2012 IEEE fifth International Conference on Advanced Computational Intelligence(ICACI)
October 18-20, 2012 Nanjing, Jiangsu, China

DPSO Based Octagonal Steiner Tree Algorithm for VLSI Routing

Genggeng Liu, Guolong Chen and Wenzhong Guo

***Abstract*—The Octagonal Steiner Minimal Tree (OSMT) problem is an NP-hard problem, which is one of the key problems in non-Manhattan routing. Particle Swarm Optimization (PSO) has been proved to be an efficient intelligent algorithm for optimization designs. This paper presents an OSMT algorithm based on discrete PSO (DPSO), namely OSMT_DPSO, to optimize the wire length. 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. The experimental results show that the proposed algorithm can efficiently provide the solution of OSMT problem with good quality. Moreover, the algorithm can obtain several topologies of OSMTs which is beneficial for optimizing congestion in VLSI global routing stage.**

I. INTRODUCTION

ROUTING is an important step in VLSI physical design. Many routing algorithms which are mostly based on Manhattan architecture have been proposed. With advance in VLSI fabrication technology, the interconnect effects become the major challenge of chip performance. Manhattan architecture which restricted the routing to only horizontal and vertical directions has the limited ability to optimize wire length. Consequently, there are more and more interests in non-Manhattan routing which allows more routing directions.

The construction of Steiner Minimum Tree (SMT), which seeks to connect the given points in the plane with additional points (here called Steiner points) in the minimum cost, is one of the key problems in VLSI routing. In order to study non-Manhattan routing, the first work is to construct the SMT in non-Manhattan architecture. Non-Manhattan Steiner tree includes Octagonal Steiner tree (OST) and Hexagonal Steiner tree. An exact algorithm and a variety of pruning techniques were introduced to construct the Octagonal Steiner Minimal Tree [1]. An O ($|V|+|E|$) algorithm was proposed to build an Octagonal Steiner 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, respectively. The spanning graph-based Octagonal Steiner tree algorithms were presented in [3]. For constructing Hexagonal Steiner minimal tree, Samanta et al. proposed a heuristic method [4]. Because the SMT problem is an NP-hard problem, some evolutionary algorithms, which had been shown to have good application prospects in solving NP-hard problems, were presented for solving Rectilinear Steiner Minimal Tree (RSMT) problem [5], [6] and OSMT problem [6].

As a swarm-based evolutionary method, PSO which has been proved to be a global optimization algorithm was introduced by Eberhart and Kennedy in 1995 [7]. The advantages of PSO over many other optimization algorithms are its implementation simplicity and ability to converge to a reasonably good solution quickly. In the past several years, PSO has been successfully applied in many researches and application areas [5], [8]-[13]. It is demonstrated that PSO can get better results in a faster, cheaper way compared with other methods.

Aiming at the aspects concerned above, this paper employs an effective algorithm based on DPSO to solve the Octagonal Steiner tree 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, we develop a suitable encoding scheme for the Octagonal Steiner tree. The experimental results show that the proposed approach is effective and feasible.

The remainder of the paper is organized as follows. In Section II, we formulate the problem and introduce some basic definitions and the basic PSO. In Section III, the OSMT algorithm based on DPSO is introduced. Section IV gives the experimental results. Section V summarizes our work and proposes possible future work.

II. PRELIMINARIES

A. Problem Description

The OSMT problem is to connect all pins in the plane through Steiner points to achieve a minimal total length in VLSI routing. Here, it allows 45^0 and 135^0 routing directions in addition to traditional horizontal and vertical orientations [3].

The OSMT problem can be stated as follows. Let $P= \{P_1, P_2, P_3,\ldots,P_n\}$ be the set for the net of m pins , where each P_i is assigned with its coordinate (x_i, y_i). The input information for pins is listed in Table I. The layout is shown in Fig. 1. Each pin has the corresponding coordinate (x_i, y_i), e.g., the pin 1 is located at (33, 33).

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 the Technology Innovation Platform Project of Fujian Province under Grant No.2009J1007.

G Liu is with the College of Mathematics and Computer Sciences, FuzhouUniversity,Fuzhou350108,Fujian,PRC(e-mail:*liu_genggeng@126.com*).

G Chen, corresponding author, is with the College of Mathematics and Computer Sciences, Fuzhou University, Fuzhou 350108, Fujian, PRC (phone:0591-22865521, e-mail: *fzucgl@163.com*).

W Guo is with the College of Mathematics and Computer Sciences, Fuzhou University, Fuzhou 350108, China (e-mail: *guowenzhong @fzu.edu.cn*).

978-1-4673-1744-3/12/$31.00 ©2012 IEEE

TABLE I

INPUT INFORMATION FOR EIGHT PINS

Number	1	2	3	4	5	6	7	8
X_label	33	2	42	47	34	38	37	20
Y_label	33	9	35	2	1	2	5	4

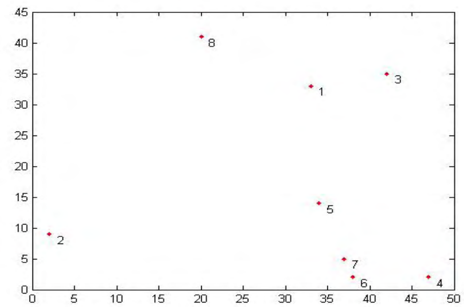

Fig. 1. The layout of the random instance

B. Definitions

Definition 1 In the λ-geometry, the routing direction is $i\pi/\lambda$, where i is an arbitrary integer and λ is an integer. Different routing directions are obtained with different values of i and λ.

1) Manhattan architecture: The value of λ is set to 2 i.e. the routing direction is $i\pi/2$, which includes 0^0 and 90^0, namely traditional horizontal and vertical orientations.

2) Hexagonal interconnect architecture: The value of λ is set to 3 i.e. the routing direction is $i\pi/3$, which includes 0^0, 60^0 and 120^0.

3) Octagonal interconnect architecture: The value of λ is set to 4 i.e. the routing direction is $i\pi/4$, which includes 0^0, 45^0, 90^0 and 135^0.

Both of *2)* and *3)* belong to non-Manhattan architecture.

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 Fig. 2. (a), let $A = (x_1, y_1)$ and $B = (x_2, y_2)$ be the two endpoints of a line segment L, $x_1<x_2$. The 0 choice of pseudo-Steiner point corresponding to edge L is given in Fig. 2. (b), which first from A leads rectilinear side to pseudo-Steiner point then leads non-rectilinear side to B.

Definition 4 (1 Choice): The 1 choice of pseudo-Steiner point corresponding to edge L, as shown in Fig. 2. (c), which first from A leads non-rectilinear side to pseudo-Steiner point then leads rectilinear side to B.

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

Definition 6 (3 Choice): The 3 choice of pseudo-Steiner point corresponding to edge L, as shown in Fig. 2. (e), which first from A leads horizontal side to pseudo-Steiner point then leads vertical side *to B*.

C. Basic PSO

PSO is a swarm intelligence method, which considers a swarm S containing n particles in a D-dimensional continuous solution space. Each i-th particle has its own position and velocity. Assuming that the search space is D-dimensional, the position of the i-th particle denoted as a D-dimensional vector: $X_i = (X_{i1}, X_{i2}, \ldots, X_{iD})$ and the best particle in the swarm S is denoted by the index g. The best previous position of the i-th particle is recorded and represented as $P_i = (P_{i1}, P_{i2}, \ldots, P_{iD})$, while the velocity for the i-th particle can be defined by another D-dimensional vector: $V_i = (V_{i1}, V_{i2}, \ldots, V_{iD})$. According to these definitions, the particle position and velocity can be manipulated according to the following equations:

$$V_i^{t+1} = w \times V_i^t + c_1 r_1 (P_{pb}^t - X_i^t) + c_2 r_2 (P_{gb}^t - X_i^t) \quad (1)$$

$$X_i^{t+1} = X_i^t + V_i^t \quad (2)$$

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

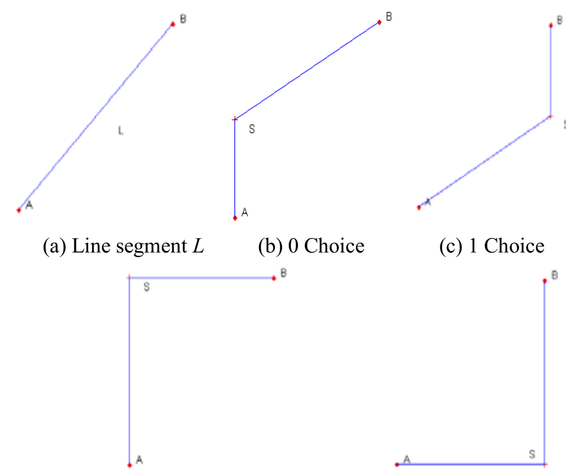

Fig. 2. Four options approach of Steiner point

III. OSMT_DPSO

Since the PSO algorithm was propose, more and more researchers attempted to construct the PSO algorithm for solving the discrete problem. Kennedy and Eberhart proposed a discrete binary PSO algorithm [14]. Clerc proposed a DPSO algorithm for solving TSP [11]. In previous work, a novel intelligent decision algorithm based on the particle swarm optimization technique was proposed to obtain a feasible floorplanning in VLSI circuit physical placement [12]. In [13], we constructed an effective DPSO algorithm to solve the multi-objective minimum spanning tree problem for VLSI wire length estimation. In [8], an improved Discrete Particle Swarm Optimization algorithm was proposed for solving the obstacle-avoiding rectilinear Steiner minimum tree problem optimally. In [5], we presented a RSMT algorithm based on DPSO to minimize the wiring length and reduce the number of bends. Based on [13] and [5], an improved DPSO algorithm was designed here to solve OSMT construction, namely,

OSMT _DPSO.

A. Coding Of The Particle

Our algorithm represents candidate OST as lists of spanning tree edges, each augmented with a pseudo-Steiner point choice so that it specifies the conversion from the spanning tree edge to the Octagonal edge. Each pseudo-Steiner point choice includes four types shown in *Definition3-6*. If a net has *n* pins, a spanning tree would have *n*-1 edges, *n*-1 pseudo-Steiner points and one extra bit which is the particle's fitness. Besides, two bits represent the two vertices of each edge, so the length of a particle is 3*(*n*-1) +1. For example, one OST tree can be expressed as one particle whose code can be expressed as the following numeric string:

7 6 *0* 6 4 *1* 7 5 *1* 5 1 *2* 1 3 *0* 1 8 *1* 5 2 *2* 0.0100

Where the number '0.0100' is the fitness of the particle and each bold figure is the pseudo-Steiner point choice. The first substring (7, 6, 0) represents one edge of the spanning tree which composed of *Vertex 7* and *Vertex 6* and the pseudo-Steiner point choice which is 0 Choice in *Definition 3*.

B. The Fitness Function Of The Particle

Definition 7: The length of the Octagonal Steiner tree is the sum of each segment's length which is formulated as:

$$L(T_X) = \sum_{e_i \in T_x} l(e_i) \tag{3}$$

Where $l\,(e_i)$ represents the length of each segment e_i in the tree T_x.

When calculating the sum of each segment's length in OST, we divide all the segments into four categories: horizontal side, vertical side, hypotenuse with 45^0, and hypotenuse with 135^0. Then we clockwise rotated hypotenuse with 45^0 into a horizontal side and hypotenuse with 135^0 into a vertical side. We sorts the horizontal sides from the bottom up and from left to right by their left vertex, while sorting the vertical sides from left to right and from bottom up by their bottom vertex [15], and then the length of OST is the sum of those segments.

The proposed fitness function of the particle formulation is

$$\text{fitness } = 1/(L(T_X)+1) \tag{4}$$

Here denominator plus 1 prevent the possible circumstance that the denominator value is 0.

C. Update Formula Of The Particle

We employ the new discrete position updating method based on genetic operations and propose the OSMT_DPSO algorithm for OSMT construction.

The update formula of the particle is represented as:

$$X_i^t = N_3(N_2(N_1(X_i^{t-1}, w), c_1), c_2) \tag{5}$$

Where w is inertia weight, c_1 and c_2 are acceleration constants. N_1 denotes the mutation operation and N_2, N_3 denote the crossover operation. Here we assume that r_1, r_2, r_3 are random numbers on the interval [0, 1).

1) The velocity of particles can be written as

$$W_i^t = N_1(X_i^{t-1}, w) = \begin{cases} M(X_i^{t-1}), & r_1 < w \\ X_i^{t-1}, & \text{others} \end{cases} \tag{6}$$

Where w denotes the mutation probability.

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

$$S_i^t = N_2(W_i^t, c_1) = \begin{cases} C_p(W_i^t), & r_2 < c_1 \\ W_i^t, & \text{others} \end{cases} \tag{7}$$

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

$$X_i^t = N_3(S_i^t, c_2) = \begin{cases} C_p(S_i^t), & r_3 < c_2 \\ S_i^t, & \text{others} \end{cases} \tag{8}$$

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

D. Processes Of OSMT_DPSO

The detail procedure of OSMT_DPSO can be summarized as follows:

Step1: Initialize various parameters 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. Experimental Results

The parameters in the proposed algorithm were given as follows: population size was 50; w decreased linearly from 0.9 to 0.1, c_1 decreased linearly from 0.9 to 0.2, c_2 increased linearly from 0.4 to 0.9, the maximum number of generations was 500; w, c_1 and c_2 adopted the idea of linear decline proposed by Shi [16] and are updated according to (9), (10), (11) in each iteration.

$$w = w\_start - \frac{w\_start - w\_end}{evaluations} \times eval \tag{9}$$

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

$$c_2 = c_2\_start - \frac{c_2\_start - c_2\_end}{evaluations} \times eval \quad (11)$$

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

TABLE II

THE LENGTHS DIFFERENT FROM OSMT AND RSMT IN 46 TEST DATA SETS

Instance	*RSMT*	*OSMT*	*Reduction* (%)
1	1.87	1.7646	5.64
2	1.68	1.5664	6.76
3	2.36	2.1855	7.39
4	2.54	2.2382	11.88
5	2.29	2.1385	6.62
6	2.48	2.3285	3.79
7	2.54	2.3475	7.58
8	2.42	2.2685	6.26
9	1.72	1.6543	3.82
10	1.85	1.7057	7.80
11	1.44	1.3511	6.17
12	1.80	1.6828	6.51
13	1.50	1.3243	11.71
14	2.60	2.3657	9.01
15	1.48	1.2895	12.87
16	1.60	1.2485	21.97
17	2.01	1.6920	15.82
18	4.06	4.0566	0.08
19	1.93	1.8128	6.07
20	1.12	1.0685	4.60
21	2.16	1.9188	11.17
22	0.63	0.5363	14.87
23	0.65	0.5277	18.81
24	0.30	0.2680	10.67
25	0.23	0.2097	8.83
26	0.15	0.1290	14.00
27	1.33	1.2070	9.25
28	0.28	0.2156	23.00
29	2.00	1.4728	26.36
30	1.10	1.1000	0.00
31	2.66	2.4870	6.50
32	3.30	3.0381	7.94
33	2.69	2.3317	13.32
34	2.54	2.2682	10.70
35	1.57	1.4243	9.28
36	0.90	0.9000	0.00
37	0.90	0.8070	10.33
38	1.66	1.4808	10.80
39	1.66	1.4808	10.80
40	1.62	1.5204	6.15
41	2.24	2.0634	7.89
42	1.53	1.3833	9.59
43	2.68	2.5443	5.76
44	2.61	2.2910	12.22
45	2.26	2.0881	7.61
46	1.50	1.5000	0.00
Average	-----	---------	*9.84*

In order to validate the proposed algorithm, two experiments are carried out in this section and our proposed algorithm is implemented 10 runs in every experimental data. In the following experiments the RSMT was constructed using the method in [5], while the OSMT was constructed using OSMT_DPSO which was presented in this paper. In the first experiment, one of the test problems in OR-Library [17] was solved to compare the lengths of OSMT with the lengths of RSMT. In Table , the lengths different from our OSMT and RSMT [5] in 46 test data sets were given. From Table , we can observe that the Octagonal interconnect architecture demonstrates a average wire length reduction is 9.84%. And in [1], an exact algorithm for constructing OSMT was proposed and showing that OSMTs are consistently 10% smaller than RSMTs. Meanwhile, the test data sets are also obtained from OR-Library [17] and the reduction 9.75±2.29% is given in [1]. Our proposed the average wire

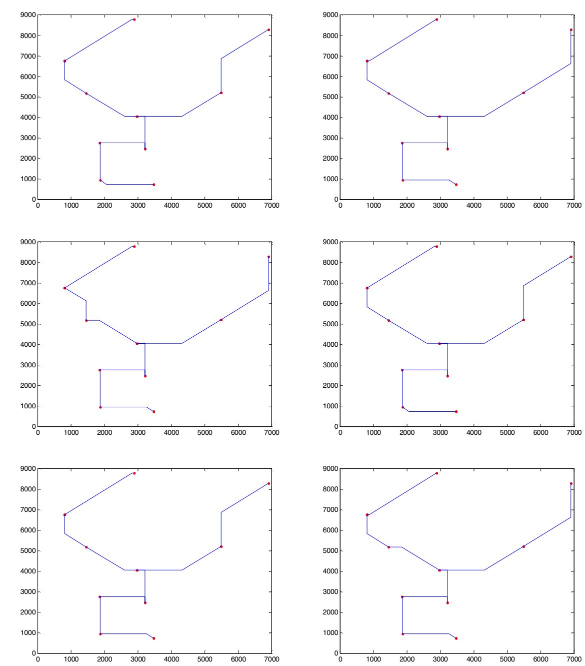

Fig. 3. The instance with 10 points which has six topologies in the same optimum length

length reduction 9.84% is within the scope of 9.75±2.29%.

TABLE III

THE LENGTHS DIFFERENT FROM OSMT AND RSMT IN 10 RANDOM INSTANCES USING *RAND_POINTS [18]*

Instance	Size	RSMT	OSMT	Reduction (%)
1	8	17931	17040	5.0
2	9	20503	18163	11.4
3	10	21910	19818	9.5
4	20	35723	32199	9.9
5	50	53383	47960	10.2
6	70	61987	55980	9.7
7	100	76016	68743	9.6
8	410	156520	142880	8.7
9	500	170273	154290	9.4
10	1000	245201	222050	9.4
Average	------	---------	----------	9.28

In the second experiment, we create 10 random instances using *rand_points* [18] and constructed the corresponding OSMT and RSMT. The size of each instance and results are listed in Table and here the OSMT_DPSO can achieve 9.28% average wire length reduction. The average wire length reduction in this experiment is also within the scope of the given value $9.75 \pm 2.29\%$ [1]. Furthermore, we also find the following observation in our experiment. In most of the instances, our proposed algorithm can obtain several different topologies of OMSTs with the same optimum or quasi optimum wire length. In the VLSI global routing stage, those various topologies are helpful for the purpose of congestion optimization. Here the instance with 10 points which has six topologies in the same optimum length is shown in Fig. 3.

As mentioned above, these two experiments have achieved similar wire length reduction to the exact algorithm for constructing OSMT in [1]. Moreover, our proposed algorithm could get various topologies of OSMTs which can facilitate congestion optimization.

V. CONCLUSIONS AND FUTURE WORK

This paper presents an efficient algorithm based on discrete PSO for the OSMT construction to optimize the wiring length and simultaneously obtain several topologies of OSMTs which is beneficial for the purpose of congestion optimization in VLSI global routing stage. The algorithm has been implemented in MATLBA 2009a. A number of test data sets in OR-Library and random instances have been tested and the results are very promising. However, there are some issues must be explored in the future. For example: (1) We can further construct OSMT with obstacles; (2) In the meantime, we can consider delay optimization which brought a lot of attention.

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.

[2] C. Chiang, "C.S. Chiang. Octilinear Steiner tree construction," in *Proc. 45th Midwest Symposium on Circuits and Systems*, 2002, pp. 603-606.

[3] Q. Zhu, H. Zhou, T. Jing, et al., "Spanning graph-based nonrectilinear Steiner tree algorithms," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 4, pp. 1066-1075, July 2005.

[4] T. Samanta, P. Ghosal, H. Rahaman, et al, "A heuristic method for constructing hexagonal Steiner minimal trees for routing in VLSI," in *2006 IEEE Int. Conf. Circuits and Systems*, pp. 1788-1791.

[5] 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.

[6] T. Arora, M.E. Mose, "Ant colony optimization for power efficient routing in Manhattan and non-Manhattan VLSI architectures," in *2009 Swarm Intelligence Symposium,* pp.137-144.

[7] R.C. Eberhar, J. Kennedy, "A new optimizer using particles swarm theory," in *Proc.6th Int. Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39-43.

[8] Y.L. Shen, Q.Z. Liu, W.Z. Guo, "Obstacle-avoiding rectilinear Steiner minimum tree construction based on discrete particle swarm optimization," *in Proc. 7th Int. Conf. Natural Computation*, 2011, pp. 2179 – 2183.

[9] G. Tyagi, M. Pandit, "Combined heat and power dispatch using Particle swarm optimization," in *2012 IEEE Students' Conf. Electrical, Electronics and Computer Science*, pp. 1-4.

[10] M.P. Padma, G. Komorasamy, "A modified algorithm for clustering based on particle swarm optimization and K-means," in *2012 Int. Conf. Computer Communication and Informatics,* pp. 1-5.

[11] M. Clerc. (2000, February 29). Discrete particle swarm optimization-illustrated by the traveling salesman problem. [Online]. Available: http://www.mauriceclerc.net

[12] G.L. Chen, W.Z. Guo, Y.Z. Chen, "A PSO-based intelligent decision algorithm for VLSI floorplanning," *J. Soft Computing*, vol. 14, no. 12, pp. 1329-1337, 2010.

[13] W.Z. Guo, G.L. Chen, "An efficient discrete particle swarm optimization algorithm for multi-criteria minimum spanning tree," *J. Pattern Recognition and Artificial Intelligence*, vol. 22, no. 4, pp. 597-604, August 2009. [*In Chinese with English abstract*].

[14] J. Kennedy, R.C. Eberhart, "A discrete binary bersion of the particle swarm optimization algorithm," in Proc. IEEE Int. Conf. Systems Man and Cybernetics, Orlando, USA, 1997, pp. 4104-4109.

[15] R.M Hare, B.A. Julstrom, "A spanning-tree-based genetic algorithm for some Instances of the rectilinear Steiner problem with obstacles," in *2003 ACM symposium on Applied computing*, pp. 725-729.

[16] Y.H. Shi, R.C. Eberhart, "A modified particle swarm optimizer," in IEEE Int. Conf. Evolutionary Computation, Piscataway, NJ, 1998, pp. 69-73.

[17] J.E. Beasley, "OR-Library: distributing test problems by electronic mail," *J. the Operational Research Society*, vol. 41, no. 11, pp. 1069-1072, 1990.

[18] M. Zachariasen. (2003, January 7). GeoSteiner Homepage [Online]. Available: http://www.diku.dk/geosteiner