Y_2) are two points in the routing area.

Definition 8(Disconnected model): In routing model, there is one pin at least by two or more times of 0 or 1, or 2, or 3 choice can be connected to the other pin, as shown in Figure 3, we call it disconnected model.

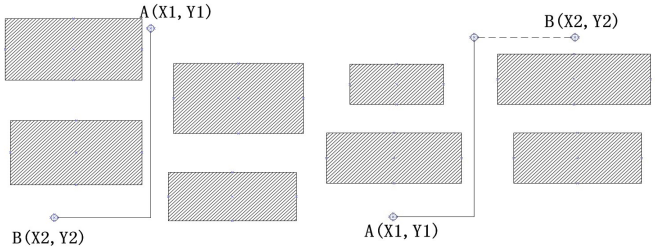


Figure 2. Connectivity model

Figure 3. Disconnected model

B. Basic PSO

Particle swarm optimization algorithm is an intelligent algorithm based on population. Therefore, we consider a population S contains N particles. Each particle with a D continuous space, and has its own speed and position, the current position of the i-th particle can be expressed as the $X_i = (X_{i1}, X_{i2} \ldots X_{iD})$, the current speed of the i-th a particle can be

represented as $V_i = (V_{i1}, V_{i2} \ldots V_{iD})$, we mark the optimal particle of population S as g_j. The best previous position of the i-th particle is recorded and represented as $P_i=(P_{i1}, P_{i....}P_{iD})$, Update formula for the particle position and velocity can be expressed as follows:

$$v_{ij}^{t} = w v_{ij}^{t-1} + c_1 r_1 (p_{ij} - x_{ij}^{t-1}) + c_2 r_2 (g_j - x_{ij}^{t-1}) \quad (1)$$

$$x_{ij}^{t} = x_{ij}^{t-1} + v_{ij}^{t} \quad (2)$$

where w is the inertia weight, c_1 and c_2 is acceleration coefficient, r_1 and r_2 are both random numbers on the interval [0, 1].

III. OUR ALGORITHM

The initial prototype of PSO is used to solve continuous optimization problems, but since PSO algorithm has been proposed, a growing number of researchers have tried to use PSO to solve the discrete problem. For example, Clerc proposed a DPSO algorithm for solving TSP [9]. In our previous work, we presented an OSMT algorithm based on discrete PSO to optimize the wire length [10], Moreover, we presented a RSMT algorithm based on DPSO to minimize the wiring length and reduce the number of bends [11].

A. Coding of the Particle

In this paper, we use a set of minimum spanning tree edges to represent a candidate routing tree, and add one extra bit to each edge indicates the option of the Steiner point choice in order to subsequently transform it into octagonal Steiner tree. Each pseudo-Steiner point choice includes four types which are shown in Definition 3-6. if there are n pins in the net, the minimum spanning tree will include $n-1$ edges (two bit) , $n-1$ Steiner point and one extra bit which is the particle's fitness value, so the length of the particle is $3*(n-1)+1$. We take an example, one particle can be expressed as the following numeric string: 1 2 3 2 3 0 3 4 1 4 5 2 23. 7, where 23.7 represents the fitness value of the particle, the first string 1 2 3 represents one edge which includes pin 1 and pin 2 according to the third choice to connect.

In particular, if there is a disconnected model (Definition 8), our algorithm will automatically find an inflection point from the nearby obstacles as an additional Steiner points, in this way, the original disconnected model becomes a Connectivity model, the principle is shown in Figure 4.

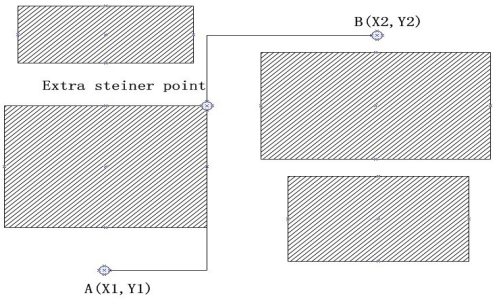


Figure 4. Additional Steiner points

B. The Fitness Function of the Particle

Definition 9：The length of the octagonal Steiner tree is the sum of each edge's length which is formulated as:

$$Length\ (T_S) = \sum_{e_i \in T_s} l(e_i) \quad (3)$$

where $l(e_i)$ is the length of edge e_i in the tree T_S.

The fitness function of the particle can defined as follow:

$$Fitness\_value = Length(T_S) * (T\_count + 1) \quad (4)$$

where T_count is the number of the edge which belongs to the Ts but though the obstacles.

C. Update Formula of the Particle

We employ the new discrete position updating method based on genetic operations and Union-Find partition. The update formula of the particle is represented as:

$$X_i^t = F_3(F_2(F_1(X_i^{t-1}, w), c_1), c_2) \quad (5)$$

where w is inertia weight, c_1 and c_2 is acceleration constant. F_1 denotes the mutation operation and F_2, F_3 denote the crossover operation. Here we assume that r_1, r_2, r_3 are random number on the interval [0, 1).

1) The velocity of particles can be written as

$$V_i^t = F_1(X_i^{t-1}, w) = \begin{cases} M(X_i^{t-1}), r_1 < w \\ X_i^{t-1}, others \end{cases} \quad (6)$$

where w denotes the mutation probability.

2) The cognitive personal experience of particles can be written as

$$C_i^t = F_2(H_i^t, c_1) = \begin{cases} C_p(H_i^t), r_2 < c_1 \\ H_i^t, others \end{cases} \quad (7)$$

where c_1 denotes the crossover probability of the particles and the personal optimal solution.

3) The cooperative global experience of particles can be written as

$$X_i^t = F_3(C_i^t, c_2) = \begin{cases} C_p(C_i^t), r_2 < c_1 \\ C_i^t, others \end{cases} \quad (8)$$

where c_2 denotes the crossover probability of the particles and the global optimal solution.

D. The Union-Find Partition in Genetic Operations

- Mutation operation

Mutation operation randomly deletes an edge from the spanning tree, and replaces it with another randomly generated edge in order to make sure that the spanning tree is connected. We use a Union-Find Partition to record all the rest of points as two sets after one edge is deleted, and randomly selects a point from the two point sets respectively, then connects the two points to form a new spanning tree. The principle is shown in Figure 5 where the red line represents the edge to be removed, and blue line represents the new edge.

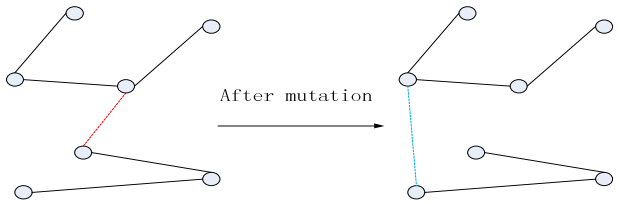


Figure 5. Mutation operation with Union-Find Partition

- Crossover operation

As mentioned in section II, each pin has a serial number. when carrying on the crossover operation, we sort the edges according to the serial number of pins from small to large order in both of the two spanning tree, then by using Union-Find Partition to choose the same edge from the two sorted tree as a set, the rest of the different edges as another set, and the first set of edges directly as the new spanning tree's edge. Finally, we select one edge randomly from the second set and add to the new spanning tree, while we prevent generating circle by using union-find partition. The principle is shown in Figure 6 where red lines represent the different edge between the two spanning tree, green lines represent the new edge of the new spanning tree.

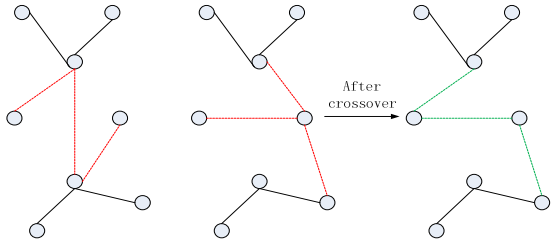


Figure 6. Crossover operation with Union-Find Partition

E. The Strategy of the Parameters

The parameters in the proposed algorithm were given as follows: population size is 200, w decreased linearly from 0.95 to 0.05, c_1 decreased linearly from 0.9 to 0.2, c_2 is equal to 1 minus the c_1, and the maximum number of generations is 2000. w, c_1 and c_2 is updated according to (9), (10), (11).

$$w\_new = w\_start - \frac{w\_start - w\_end}{evaluations} \times eval \quad (9)$$

$$c_1\_new = c_1\_start - \frac{c_1\_start - c_1\_end}{evaluations} \times eval \quad (10)$$

$$c_2\_new = 1 - c_1\_new \quad (11)$$

where *eval* represents the current iteration and *evaluations* represents the maximum number of iterations.

F. Processes of the OA_OST_DPSO

Step1: Initialize various parameters and obstacles, and randomly generated the initial population.

Step2: Calculate the fitness value of each particle according to (4). And initialize the personal optimal solution of each particle and the global optimal solution of the population.

Step3: Adjust the position and velocity of each particle according to (5)-(8).

Step4: Recalculate the fitness value of each particle and update the personal optimal solution of each particle.

Step5: Recalculate the global optimal solution of the population.

Step6: Check the termination condition (a good enough position or the maximum number of iterations is reached). If fulfilled, the run is terminated. Otherwise, go to Step3.

IV. EXPERIMENT RESULT

We use C language to implement our algorithm, and use matlab 7.0 to draw the final routing diagram. Our algorithm runs sequentially on a single processor. There are 10 benchmarks which commonly used as test case for the OAOSMT problem. IND1-IND5 is industrial test case from Synopsys. RC0-RC1 is benchmark of the obstacle-avoiding problem. Note, pins may be along the boundary of the obstacle, and there may be overlap between obstacles, we regard it as one obstacle. TABLE III compares the results of our algorithm with some heuristics algorithm [12, 13, 4, 14, 15, and 16] which was published recently. In TABLE III, benchmark represents the test case, P represents the number of the pins, B represents the number of obstacles, Our_L stands for the length of the routing tree which our program generated. Our result is the optimal value of ten times running. Other experiment results references from relevant papers. Adding * indicates the group of test case include disconnected model which has described in Definition 8, our algorithm will automatically find an inflection point from the nearby obstacles as an additional Steiner points. From TABLE III, we can see that compared with other algorithm, we get a more larger optimization in the aspect of wire length, the maximum compression ratio can reach 9.6%, the highest average compression ratio of the ten group test cases can reach 5.1%.

As mentioned above, the experiment has obtained a great result, as a representative, we give the final routing diagram of RC05 and IND5, which is as shown in Figure 7 and Figure 8. We can see from Figure 7 and Figure 8 that the final routing tree has successfully connected all the pins, and bypassed all the obstacles at the same time.

V. CONCLUSIONS AND FUTURE WORK

This paper presents an efficient algorithm called OA_OST_DPSO based on discrete PSO for obstacle-avoiding octagonal Steiner Minimal tree construction to optimize the wiring length in VLSI global routing stage. The algorithm has been implemented in Visual C++ 6.0 and matlab 7.0. Ten groups of test cases and some random instances have been tested, the results are very promising. Compared with the existing work, the maximum reduction ratio that we can get is 5.1% on average. However, there are some issues must be explored in the future, for example: 1) we can consider the obstacle with any shape. 2) We can present some local search strategies to improve the running speed.

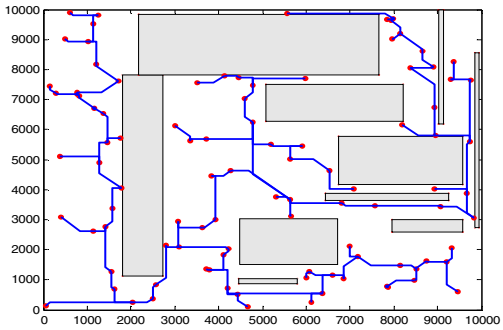


Figure 7. The routing diagram of RC05

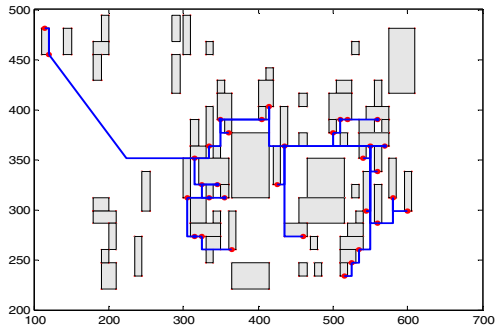


Figure 8. The routing diagram of IND5

TABLE III COMPARISION OF EXPERIMENTAL RESULT

benchmark	P	B	Our_L	Wire_length						Reduction (X-Our_L)/X(%)					
				[12]A	[13]B	[4]C	[14]D	[15]E	[16]F	X=A	X=B	X=C	X=D	X=E	X=F
IND1	10	32	579	632	639	619	626	604	604	8.4	9.4	6.5	7.5	4.1	4.1
IND2*	10	43	9489	9700	10000	9500	9700	9600	9500	2.2	5.1	0.1	2.2	1.2	0.1
IND3	10	59	588	632	623	600	600	600	600	7.0	5.6	2.0	2.0	2.0	2.0
IND4	25	79	1085	1121	1130	1096	1095	1092	1129	3.2	4.0	1.0	0.9	0.6	3.9
IND5*	33	71	1321	1392	1379	1360	1364	1374	1364	5.1	4.2	2.9	3.2	3.9	3.2
RC01*	10	10	25130	27790	26120	25980	26740	26040	25980	9.6	3.8	3.3	6.0	3.5	3.3
RC02	30	10	40924	42240	41630	42010	42070	41570	42110	3.2	1.7	2.6	2.7	1.6	4.1
RC03*	50	10	54039	56140	55010	54390	54550	54620	56030	5.1	3.1	2.0	2.3	2.5	2.8
RC04	70	10	58536	60800	59250	59740	59390	59860	59720	3.7	1.2	2.0	1.4	2.2	2.0
RC05	100	10	74063	76760	76240	74650	75440	74770	75000	3.5	2.9	0.9	1.8	1.0	1.3
average										5.1	4.1	2.3	3.0	2.3	2.7

REFERENCES

[1] G. C. S. Coulston, "Constructing exact octagonal Steiner minimal trees." in Proc. 13th ACM Great Lakes symposium on VLSI, Great Lakes, 2003, pp. 1-6.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Chiang, Charles, and Ching-Shoei Chiang. "Octilinear Steiner tree construction." Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on. Vol. 1. IEEE, 2002.

[3] Chang, Yung-Tai, et al. "Obstacle-Avoiding Rectilinear Steiner Minimal Tree construction." VLSI Design, Automation and Test, 2008. VLSI-DAT 2008. IEEE International Symposium on. IEEE, 2008.

[4] Li L, Young E F Y, "Obstacle-avoiding rectilinear Steiner tree construction." Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design. IEEE Press, 2008: 523-528. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[5] L.Li, Z. Qian, and Evangeline F. Y. Young, "Generation of Optimal Obstacle-avoiding Rectilinear Steiner Minimum Tree." in Proc. Int. Conf. Comput.-Aided Des, pp.21-25, 2009.

[6] Chuang, Jia-Ru, and Jai-Ming Lin. "Efficient multi-layer obstacle-avoiding preferred direction rectilinear Steiner tree construction." Proceedings of the 16th Asia and South Pacific Design Automation Conference. IEEE Press, 2011.

[7] Liu, Chih-Hung, I-Che Chen, and D. T. Lee. "An efficient algorithm for multi-layer obstacle-avoiding rectilinear Steiner tree construction." Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE. IEEE, 2012.

[8] Eberhart, Russell, and James Kennedy. "A new optimizer using particle swarm theory." Micro Machine and Human Science, 1995. MHS'95. Proceedings of the Sixth International Symposium on. IEEE, 1995.

[9] Clerc, Maurice. "Discrete particle swarm optimization, illustrated by the traveling salesman problem." New optimization techniques in engineering. Springer Berlin Heidelberg, 2004. 219-239.

[10] G.G. Liu, G.L. Chen, W.Z. Guo, et al, "DPSO Based Octagonal Steiner Tree Algorithm for VLSI Routing." 2012 IEEE fifth International Conference on Advanced Computational Intelligence(ICACI), October 18-20, 2012.

[11] G.G. Liu, G.L. Chen, W.Z. Guo, et al, "DPSO-based rectilinear Steiner minimal tree construction considering bend reduction." in Proc. 7th Int.Conf. Natural Computation, 2011, pp. 1161-1165.

[12] Lin, Chung-Wei, et al. "Obstacle-avoiding rectilinear Steiner tree construction based on spanning graphs." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 27.4 (2008): 643-653.

[13] J. Long, H. Zhou, and S. O. Memik, "EBOARST: An efficient edge based obstacle avoiding-rectilinear Steiner tree construction algorithm." IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 27, no. 12, pp. 2169–2182, Dec. 2008.

[14] C.-H. Liu, S.-Y. Yuan, S.-Y. Kuo, and Y.-H. Chou,"An O(n log n) pathbased obstacle-avoiding algorithm for rectilinear Steiner tree construction." in Proc. DAC, 2009, pp. 314–319.

[15] C.-H. Liu, S.-Y. Yuan, S.-Y. Kuo, and J.-H. Weng, "Obstacle-avoiding rectilinear Steiner tree construction based on Steiner point selection." in Proc. ICCAD, 2009, pp. 26–32.

[16] G. Ajwani, C. Chu, and W.-K. Mak, "FOARS: FLUTE based obstacle avoiding rectilinear Steiner tree construction." in Proc. ISPD, 2010, pp.185–192.