점집합을 개체로 이용한 직각거리 스타이너 나무 문제의 하이브리드 진화 전략에 관한 연구

양 병 학*

A Nodes Set Based Hybrid Evolutionary Strategy on the Rectilinear Steiner Tree Problem

Byounghak Yang*

■ Abstract ■

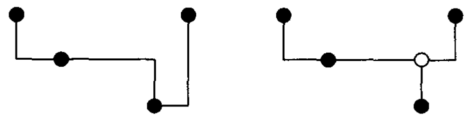
The rectilinear Steiner tree problem (RSTP) is to find a minimum-length rectilinear interconnection of a set of terminals in the plane. It is well known that the solution to this problem will be the minimal spanning tree (MST) on some set Steiner points. The RSTP is known to be NP-complete. The RSTP has received a lot of attention in the literature and heuristic and optimal algorithms have been proposed. A key performance measure of the algorithm for the RSTP is the reduction rate that is achieved by the difference between the objective value of the RSTP and that of the MST without Steiner points. A hybrid evolutionary strategy on RSTP based upon nodes set is presented. The computational results show that the hybrid evolutionary strategy is better than the previously proposed other heuristic. The average reduction rate of solutions from the evolutionary strategy is about 11.14%, which is almost similar to that of optimal solutions.

Keyword : Rectilinear Steiner Tree Problem, Evolutionary Strategy, Meta Heuristic

본 논문은 2005년도 한국경영과학회 추계학술대회(2005년 10월 29일) 우수논문상(응용부문) 수상논문으로 소정의 심사과정을 거쳐 게재 추천되었음.
* 경원대학교 산업정보시스템공학과

1. Introduction

Given a set V of n terminals in the plane, the rectilinear Steiner tree problem (RSTP) in the plane is to find a shortest network, a Steiner minimum tree, interconnecting S. The points in S are called Steiner points. We should find the optimal number of Steiner points and their location on rectilinear plane. It is well-known that the solution to RSTP will be the minimal spanning tree (MST) on some set of points $V \cup S$.



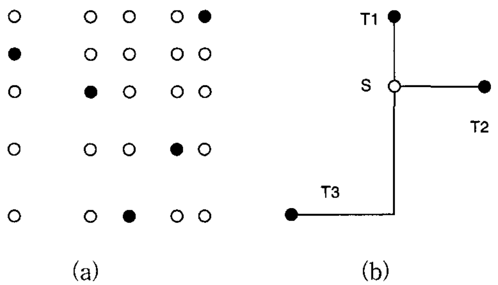
〈Figure 1〉 The minimal spanning tree and minimal Steiner tree

The RSTP is known to be NP-complete [8]. Polynomial-time algorithm for the optimal solution is unlikely to be known [7].

Let MST(V) be the cost of MST on set of terminals V. Then the reduction rate $R(V)$= {MST($V \cup S$) - MST(V) } / MST(V) is used as performance measure for algorithms on the Steiner tree problem. Warme, Winter and Zachariasen [19] presented an exact algorithm for RSTP known to be the best exact algorithm and showed that the average reduction rate was about 11.4% for the optimal solution. Beasley [3] presented standard instances for Steiner tree problem and a heuristic algorithm for the RSTP [4]. Many heuristics for RSTP start with an MST. Ho, Vijayan, and Wong [11] proposed for finding optimal L-shaped and Z-shaped embedding of rectangular layouts of a separable MST. Lee, Bose and Hwang [16] presented algorithm similar to the Prim algorithm for the MST. Hwang[12] improved their heuristic to make use of the MST to reduce the calculation time. The one of best heuristic of RSTP is Kahng and Robins' [15] the Batched Iterated 1-Steiener (BI1S) which starts with the set of nodes and iteratively adds a new Steiner point to the set such that the MST for the new set is minimized among all such sets with one extra node. Another best heuristic is Borah and Owens' [5] the edge-based heuristic which starts with a MST and repeatedly connects a node to the nearest point on the rectangular layout of an edge, removing the longest edge of the loop thus formed. The average reduction rates of the both heuritics are about 10.7%.

Some Genetic Algorithms (GA) for Euclidean Steiner tree problem were presented by Hesser et al. [10], Jones and Harris [13] and Barreiros [2]. Hesser [10] used an individual as an assembly of the (x, y) positions of a fixed number of Steiner points. Jones and Harris [13] introduced an individual which was an array of the (x, y) position of Steiner points and some extra flags used in Steiner point generation. They mentioned that the solution quality of their GA was not so good and the problem lay in how they generate Steiner points, not in how the GA operators work. Barreiros [2] assumed the number of Steiner point is equal to the number of the problem terminals. Two consecutive Steiner points are always considered to be connected and these points form a 'backbone' of the Steiner tree. Every terminal is always considered to be connected to the nearest Steiner points in backbone. He used this backbone as individual in GA for small sized problem. For solving the normal sized problem, he first divided terminals to small sub set to solve small sized problem and then connected small tree to original tree.

Wakabayashi [18] encoded rectilinear Steiner tree as permutations of vertex labels and '+' symbols and introduced a genetic algorithm for multi-objective RSTP. Julstrom [14] compared three kinds of genetic algorithm for RSTP which were the Prüfer numbers method, the string of weights and the list of edges method. In his research, the genetic algorithm based on lists of edges was relatively better than the other genetic algorithm. We will compare our result with Beasley's, Kahng's, Borah's and Julstrom's results. Main motivation of this research is to develop an hybrid evolutionary strategy to meet the reduction rate to 11% of optimal solutions.



<Figure 2> Hanan's grid graph and optimal Steiner point for 3 Terminals

Hanan showed that for any instance, an optimal rectilinear Steiner tree (RST) exists in which every Steiner point lies at the intersection of two orthogonal lines that contain terminals [9]. Hanan's theorem implies that a graph G called the Hanan grid graph is guaranteed to contain an optimal RST. Hanan's grid graph is constructed as follows: draw a horizontal and vertical line through each terminal. The vertices in graph correspond to the intersections of the lines. In <Figure 2>(a), the black dots are the terminals and white dots are the intersection of two lines. The optimal Steiner point should be white dots in Hanan's gird.

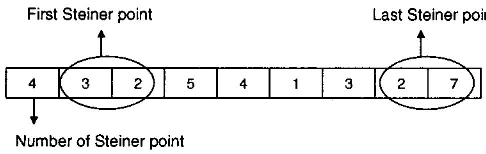
For 3 terminals RSTP, it is easy to find the optimal Steiner point. Let (x_i, y_i) be the coordinates of the given terminal T_i; France [6] proved that the Steiner point S is located at (x_m, y_m), where x_m and y_m are the medians of $\{x_i\}$and $\{y_i\}$, as like <Figure 2>(b).

2. Evolutionary Strategy

Evolutionary strategy is based on models of organic evolution. They model the collective learning process within a population of individuals. The starting population is initialized by randomization or some heuristic method, and evolves toward successively better regions of search space by means of randomized process of recombination, mutation and selection. Each individual is evaluated as the fitness value, and the selection process favors those individuals of higher quality to reproduce more often than worse individuals. The recombination mechanism allows for mixing of parental information while passing it to their dependents, and mutation introduces innovation into the population. Although simplistic from a biologist's viewpoint, these algorithms are sufficiently complex to provide robust and powerful adaptive search mechanisms [1]. In this research, some Steiner points in Hanan's grid may become an individual, and the individual is evaluated by minimum spanning tree with terminals and Steiner points in the individual.

2.1 Individual and evaluation

By the Hanan's theorem, the optimal Steiner points should be on the Hanan's grid. An individual is introduced as represented the location of Steiner point on Hanan's grid. Each vertical line and horizontal line is named as increasing number. Most left vertical line is the first vertical line, and most bottom horizontal line is the first horizontal line. Each individual has multiple (v_i, h_i) which is the location of the i-th Steiner point S_i on Hanan's grid, which v_i is the index of vertical line on Steiner point S_i and h_i is the index of horizontal line on Steiner point S_i. And (x_i, y_i) is the real location on plane for Steiner point S_i, and is calculated from (v_i, h_i). In this research, an individual is an assembly of the number of Steiner points and the locations of Steiner points as follows ; $\{m, (v_1, h_1), (v_2, h_2), \cdots, (v_m, h_m)\}$ where m is the number of Steiner points and vary on each individual.



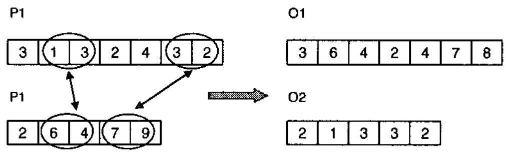
〈Figure 3〉 The individual for evolutionary strategy

Each individual has different length depending on its number of Steiner points, and represents one Steiner Tree. The fitness of the individual corresponds to the cost of the MST that can be constructed by using the original terminals and the Steiner points in individual by Prim algorithm.

2.2 Recombination

The recombination or crossover operator exchanges a part of an individual between two individual. We choose some Steiner points in each individual and exchange with other Steiner points in the other individual. We have two parents to crossover as P1={3, (1,3), (2,4), (3,2)} and P2{2, (6,4), (7,8)}. By the random number, we choose the number of exchanging which is less than the number of Steiner points in both parents. Then by the random number, some Steiner points are choose and exchanged.

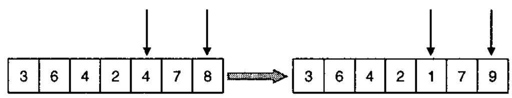
In <Figure 4>, random function chooses the 2 as exchanging number. First and third Steiner points are selected in P1, and two Steiner points are selected in P2. Therefore we have two children by performing crossover as follows : O1={3, (6,4), (2,4), (7,8)} and O2{2, (1,3), (3,2)}.



〈Figure 4〉 The crossover operator

2.3 Mutation

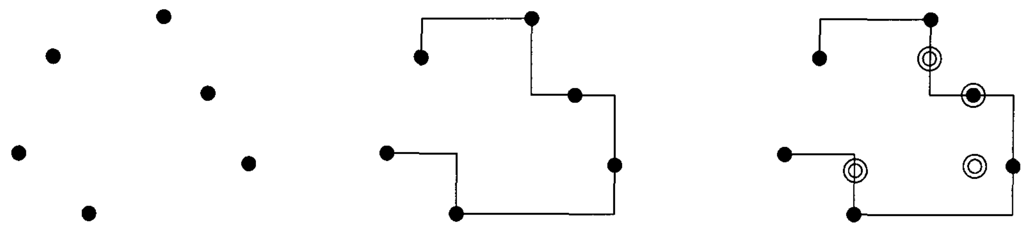
The mutation operator changes the locations of some selected Steiner points. Let (v_i, h_i) be the selected location of Steiner point. Then we can have new location $(v_i, +v, h_i + h)$ where v and h are some random integer number.



〈Figure 5〉 The mutation Operator

2.4 Selection

We use the tournament selection. The k-tournament selection method selects a single individual by choosing some number k of in



(a) Terminals (b) Minimum Spanning tree (c) Steiner points Pool

<Figure 6> The Steiner points pool from a minimum spanning tree

dividuals randomly from the population and selecting the best individual from this group to survive to the next generation. The process is repeated as often as necessary to fill the new population. A common tournament size is k=2.

2.5 Initial population

For the 3-terminal RSTP, the median of terminals is the optimal Steiner point. We assume that a median point of some adjacent 3-terminal has high probability to be survived in the optimal Steiner tree. For the convenience to find adjacent 3-terminal, we make a minimum spanning tree for the V and find every directly connected 3 terminals in the minimum spanning tree as like <Figure 6>(b). And a Steiner point for these connected 3 terminals is calculated. We make the Steiner point for every connected 3 terminals in minimum spanning tree, and put the Steiner point in the Steiner points pool as like <Figure 6>(c) and use one of candidate Steiner point. We make an initial population for the evolutionary strategy as following procedure :

Step 1 : Make a minimum spanning tree (MST) for the terminals.

Step 2 : Choose every connected three terminals in MST. Let them be 3-neighbor terminals (3NT).

Step 3 : Make a Steiner point for each 3NT and add it in the Steiner points pool.

Step 4 : Choose some Steiner points from Steiner points pool by random function and make one individual. Repeat Step 4 until all individuals are decided.

We make an initial population with 200 individuals.

2.6 Hybrid operator

For searching new solution in evolutionary strategy, the local search can be used. In this research, an insertion operator, a deletion operator and a moving operator are introduced. Mutation operator and crossover have the location of Steiner point moved to search a new solution, but they don't search new Steiner point. We need to introduce a new Steiner point in current individual to enforce the variability of the solution. For some selected individual, randomly generated Steiner point in Hanan's grid is inserted in individual and increase the number of Steiner point as an insertion operator. On the other hand, some Steiner points in tree are connected with only one or two other node. Those kinds of Steiner point in tree are obviously useless to reduce the tree cost. We introduce deletion operator to delete poor Steiner points which are connected less than 2 other node in Steiner tree. For the Steiner point connected with 3 other node, we know the optimal location of Steiner point as using the results of 3-terminal case. We trace the Steiner point (S) which is connected with exactly 3 other node(A,B,C) and calculate the optimal location(S*) of Steiner point for those 3 node(A,B,C) and move the location of S to S* as a moving operator. The proposed hybrid evolutionary algorithm is shown in <Figure 7>.

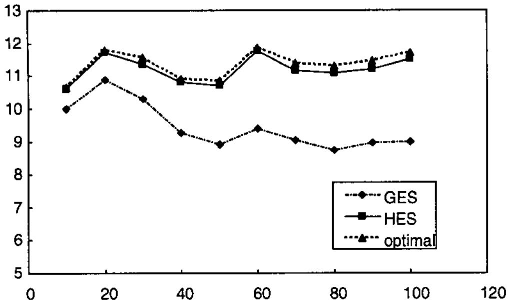
```
HES (Hybrid Evolutionary Strategy)

begin
  t = 0
  Initialize population P(t)
  Fitness evaluation P(t)
while (do not satisfy termination criteria) do
  begin
    t = t +1
    Selection P(t) from P(t-1)
    Crossover on P(t)
    Mutation on P(t)
    Local search on P(t)
    Fitness evaluation P(t)
    end
end
```

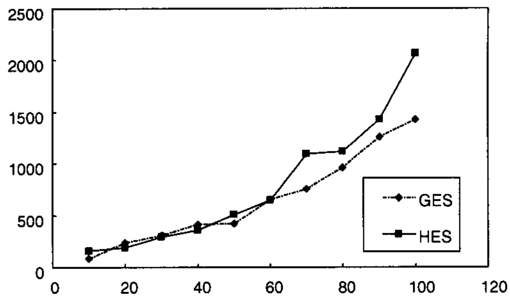
〈Figure 7〉 The procedure of hybrid evolutionary strategy

3. Computational Experience

The computational study was made on Pentium IV processor. The evolutionary strategy was programmed in Visual Studio. Problem instances are from OR-Library [3], 15 instances for each problem size 10, 20, ⋯, 100, 250, 500. We introduced a general evolutionary strategy (GES) without the hybrid operator in 2.6 and a hybrid evolutionary strategy (HES) with the hybrid operator. First computational experiment is to compare the reduction rate and the calculation time of the GES and those of HES for the 10 to 100 terminal instances. The results are shown in <Figure 8> and <Figure 9>. For the reduction rate, HES is very close to the optimal solution and GES is worse than the other heuristics. In <Figure 9>, for the calculation time, GES is a little bit better than HES. Even the calculation time of HES is longer than GES, the solution quality of HES is quite good to accept HES. We need to further research to reduce the calculation time of HES. Next tests focused only on the performance of HES. By our parameter setting, the crossover rate was 0.09, the mutation rate was 0.01 and the hybrid operation rate was 0.31. The population size was 200. Each test problem was solved by our HES. The results are shown in <Table 1>. In <Table 1>, for each value of n, the minimum, the average, the maximum for the reduction rate and the number of Steiner points. For the RSTP, Beasley has solved same instances as our ones. The optimal solutions of same instances are shown by Warme [20]. The result of Borah's and Kahng's heuristics and Julstrom's genetic algorithm are shown their papers. Those results are in <Table 2>. In <Table 2>, HES gives larger percentage reduction (about 11.14%) in most instances than any other heuristic. The 46 test problems for optimal solution were given by Soukup and Chow [17]. Warme [20] solved all of this problem and showed the optimal solutions. The 46 test problem were solved by HES to compare with the optimal solutions. <Table 3> shows the Steiner tree cost of optimal solutions and those of HES. In <Table 3>, the HES found the optimal solutions in 44 instances among 46 instances. Finally, we show



〈Figure 8〉 The reduction rate of HES and GES



〈Figure 9〉 The calculation time of HES and GES

〈Table 1〉 Summary of the Computational results by HES

n	Reduction Rate			Number of Steiner points		
	min	average	max	min	average	max
10	4.377	10.611	19.299	2	3.933	6
20	7.462	11.721	14.734	7	9.067	12
30	8.496	11.356	16.542	12	13.867	16
40	8.367	10.807	13.884	15	18.533	22
50	8.628	10.705	12.457	19	22.933	27
60	9.292	11.755	13.940	24	27.933	31
70	9.733	11.163	12.775	29	32.133	34
80	8.957	11.081	13.212	34	37.467	41
90	9.453	11.203	13.000	37	40.600	45
100	9.777	11.501	13.478	41	45.400	51
250	10.016	11.090	11.845	102	111.133	117
500	10.231	10.811	12.054	209	217.000	230

〈Table 2〉 The reduction rate of the optimal solution, Hybrid evolutionary Strategy (HES), Beasley's, Kahng's, Borah's and Julstrom's heuristic (*: They didn't show the results)

n	HES	Beasley's	Kahng's	Borah's	Julstrom's	Optimal solution
10	10.611	9.947	10.36	10.33	*	10.656
20	11.721	10.590	10.44	10.4	*	11.798
30	11.356	10.250	*	*	*	11.552
40	10.807	9.956	*	*	*	10.913
50	10.705	9.522	10.71	10.71	9.167	10.867
60	11.755	10.146	*	*	*	11.862
70	11.163	9.779	*	*	2.8	11.387
80	11.081	9.831	*	*	*	11.301
90	11.203	10.128	*	*	4.06	11.457
100	11.501	10.139	10.89	10.84	0.72	11.720
250	11.090	9.964	10.88	10.88	*	11.646
500	10.811	9.879	*	10.94	*	11.631
mean	11.143	10.011	10.656	10.683	4.187	11.399

〈Table 3〉 The comparison for tree cost in the Soukup and Chow problems(*: The HES failed to find the optimal solution.)

Instances	n	HES	Optimal solution
1	5	1.87	1.87
2	6	1.64	1.64
3	7	2.36	2.36
4	8	2.54	2.54
5	6	2.26	2.26
6	12	2.42	2.42
7	12	2.48	2.48
8	12	2.36	2.36
9	7	1.64	1.64
10	6	1.77	1.77
11	6	1.44	1.44
12	9	1.8	1.8
13	9	1.5	1.5
14	12	2.6	2.6
15	14	1.48	1.48
16	3	1.6	1.6
17	10	2	2
18	62	4.04	4.04
19	14	1.88	1.88
20	3	1.12	1.12
21	5	1.92	1.92
22	4	0.63	0.63
23	4	0.65	0.65
24	4	0.3	0.3
25	3	0.23	0.23
26	3	0.15	0.15
27	4	1.33	1.33
28	4	0.24	0.24
29	3	2	2
30	12	1.1	1.1
31	14	2.59	2.59
32	19	3.13	3.13
33	18	2.68	2.68
34	19	2.43	2.41
35	18	1.51	1.51
36	4	0.9	0.9
37	8	0.9	0.9
38	14	1.66	1.66
39	14	1.66	1.66
40	10	1.55	1.55
41	20	2.24	2.24
42	15	1.53	1.53
43	16	2.57*	2.55
44	17	2.54*	2.52
45	19	2.2	2.2
46	16	1.5	1.5

〈Table 4〉 Results for OR-library problems 10~40

n	STC	MSTC	Red	OPT	%gap
10	2.301969	2.501884	7.990575	2.292075	0.431683
10	1.913410	2.072885	7.693342	1.913410	0.000000
10	2.600368	2.908126	10.582701	2.600368	0.000000
10	2.050677	2.210872	7.245784	2.046112	0.223121
10	1.881892	2.085090	9.745302	1.881892	0.000000
10	2.654077	3.130105	15.208053	2.654077	0.000000
10	2.602507	2.845864	8.551251	2.602507	0.000000
10	2.505621	2.824229	11.281221	2.505621	0.000000
10	2.207882	2.308941	4.376868	2.206236	0.074625
10	2.393610	2.635232	9.168931	2.393610	0.000000
10	2.223954	2.691523	17.371921	2.223954	0.000000
10	1.962632	2.144338	8.473750	1.962632	0.000000
10	1.948391	2.257994	13.711397	1.948391	0.000000
10	2.185613	2.708285	19.299010	2.185613	0.000000
10	1.864192	2.036604	8.465651	1.864192	0.000000
20	3.378281	3.844161	12.119174	3.370389	0.234163
20	3.263949	3.676536	11.222166	3.263949	0.000000
20	2.784742	3.074551	9.426057	2.784742	0.000000
20	2.762439	3.236090	14.636512	2.762439	0.000000
20	3.404032	3.992245	14.733880	3.403316	0.021041
20	3.601424	4.140105	13.011292	3.601424	0.000000
20	3.493487	3.897788	10.372574	3.493487	0.000000
20	3.801635	4.333770	12.278808	3.801635	0.000000
20	3.673994	4.223400	13.008625	3.673994	0.000000
20	3.402474	3.913073	13.048547	3.402474	0.000000
20	2.712681	2.931421	7.461901	2.712391	0.010692
20	3.045140	3.526389	13.647082	3.045140	0.000000
20	3.452218	3.809804	9.385927	3.443867	0.242489
20	3.406237	3.850494	11.537657	3.406237	0.000000
20	3.255649	3.614454	9.926951	3.230375	0.782405
30	4.102876	4.586079	10.536310	4.069299	0.825110
30	4.090006	4.565851	10.421829	4.090006	0.000000
30	4.318269	4.815896	10.333002	4.312044	0.144351
30	4.216786	4.802889	12.203140	4.215096	0.040094
30	4.173975	4.652218	10.279887	4.173975	0.000000
30	3.998215	4.540988	11.952758	3.995514	0.067596
30	4.433955	4.960538	10.615439	4.376139	1.321171
30	4.172371	4.559787	8.496356	4.169122	0.077940
30	3.713366	4.153980	10.607037	3.713366	0.000000
30	4.268661	4.900239	12.888711	4.268661	0.000000
30	4.185465	4.625424	9.511755	4.164799	0.496194
30	3.841672	4.309165	10.848811	3.841672	0.000000
30	3.740665	4.284652	12.696192	3.740665	0.000000
30	4.294111	5.145219	16.541723	4.289703	0.102779
30	4.314373	4.925398	12.405610	4.303558	0.251304
40	4.517278	4.988419	9.444698	4.484152	0.738731
40	4.681131	5.278928	11.324214	4.681131	0.000000
40	4.997416	5.628845	11.217737	4.997416	0.000000
40	4.528986	4.942542	8.367273	4.528986	0.000000
40	5.194466	5.785622	10.217663	5.194041	0.008184
40	4.976504	5.618419	11.425179	4.975339	0.023432
40	4.568207	5.119388	10.766540	4.563901	0.094343
40	4.874600	5.349185	8.872100	4.874600	0.000000
40	5.176179	6.010718	13.884180	5.176179	0.000000
40	5.713685	6.438572	11.258508	5.713685	0.000000
40	4.692986	5.233565	10.329065	4.673421	0.418642
40	4.384338	4.809879	8.847225	4.384338	0.000000
40	5.214076	5.850367	10.876088	5.188455	0.493818
40	4.916695	5.580131	11.889248	4.916695	0.000000
40	5.082807	5.868208	13.384004	5.082807	0.000000

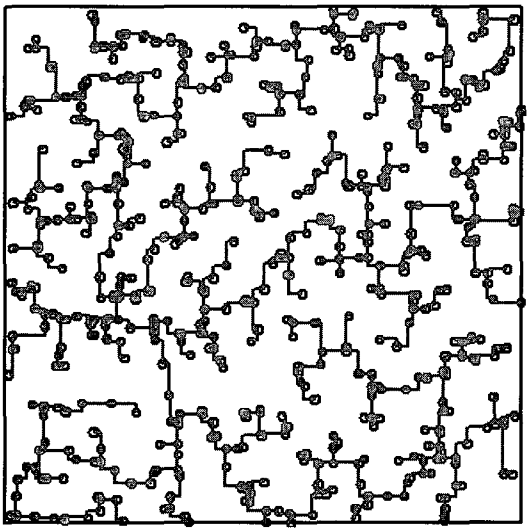
〈Table 5〉 Results for OR-library problems 50-80

n	STC	MSTC	Red	OPT	%gap
50	5.508544	6.292395	12.457112	5.494866	0.248929
50	5.548425	6.212306	10.686561	5.548425	0.000000
50	5.476689	6.221504	11.971612	5.469104	0.138703
50	5.161548	5.800274	11.011987	5.153577	0.154685
50	5.518602	6.039675	8.627501	5.518602	0.000000
50	5.593741	6.322031	11.519873	5.580429	0.238553
50	5.000242	5.692668	12.163472	4.996118	0.082538
50	5.375471	5.976837	10.061619	5.375471	0.000000
50	5.381767	5.993608	10.208221	5.345677	0.675116
50	5.421111	6.012603	9.837544	5.403796	0.320412
50	5.253292	5.910225	11.115191	5.253292	0.000000
50	5.363311	5.887089	8.897069	5.340929	0.419062
50	5.392747	6.016392	10.365761	5.389102	0.067642
50	5.370040	6.011994	10.677884	5.355142	0.278204
50	5.223717	5.867421	10.970823	5.218086	0.107907
60	5.384679	6.140916	12.314733	5.376142	0.158781
60	5.536780	6.226545	11.077811	5.536780	0.000000
60	5.667230	6.472112	12.436158	5.656680	0.186509
60	5.537723	6.289463	11.952373	5.537104	0.011179
60	5.470499	6.203574	11.816977	5.470499	0.000000
60	6.049538	6.784063	10.827204	6.042196	0.121515
60	5.926492	6.800563	12.852917	5.897804	0.486415
60	5.813818	6.620793	12.188503	5.813818	0.000000
60	5.597639	6.328406	11.547410	5.587711	0.177676
60	5.769777	6.391773	9.731191	5.762449	0.127173
60	5.631777	6.208662	9.291612	5.614167	0.313683
60	5.993593	6.964434	13.939994	5.979136	0.241779
60	6.121353	6.872804	10.933686	6.121353	0.000000
60	5.603553	6.417035	12.676917	5.603553	0.000000
60	5.662258	6.488738	12.737147	5.662258	0.000000
70	6.216743	6.986094	11.012599	6.205886	0.174945
70	6.105044	6.875814	11.209876	6.092849	0.200148
70	6.205756	6.990516	11.226071	6.193466	0.198424
70	6.306707	6.991941	9.800340	6.293858	0.204148
70	6.225699	6.913773	9.952213	6.225699	0.000000
70	6.212561	7.122416	12.774533	6.212453	0.001737
70	6.254855	7.043327	11.194592	6.222367	0.522120
70	6.198887	7.020859	11.707573	6.187285	0.187515
70	6.339620	7.077673	10.427903	6.298613	0.651040
70	6.255041	7.025816	10.970614	6.251183	0.061718
70	6.645576	7.516908	11.591630	6.645576	0.000000
70	6.319954	7.181375	11.995205	6.304713	0.241743
70	6.353336	7.038404	9.733296	6.291226	0.987246
70	6.043727	6.828391	11.491198	6.041112	0.043282
70	6.250955	7.132336	12.357542	6.231846	0.306630
80	7.104259	8.002857	11.228458	7.092744	0.162348
80	6.532461	7.434127	12.128740	6.527381	0.077824
80	6.543591	7.432810	11.963429	6.533255	0.158212
80	6.440664	7.176152	10.249065	6.419345	0.332109
80	6.636066	7.585674	12.518437	6.635053	0.015267
80	7.112493	7.871256	9.639677	7.100744	0.165452
80	6.839466	7.880621	13.211591	6.822848	0.243567
80	6.820746	7.690611	11.310744	6.745238	1.119425
80	6.983243	8.040083	13.144638	6.982565	0.009707
80	6.559871	7.290314	10.019368	6.549799	0.153774
80	6.652729	7.307250	8.957145	6.628310	0.368405
80	6.517687	7.171462	9.116339	6.507009	0.164108
80	6.815223	7.656879	10.992144	6.802265	0.190506
80	7.041204	7.904100	10.917071	7.007790	0.476809
80	6.999139	7.848814	10.825522	6.993907	0.074804

〈Table 6〉 Results for OR-library problems 90-500

n	STC	MSTC	Red	OPT	%gap
90	6.842857	7.717188	11.329668	6.835036	0.114422
90	7.129485	8.006525	10.954075	7.129485	0.000000
90	7.520222	8.395115	10.421461	7.481747	0.514242
90	7.128021	7.872189	9.453135	7.091006	0.521989
90	7.192045	8.171431	11.985483	7.183122	0.124218
90	6.870718	7.760545	11.466032	6.864035	0.097371
90	7.212608	8.120775	11.183251	7.203689	0.123819
90	7.300638	8.197037	10.935638	7.234167	0.918855
90	6.785945	7.799959	13.000248	6.785601	0.005071
90	7.247792	8.279729	12.463420	7.231041	0.231650
90	7.232231	8.124839	10.986163	7.231004	0.016967
90	6.964982	7.931811	12.189265	6.936726	0.407342
90	7.320608	8.125796	9.909033	7.281066	0.543078
90	6.952070	7.749269	10.287415	6.918899	0.479419
90	7.191411	8.124766	11.487780	7.177829	0.189215
100	7.289523	8.251678	11.660119	7.252217	0.514414
100	7.549567	8.586721	12.078585	7.517663	0.424382
100	7.274682	8.232048	11.629745	7.274601	0.001113
100	7.442618	8.334606	10.702228	7.434239	0.112702
100	7.576702	8.427539	10.095911	7.567020	0.127953
100	7.474154	8.542745	12.508752	7.441499	0.438823
100	7.782745	8.903611	12.588894	7.774058	0.111742
100	7.315364	8.231337	11.127880	7.303318	0.164936
100	7.802998	9.018468	13.477568	7.795203	0.099995
100	7.603212	8.572570	11.307670	7.595220	0.105216
100	7.884077	8.738441	9.777071	7.867486	0.210887
100	7.626623	8.699562	12.333253	7.613110	0.177494
100	7.496535	8.503628	11.843095	7.460499	0.483028
100	7.888076	8.798508	10.347570	7.863280	0.315347
100	7.076385	7.953732	11.030638	7.044649	0.450489
250	11.759209	13.082227	10.113098	11.660981	0.842359
250	11.576572	13.034454	11.184830	11.515008	0.534646
250	11.519080	12.915711	10.813422	11.465040	0.471349
250	11.871394	13.290703	10.678958	11.781953	0.759136
250	11.770383	13.287220	11.415757	11.692709	0.664299
250	11.649603	13.000556	10.391499	11.625625	0.206250
250	11.603117	12.894627	10.015877	11.527735	0.653919
250	11.798068	13.383269	11.844648	11.683332	0.982047
250	11.737671	13.251040	11.420762	11.682199	0.474840
250	11.787857	13.356546	11.744722	11.685763	0.873662
250	11.357994	12.833252	11.495593	11.288961	0.611502
250	11.952655	13.436222	11.041549	11.903526	0.412728
250	11.670725	13.200288	11.587350	11.604950	0.566783
250	11.686791	13.165300	11.230345	11.618879	0.584498
250	11.649770	13.144921	11.374367	11.555820	0.813009
500	16.469137	18.419000	10.586151	16.297881	1.050784
500	16.213260	18.435407	12.053688	16.075685	0.855796
500	16.445173	18.568322	11.434250	16.266466	1.098623
500	16.546545	18.494398	10.532125	16.411100	0.825326
500	16.172066	18.060572	10.456512	16.058616	0.706473
500	16.638710	18.648836	10.778828	16.468507	1.033503
500	16.176323	18.142319	10.836520	16.012423	1.023578
500	16.283130	18.221092	10.635817	16.124814	0.981817
500	16.310447	18.245223	10.604291	16.210044	0.619389
500	15.684610	17.586729	10.815646	15.558120	0.813015
500	16.301537	18.302882	10.934590	16.167432	0.829479
500	16.582771	18.472623	10.230557	16.400959	1.108541
500	16.278519	18.261219	10.857437	16.132420	0.905622
500	16.795443	18.865160	10.971108	16.598433	1.186920
500	16.217036	18.107129	10.438394	16.075847	0.878267

the computational results of each OR-library instance solved by HES. In <Table 4> through <Table 6>, STC is the Steiner tree cost by HES, MSTC is the minimal spanning tree cost, Red is the reduction value, OPT is the optimal Steiner tree cost from Warme [20] and %gap is the percentage of gap between the HES solution and optimal solution as 100 (STC-OPT) / OPT. Zero %gap means that HES finds the optimal solution. In small size problem, the rate of finding the optimal solution is about 50%. The rate of finding the optimal solution decreases while the problem size increases. The worst case of %gap is 1.32%, therefore our solution is in 98.5% of the optimal solution. In <Figure 10>, the rectilinear Steiner tree for 500 terminals is shown.



<Figure 10> The rectilinear Steiner tree for 500 terminals by HES

4. Conclusion

In this paper we have introduced an hybrid evolutionary strategy for rectilinear Steiner tree problem. There were a couple of genetic approaches for ESTP or RSTP. They focused on the representation of Steiner tree in genetic algorithm. On the other hand, we focused on how we generate or search better Steiner points. For generating new Steiner point and improving the solution quality in evolutionary strategy, we introduced the hybrid operator which consisted of the insertion, the deletion and the moving Steiner points. Computational results show that the hybrid evolutionary strategy gives better Steiner tree solution than the other heuristics. As comparing with optimal solution, our hybrid evolutionary strategy found 44 optimal solutions among 46 instances. And the worst gab between the values of optimal solution and that of evolutionary strategy is less than 1.5%. Our evolutionary strategy should solve the minimum spanning tree problems for the all individual at every iteration, therefore the calculation time of evolutionary strategy is much bigger than that of the non-genetic heuristic algorithm. We are trying to modify our methods to reduce the calculation time to adapt our algorithm to RSTP with obstacle which is common in real field.

References

[1] Back, Thomas, *Evolutionary Algorithm in Theory and Practice*, Oxford University Press, 1996.

[2] Barreiros, J., "An Hierarchic Genetic Algorithm for Computing (near) Optimal Euclidean Stein Steiner Trees," *Workshop on Application of hybrid Evolutionary Algorithms to NP-Complete Problems*, Chicago, 2003.

[3] Beasley, J.E., "OR-Library : Distributing Test Problems by Electronic Mail," *Journal*

of the Operational Research Society, Vol.41 (1990), pp.1069-1072.

[4] Beasley, J.E., "A Heuristic for Euclidean and Rectilinear Steiner Problems," *European Journal of Operational Research,* Vol.58 (1992), pp.284-292.

[5] Borah, M. and R.M. Owens, "An Edge-Based Heuristic for Steiner Routing," *IEEE Trans. on Computer Aided Design,* Vol. 13(1994), pp.1563-1568.

[6] France, R.L., "A Note on the Optimum Location of New Machines in Existing Plant Layouts," *J. Industrial Engineering,* Vol.14 (1963), pp.57-59.

[7] Ganley, J.L., "Computing Optimal Rectilinear Steiner Trees : A Survey and Experimental Evaluation," *Discrete Applied Mathematics,* Vol.90(1999), pp.161-171.

[8] Garey, M.R. and D.S. Johnson, "The rectilinear Steiner Tree Problem is NP-complete," *SIAM Journal on Applied Mathematics,* Vol.32(1977) pp.826-834.

[9] Hanan, M., "On Steiner's Problem with Rectilinear Distance," *SLAM Journal on Applied Mathematics*, Vol.14(1966), pp.255-265.

[10] Hesser, J., R. Manner, O. Stucky, "Optimization of Steiner Trees using Genetic Algorithms," *Proceedings of the Third International Conference on Genetic Algorithm*, (1989), pp.231-236.

[11] Ho, J.M., G. Vijayan, and C.K. Wong, "New Algorithm for the Rectilinear Steiner Tree Problem," *IEEE Trans. on Computer Aided Design*, Vol.9(1990), pp.185-193.

[12] Hwang, F.K., "An O(n logn) Algorithm for Suboptimal Rectilinear Steiner Trees," *IEEE Transaction son Circuits and Systems*, Vol.26(1979), pp.75-77.

[13] Joseph, J. and F.C. Harris, "A Genetic Algorithm for the Steiner Minimal Tree Problem," *Procedings of Inernational Conference on Intelligent Systems*, 1996.

[14] Julstrom, B.A., "Encoding Rectilinear Trees as Lists of Edges," *Proceedings of the 16th ACM Symposium on Applied Computing*, (2001), pp.356-360.

[15] Kahng, A.B. and B. Robins, "A New Class of Iterative Steiner Tree Heuristics with Good Performance," *IEEE Trans. on Computer Aided Design*, Vol.11(1992), pp. 893-902.

[16] Lee, J.L., N.K. Bose, and F.K. Hwang, "Use of Steiner's Problem in Suboptimal Routing in Rectilinear Metric," *IEEE Transaction son Circuits and Systems*, Vol.23(1976), pp.470-476.

[17] Soukup, J. and W.F. Chow, "Set of Test Problems for the Minimum Length Connection Networks," *ACM/SIGMAP Newsletter*, Vol.15(1973), pp.48-51.

[18] Wakabayashi, S., "A Genetic Algorithm for Generating a Set of Rectilienar Steiner Trees in VLSI Interconnection Layout," *Information processing Society of Japan Journal*, Vol.43, No.5, 2002.

[19] Warme, D.M., 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 Tree*, Kluser Academic Publishers (1998).

[20] Warme, D.M., http://www.group-w-inc.com /~warme/research