Global Shortest Path Programming using Genetic Algorithms

Kun YE, Zong-Xiao YANG[*], Lei SONG, Li-Li XU

Institute of Systems Science and Engineering, Henan University of Science and Technology, Luoyang 471003, China

Corresponding author, Z-X Yang, Email: zxyang@mail.haust.edu.cn

***Abstract* — The global shortest path programming (GSPP) has extensive applications in engineering practices. The Steiner tree problem is a nonlinear programming conundrum with fixed points and fictive points and is typical theoretical basis of GSPP. The Steiner minimum tree (SMT) problem can be changed to a combination-optimization problem, a test selection algorithm for the construction of the initial population is proposed correspondingly, and an improved genetic algorithms (GA) is discussed to solve the objective of SMT problem. The simulation shows that the global optimum can be quickly obtained by the improved algorithm. Compared with the visualization experiment approach, the proposed approach can be fulfilled accurately and rapidly and it provides a convenient way and tool for the solution to the practical application problems in engineering fields.**

Keywords* — *Shortest path programming; Steiner minimum tree(SMT); Combinatorial optimization; Genetic algorithm(GA).

I. INTRODUCTION

GSPP problem has got more and more focuses in science and engineering fields. As a theoretical basis of shortest path, experts in different fields devote themselves to solving the SMT problem by researching various heuristic algorithms, and try to find its practical application in various fields [1]. But SMT problem has been proved to be a NP-hard problem, so reasonable solution is looking for heuristic algorithm [2][3].

GA is a kind of random search algorithm which takes example by biological natural selection and natural Genetic mechanism, proposed by Professor J. H. Holland in 1975[4]. It is simple, universal, robust, and suitable for parallel processing. GA is one of the most effective ways for the problems which were difficult to solve formerly, such as the function optimization problems, complex multi-objective programming, industrial and agricultural production in the piping, wiring problems, and machine learning, image recognition, weight coefficient adjustment and network structure of artificial neural network [5] [6].

In this paper, the SMT problem is transformed into a combinatorial optimization problem to search fictive Steiner points, and then solve the problem by improved GA, search out the optimal fictive Steiner points to construct SMT.

II. SMT PROBLEM

Given the fixed point set P (also called regular point set), SMT is the shortest tree which Join-point set P all points in Euclidean plane [7]. Because of allowing add auxiliary point set S (fictive Steiner point set), therefore, SMT is the problem that seeking point set S to make connection PUS spanning tree minimized [8][9]. There are many theories about SMT, but some basic properties needed are listed here only:

- Property 1 In SMT the angle of any two adjacent edges not less than 120 °.
- Property 2 In SMT adjacency edges of any vertex not more than three edges.
- Property 3 In SMT adjacency edges of fictive Steiner points will surely for three, and the included angle of any two edges is 120 ° in the three edges.
- Property 4 Hypothesis the number of fixed points is n in SMT, then the number of fictive Steiner points less than or equal n-2.
- Property 5 Hypothesis the area surrounded by fixed n-points is convex hull, then all fictive Steiner points must be included in the convex hull.
- Property 6 Each leaf is fixed point in SMT.

For SMT problem, if the number of fictive Steiner points and their position can be found out, the conventional minimum spanning tree algorithm can be used to solve it. Therefore, the key to the problem is to find fictive Steiner points.

III. GA SOLVING SMT

Genetic algorithm is used here to solve SMT problem. First for the following treatments:

1) A rectangular region of $length\_x \times width\_y$ that contains all the fixed points determined by coordinates of the fixed points set ($length\_x$, $width\_y$ respectively represent length and width);
2) According to specific problems accuracy requirement, and put the above areas into $n\_x = length\_x \times m{+}1$, $n\_y = width\_y \times m{+}1$ (m is grid subdivision multiples, its value can select any positive integer);
3) Fixed points and fictive Steiner points put on the grid vertex [10].

So, the rectangular area is discrete to $n\_x \times n\_y$ points. Through property 5 we know, fictive Steiner points will be selected in these discretization points. Then the SMT problem is transformed into a combinatorial optimization problem, which determines the optimal combination in all known possible fictive Steiner points, and makes it constitute Steiner minimum tree with the fixed points. So we can use

978-1-61284-840-2/11/$26.00 ©2011 IEEE

GA to solve the problem. Figure 1 shows the main program flowchart of the GA.

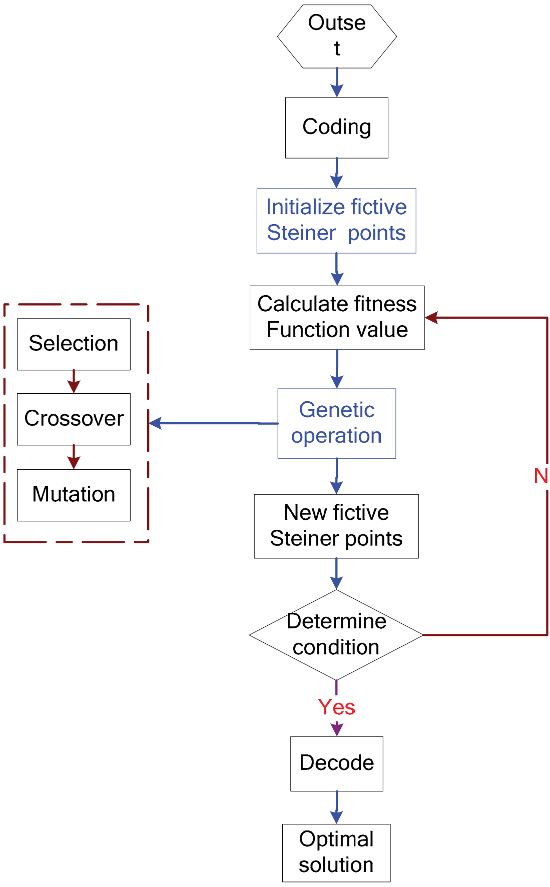

Figure 1. GA main program flowchart.

A. *Coding*

Binary coded strings is used here [11]. Since a point coordinates is composed of two values, so the coordinates value of the discrete points above is represented by two such long binary coding string, then the two coding strings constitute a new binary code strings, which is representative of a chromosome individual, thus chromosome coding method of the problem is obtained. When this coding method is used, the solution space and the searching space of genetic algorithm have one-to-one relationship.

B. *Fictive Steiner Point Initialization*

A tempted selection algorithm is used here to select fictive Steiner points (initial population). The main idea of algorithm: each iteration by adding a random point to reduce the costs of minimum spanning tree, until you have added *n-2* points, or join any of the remaining points are not likely to decrease so far. The algorithm main program flowchart is described in Figure 2, and the steps are as follow.

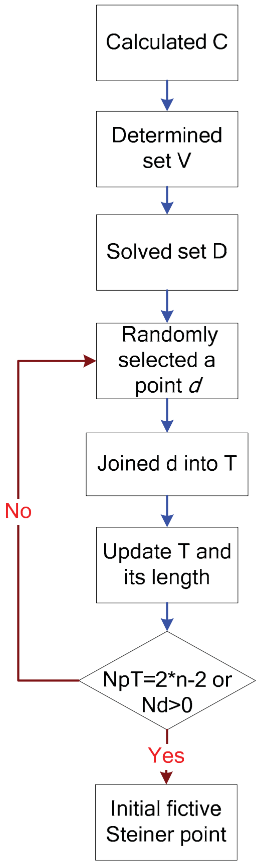

Figure 2. Selection algorithm main program flowchart.

1) Seek the minimum spanning tree *T* with a given fixed *n*-points , record its line length for *C*;
2) For possible fictive Steiner point set *V*, calculate respectively reduced costs that each candidate point as fictive Steiner point to join *T* later, denote by *D*;
3) Randomly selected a candidate point $d>0$ ($d \in D$), add it to the tree *T*, update the tree and its line length; also remove it from the points set *V* ;
4) Repeat 2) and 3) until the number of fictive Steiner points are *n-2* or all the d are less than or equal to zero (adding any remaining candidate points cannot reduce the line length).

Figure 2 includes: (1) *T* is the minimum spanning tree; (2) *C* is the length of *T*; (3) *V* is the set of possible fictive Steiner points; (4) *D* is the set of reduced costs that each candidate point as fictive Steiner point to join *T* later; (5) $d \in D$, $d>0$; (6) the number points of *T* (N_PT); (7) none of d (Nd).

C. *Calculate FitnessFunction Value*

Fitness function is the basis for the operation of genetic evolution, and its structure is the key to the genetic algorithm. Formula (1) is used to solve fitness function.

$$f(x)=\begin{cases} 0, & kru(x)\geq kru(v_0) \\ kru(v_0)-kru(x), & kru(x)<kru(v_0) \end{cases} \qquad (1)$$

Where, $x=[x\ (1), x\ (2) \ldots x\ (n\text{-}2)]$, $kru(x)$ means minimum spanning tree line length by Kruskal algorithm which is determined by fictive Steiner points $x=[x\ (1), x\ (2) \ldots x\ (n\text{-}2)]$ and fixed n-points set v_0. $kru(v_0)$ is the minimum spanning tree line length of set v_0.

The basic idea of Kruskal algorithm: Suppose given n-points, V is the set of given n - points, E is the set of edges by n -points. Then the process of construct minimum spanning tree according to Kruskal algorithm is: Firstly construct a sub-graph that only contains n-vertexes and void edge sets. If all vertices of the sub-graph serve as the root of the tree, it is a tree in a forest with n. Then select a minimum weight edge from the edge set E, if two vertices of the edge belong to different trees, then join it to the sub-graph, that is, two different trees which contain the two vertices merge into a tree; otherwise, if the two vertices have been on the same tree, then the edge is undesirable, and the minimum weight edge should be removed and tried again, until only one tree in the forest, which means the sub-graph contains $n\text{-}1$edges. The steps are described as follow:

1) Defined V is the set of given n-points, E is the set of edges connected by the n -points.
2) Construct a sub-graph T which only contains n vertices and the edge set is empty.
3) Calculate the weight of each edge in E according to ascending order.
4) Select the smallest weight edge, if the two vertices of the edge belong to different trees, then join it to the sub-graph T, if not, give up this edge.
5) Select the minimum edge from the remaining edges and repeat 4) until T contains $n\text{-}1$edges.

D. Selection Operation

Roulette wheel selection method is used here and we calculate the selection probability of each individual. The selection probability of individuals should be able to reflect the outstanding level. Here the selection probability p_i is determined by formula (2).

$$p_i = f_i \Big/ \sum_{i=1}^{n} f_i \qquad (2)$$

Here, f_i as the fitness value, n is the individual number for population. Roulette wheel selection method implementation is: a random generator is used to select a random number between 0-1, which falls within the scope of individual choice. For example:

25%	30%	20%	15%
……………	…………………	…………	………
Individual 1	Individual 2	Individual 3	Individual 4

E. Crossover Operation

Define p_c as the probability of crossover operation, the probability means that the population has an expected value of $p_c{\times}M$ chromosomes for crossover operation (M is the individual number of the population). Therefore, the greater p_c is, the faster speed of new individual creation is. However, when p_c is too large, the possibility of genetic pattern destruction is greater, then the individual structure with high fitness will soon be destroyed; but if p_c is too small, it will make the search process slow, even stagnant. Here an adaptive process is to determine the crossover probability. Formula (3) is used to determine p_c.

$$p_c = p_{c1} - p_{c2}\frac{f' - f_{avg}}{f'} \qquad (3)$$

In the formula, f' is the larger fitness value needed to cross between the two individuals; f_{avg} is the average fitness value of each generation population. p_{c1}, p_{c2} are variables. When individual fitness is lower than the average fitness value, it shows that the individual is a poor performance individual, a large cross probability should replace it; if the fitness value is greater than the average fitness value, it indicates that the individual performance is good, because it gets corresponding crossover probability on the basis of their fitness value.

F. Mutation Operation

Single-point mutation operator is adopted in this paper, and defined p_m as the probability of mutation in the genetic system. This probability shows that expectation is $p_m \times M$ (M is the individual number of the population) chromosome can be used for mutation in population. Therefore, if p_m is too small, it is hard to generate a new individual; if p_m is too great, then the GA will become a pure random search algorithm. Mutation probability p_m generally ranges from 0.0001 to 0.1.

G. Restore the Fictive Steiner Points

After Selection, Crossover and Mutation, a new population is generated. Checking whether this new population satisfies the ending, if it satisfies the ending, we will decode it, by which we can ensure the location of the fictive Steiner points and get the optimal solution. If this new population dissatisfies the ending, the fitness of each individual needs to be calculated among this new population. The cycle starts again when Genetic factor processing is conducted.

For the population who satisfies the ending, the fictive Steiner point represented in binary needs to be decoded to decimal. Firstly, cut the binary-encoding string into two equal length binary-encoding strings. Secondly convert these two strings into decimal integer. Last but not least, binary to decimal decoding is converted by using the individual-coding method and discretization method of definitional domain. X, Y *is* determined by formula (4).

$$X = \frac{x^{'}}{m}$$
$$Y = \frac{y^{'}}{m} \tag{4}$$

Where, x', y' *is* determined by formula (5).

$$x^{'} = \sum_{i=0}^{n} a_i \times 2^i$$
$$y^{'} = \sum_{i=0}^{n} b_i \times 2^i \tag{5}$$

Where, a_i, b_i is binary code; n is the length of binary code; m is grid subdivision multiples, X, Y is coordinates value. After decoding, the coordinates of fictive Steiner point can be obtained, and then the optimal solution is found.

IV. Computational Experiments

Algorithm mentioned above is programmed in visual c++ and Windows XP platform. The required data is calculated by using the case from SteinLib international test database. There are some 3-6 fixed points of the instance data in Table Ⅰ, in which each cell in the left and right columns represents the horizontal and vertical coordinates.

TABLE I. Instance Data

Test data	Number of fixed points			
	3	*4*	*5*	*6*
Coordinates of the fixed points	0, 1	0.10, 0.08	0.70, 0.96	0.14, 0.45
	0.9, 0	0.10, 0.20	0.88, 0.46	0.21, 0.27
	1, 0.1	0.80, 0.20	0.88, 0.16	0.75, 0.19
		0.80, 0.28	0.19, 0.26	0.87, 0.34
			0.19, 0.06	0.67, 0.84
				0.48, 0.86

Using the Plateau geometry general of Surfactant solution rule and SMT theory, Visualization Experiment [12][13]creates fictive Steiner points in the internal fixed points, and obtains the global shortest path. The results of Visualization Experiment are optimal solutions when the numbers of fixed points are smaller. However, when the number of fixed points increases and distributes irregularly, it's difficult to form a film path and the film path is unstable. Details are introduced in the references [12] and [13]. Compared with the MST, Visualization Experiment is obtained through calculating the length of SMT.The results are shown in TABLE II.

TABLE II. Calculation Results

Methods of solving	Number of fixed points			
	3	*4*	*5*	*6*
Genetic algorithm	1.41749	0.85671	1.66689	1.61558
Visualization Experiment	1.41000	0.85333	1.65500	1.61000
Minimum spanning tree	1.48678	0.90000	1.72862	1.64743

In TABLE II by comparing the experimental data we can found, the lengths obtained by GA are respectively shorter than the MST, approximately 4.67%, 4.81%, 3.58%, 1.94%, and the errors of GA with the optimal solution got by visualization experiments are respectively 0.53 %, 0.39%, 0.71%, 0.35%, less than 1%, which is very close to the optimal solution. So the optimization effect is quite obvious, and it fully meets the conditions for engineering applications.

V. Concluding Remarks

A modified GA is proposed here and a test selection algorithm is used to select the initial population. Crossover operator which meet the adaptive process is constructed to make the GA in accuracy and convergence rate gets improved significantly. The instance of the test and comparison of the results show that the algorithm is fast, adaptable, easy to implement. In dealing with SMT, it can play a certain role for the optimization, and provide a convenient way and tool for the solution of practical application problems in real life. In this paper we only test 3-6 fixed points. When the fixed points are more, for example n=50, the time we get the optimum solution may be very long. Therefore, how to process more fixed point's problems is the next to study.

Acknowledgment

The financial support from the National Natural Science Foundation of China (NSFC) under the grant No.71071078 is acknowledged gratefully. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] L.L. Pinto, G.Laporte. An efficient algorithm for the Steiner Tree Problem with revenue, bottleneck and hop objective functions. European Journal of Operational Research, Volume 207, Issue 1, 16 November 2010, Pages 45-49.

[2] M.C. Dourado, R.A. Oliveira, F. Protti**.** Generating all the Steiner trees and computing Steiner intervals for a fixed number of terminals. Electronic Notes in Discrete Mathematics,Volume 35, 1 December 2009, Pages 323-328.

[3] E. Gassner. The Steiner Forest Problem revisited. Journal of Discrete Algorithms, Volume 8, Issue 2, June 2010, Pages 154-163.

[4] Q. Yuan, F. Qian, W.L. Du· A hybrid genetic algorithm with the Baldwin effect. Information Sciences,. Volume 180, Issue 5, 1 March 2010, Pages 640-652.

[5] S. Kumar, M.P.Singh. Pattern recall analysis of the Hopfield neural network with a genetic algorithm. Computers and Mathematics with Applications,Volume 60, Issue 4, August 2010, P1049-1057.

[6] N. Pillay, W. Banzhaf. An informed genetic algorithm for the examination timetabling problem. Applied Soft Computing, Volume 10, Issue 2, March 2010, Pages 457-467.

[7] S.W. Bae, H. Lee, S. Choi. On exact solutions to the Euclidean bottleneck Steiner tree problem. Information Processing Letters, Volume 110, Issue 16, 31 July 2010, Pages 672-678.

[8] M.A. Henning, M.H. Nielsen, O.R. Oellermann. Local Steiner convexity. European Journal of Combinatorics, Volume 30, Issue 5, July 2009, Pages 1186-1193.

[9] L. Gouveia, P. Moura, A.D. Sousa. Prize collecting Steiner trees with node degree dependent costs. Computers & Operations Research,Volume 38, Issue 1, January 2011, Pages 234-245.

[10] H.M. Jin, L. Ma, Z.M. Wang. Intelligent Optimization Algorithms for Euclidean Steiner Minimum Tree Problem. Computer Engineering, Volume32, Issue10, May 2006, Pages 201-203.

[11] V.R. Chechetkin,V.V. Lobzin. Stability of the genetic code and optimal parameters of amino acids .Journal of Theoretical Biology,Volume 269, Issue 1, 21 January 2011, Pages 57-63.

[12] Z.X. Yang, Y.P. Gao, C.Y. Cheng, Z.Q. Feng, Z.J. Zhang. A Visualization approach for the Steiner minimal tree problem.Systems Engineering-theory & Practice, Issue 7,July 2008, Pages 173-178.

[13] J.Y. Hao, Z.X. Yang. Exp-Geo Algorithm Based on Visualization Experiment.Communications Technology, Volume 43, Issue 1, 2010, Pages 200-202.