

## An Estimation of Distribution Algorithm for Steiner Tree Problem

Yaqing Wang, Hua Wang\*, Guohong Kong

School of Computer Science and Technology

Shandong University

Shunhua Road, Jinan, Shandong Province, 250100 P. R. CHINA

tata1661@126.com, wanghua@sdu.edu.cn, kongguohong@mail.sdu.edu.cn

**Abstract**—As one of the most well-known combinatorial optimization problems, Steiner tree problem is widely applied to optimization in transportation design, biological engineering, and communication networks. It has been proved to be NP complete, though. To solve this problem, researchers have provided many classic solutions. This paper proposes a new method of solving Steiner tree problem by using estimation of distribution algorithms (EDA). The basic idea is to initialize randomly  $n$  trees which contain the source node and the destination nodes. Some elites are selected by the selection operator. The algorithm then constructs a probabilistic model which attempts to estimate the probability distribution of the selected elites. Once the model is constructed, new trees are generated by sampling the distribution encoded by this model. These new trees are then incorporated back into the old population, possibly replacing it entirely. The process is repeated until some termination criteria are met. The algorithm constantly evolves trees to obtain a better solution tree with EDA ideas. This method leads to better performance, reduced time complexity, and optimized solution. Simulation results also show that the algorithm has better performance in searching and converging.

**Keywords**—Steiner Tree; estimation distribution algorithm; approximation algorithm; intelligence optimization; probabilistic model

### I. INTRODUCTION

The Steiner tree problem, which is one of the most well known combinatorial optimization problems, has long been studied[1,2]. Lots of research papers have been contributed to the Steiner tree problem. It has been found that Steiner tree problem can be applied to solve many critical problems in computer science, biological science etc. Researchers have proposed many approaches to solving very large scale integrated circuits design, optical and wireless communication networks[3,4,5,6]. Multi-constraint multicast application optimization, in particular, belongs to a certain Steiner tree problem[7,8,9,10]. Those applications usually require some modifications on the classical Steiner tree problem and hence demand new techniques for solving them. As a result, studying various variations of Steiner tree problems forms a very hot topic in the past twenty years.

The study of Steiner tree problem in graph theory can be traced back to 1961[10]. In 1977, Garey proved that the minimal Steiner tree problem is NP Complete[11]. Some precise algorithms[12,13] have been proposed, but they are not suitable for large-scale problems because they take immense computing time which increases with the exponential increase of node numbers. Generally, heuristic algorithms are used to solve such problems. Although heuristic algorithms cannot obtain best Steiner tree in most cases they can find quasi-Steiner tree in shorter time with the cost close to optimum[14]. Some famous Steiner algorithms like KMB[15] and ADH[16] are proposed. Recently with the prosperity of intelligence algorithms like tabu search, simulated annealing, genetic algorithm, neural network, and ant colony algorithm, many researchers apply such algorithms to the solution of Steiner tree problem[17,18,19,20]. Despite better results, such algorithms have higher demand in setting parameters and designing codes. If the parameters are set inappropriately it is easy for algorithms to fall into local optimization, resulting in low efficiency.

Estimation of distribution algorithms (EDA) are a class of novel stochastic optimization algorithms, which was first introduced in 1996[21]. Statistical theory and the principle of the genetic algorithm are contained in the EDA. Since its inception, it is a relatively new addition to the class of evolutionary computation family. This method has already been successfully applied to optimization of large class instances in two-dimensional and three-dimensional lattices[22], military antenna design[23], multi-objective knapsack [24], ground water remediation design [25,26], amino-acid alphabet reduction for protein structure prediction[27].

This paper proposes an algorithm on the basis of Estimation of distribution algorithms (EDA) for Steiner tree problem. The basic idea is to initialize randomly  $n$  trees which contain the source node and the destination nodes. Some elites are selected by the selection operator. The algorithm then constructs a probabilistic model which attempts to estimate the probability distribution of the selected elites. Once the model is constructed, new trees are generated by sampling the distribution encoded by this model. These new trees are then incorporated back into the old population, possibly replacing it entirely. The process is repeated until some termination criteria are met. The algorithm constantly evolves trees to obtain a better

\* Corresponding author: Hua Wang  
Email address: wanghua@sdu.edu.cn

solution tree with EDA ideas. Simulations made on a variety of undirected and directed graphs indicate that the algorithm is feasible and effective.

This paper consists of six sections. Section 1 describes the research background of Steiner tree problem. Section 2 defines Steiner tree and constructs its mathematical model. Section 3 briefs estimation of distribution algorithm. Section 4 describes the method and detailed steps of solving Steiner tree problem. Section 5 reports the simulation results and analysis. Section 6 is the summary of the research, and meanwhile suggestions for future research are given.

## II. DEFINITION OF STEINER TREE

Definition 1: Given graph  $G=(V, E)$ ,  $V$  is the node set of graph  $G$ ,  $E$  is the edge set of graph  $G$ ,  $|V|$  is the node number of graph  $G$ ,  $|E|$  is the number of edges or links of graph  $G$ . The cost function of edge  $cost: E \rightarrow R$ , the destination node set  $D \subseteq V$ ,  $m = |D|$ , then the minimal Steiner tree is defined as finding the minimal tree covering all the nodes in  $D$  from graph  $G$ . If the tree has the minimum cost, then it is called Steiner tree. Record the tree as  $T_s(V_T, E_T)$ , where  $D \subseteq V_T \subseteq V, E_T \subseteq E$ .

Therefore the minimal Steiner tree can be formalized as

$$\min \sum_{e \in E_T} cost(e) \quad (1)$$

## III. ESTIMATION OF DISTRIBUTION ALGORITHM

Estimation of distribution algorithms are stochastic optimization techniques that explore the space of potential solutions by building and sampling explicit probabilistic models of promising candidate solutions. This model-based approach to optimization has allowed EDA to solve many large and complex problems[28,29,30].

The idea of the EDA originated in the genetic algorithm(GA). EDAs are susceptible to what is analogous to premature convergence in genetic algorithms. But the evolutionary model of the EDA is different from GA. In a sense, GA simulates the biological evolution from the micro-level, but the EDA simulates the biological evolution from the macro-level. Traditional crossover, mutation and genetic manipulation are not included in the EDA, instead of by the learning and the sampling of the probability model. This new evolutionary model gives the unique properties of the EDA. EDA were successfully applied to optimization of large spin glass instances. This optimization can usually find the best solution and has higher efficiency.

According to the degree of complexity of the probabilistic model as well as the different sampling methods, EDA developed many different specific implementation methods, which can be summarized as the following three main steps. Start with a random population of  $M$  vector, then implement a loop consisting of

1) Select  $N$  vectors using a selection method.

2) We should build a probabilistic model for describing the solution space. According to the assessment of the population, we can choose a good collection of individuals, then construct a description of a probabilistic model of the current solution set by means of statistical learning.

3) New populations are generated by the probabilistic model of random sampling. That is, sample  $M$  vectors from the probability model.

There are many variations. Selection is often done by applying truncation selection to the sampled population. Alternatively, the selected vectors replace a fraction of the selected population from the previous generation rather than the entire population. Likewise, the probability model can be built from scratch at each generation, or can use the model from the previous generation(s) to build the current model. Another source of variation in EDAs is in the allowed structure: UMDA treats each variable independently. MIMIC assumes a chain of interactions, so every variable except the root and the terminal variables are the parent of one node and the child of another. BOA uses a general directed acyclic graph to represent the structure.

## ESTIMATION OF DISTRIBUTION ALGORITHM FOR THE STEINER TREE

The basic idea is to initialize randomly a population of  $n$  individuals and each one has the solution tree. The trees are then scored using a fitness function. This fitness function gives a numerical ranking for each tree, with the higher the number the better the tree. According to the fitness of the trees, elite trees whose fitness are lower than the others are selected. The algorithm then calculates the probability of each edge according to the elites and constructs a probabilistic model which attempts to estimate the probability distribution of the selected trees. Once the model is constructed, new solutions are generated by sampling the distribution encoded by this model and the probabilistic model is updated by the new elite. The operation is carried out until obtaining maximal iterative times or convergence of the results. EDA includes four steps: generated the initial population  $N$ ,  $M$  individuals are selected from the population, probabilistic model is introduced and find the new population  $N$  according to the probabilistic model. EDA pseudocode of tree optimization is given as follows.

**procedure** EDA\_STEINER\_TREE

**input**  $m$

**input** elite

$converge = \text{TRUE}$

**Initializaiton:**

**for**  $i = 0$  to  $n$  **do**

Population\_Init ( $source, edge$ )

$p_{li} = \text{Solution}[i]$

**end-for**

$p_g = \text{Calculate\_Tree\_Fitness}(stree)$

```

    h[elite]=Find_Elite(Solution)
    for i=0 to elite-1 do
        Initial_P(p)
    end-for
    for i = 0 to m do
Sampling:
        for j =0 to n do
            Population_Init (source, edge)

            totalcost = Steiner_TreeCost(tmpree)
            if totalcost < Steiner_TreeCost(pij)
                pij = tmpree
            end-if
            if totalcost < Steiner_TreeCost(pg)
                pg = tmpree
            end-if
        end-for
    end-for
Selection:
    h[elite]=Find_Elite(solution)
Modeling:
    Update_P(p)
    if converge = FALSE
        break
    end-if
end-for
print bestree
end-procedure

```

In the above algorithm, we define the trees in the course of searching optimal tree as solution trees. They are generated randomly at the beginning. The one with minimum cost is found among them. Via the steps of removing circles, and pruning useless node, the rest trees would have lower cost. In this process each tree selects edge by using roulette-wheel function according to the probability of the edge, selects the edge which has the higher probability in great possibilities and keeps in the tree, thus achieving the effect of optimizing Steiner tree. Once this was completed, one could sample this distribution to find new candidate solutions to the problem. Ideally, the repeated refinement of the probabilistic model based on representative samples of high quality solutions would keep increasing the probability of generating the global optimum, after a reasonable number of iteration, the procedure would locate the global optimum or its accurate approximation.

The algorithm has good performance through multiple iterations and random guaranteeing of initialized tree. In the following sections the steps of algorithm operation, including initializing the form of trees, removing circles, finding the elites, calculating the probability and pruning useless nodes, will be introduced.

#### A. Initializing EDA trees

The particle keeps a solution tree. If the number of network node is  $N$ , then the particle is a  $N \times N$  matrix  $x$ , where if  $x[i][j] = 0$ , the edge from node  $i$  to  $j$  has been

selected in the source root tree; otherwise, the edge is not used.

The algorithm initially produces the random tree. Set the node set in the source root tree  $S = \{s\}$ , where  $s$  is the root node of solution tree. Then take node  $i$  randomly which is in the neighboring domain of, but not in the set  $S$  and add node  $i$  to set  $S$  until all the destination nodes belong to set  $S$ . After all the destination nodes are added the algorithm needs to prune the leaf nodes which do not belong to the destination nodes. When one edge is added to the current solution, if the edge make the solution had a circle, we did not add it. And the prune operation is to examine each leaf node, if it is not a destination node and there is an edge  $e_{mi}$ , where the node  $m$  is the parent of node  $i$ , then remove edge  $e_{mi}$  from the tree. Do the same operation to the parent node  $m$  of  $i$  until the node  $m$  is either the destination node or its outdegree is larger than 0. A Random solution tree is then easily generated through the above mentioned steps.

#### B. Finding the elites and probabilistic model

In order to build the probabilistic model, a population of promising solutions should be selected. So we select  $M$  elites which have the lower fitness as the sample from the  $N$  initial trees. We use the PBIL[31] probabilistic model.

The probabilistic matrix  $p = (p_{ij}) (0 \leq i, j < V)$  represents the probability model, where  $p_{ij}$  denotes the probability of selecting the edge  $e_{ij}$  or  $e_{ji}$  and it is initialized as

$p_{ij} = \frac{1}{|E|}$ , every link has the same probabilistic to be chosen for constructing the solution expanding the search scope to some extent. In the  $t+1$  iteration, the matrix is updated as the following formula

$$p_{ij}(t+1) = \rho * p_{ij}(t) + (1-\rho) * (1 + \frac{1}{N} \sum_{l=1}^N x_{ij}^l), 1 \leq i, j \leq V \quad (2)$$

Where

$$x_{ij}^l = \begin{cases} 1, & \forall e_{ij} \text{ or } e_{ji} \in l \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$\rho$  is the control factor of the old probabilistic to effect the new selection. The larger of the  $\rho$  value indicates more contribution of the information learned from the history.

### SIMULATION RESULTS AND ANALYSIS

In order to test and improve the performance of the algorithm, we made analysis in terms of three aspects in the simulation process, including the influence of initial population size on search effect, the effect of destination node scale on convergence result, and the comparison of algorithm performance. The algorithm we proposed was programmed with VC6.0 and run on the machine Intel

Pentium® 4 3.00 GHz, 2G memory in the operating system of Windows 7.

#### A. The influence of the initial population size

The size of initial population size plays a critical role in the algorithm in this study. The bigger the size, the more likely the algorithm finds best solution, and the more time it takes the algorithm to converge. Proper size of the initial population enables the algorithm to obtain satisfactory cost with little expense. We did lots of simulation experiments in order to get proper size.

First, in the topology of 60 nodes, we obtained the relative curves between the initial population size and Steiner tree cost and between the initial population size and convergence time. The selection operator is truncation selection with threshold  $\tau = 50\%$ , which selects the 50% best solutions. The results are shown in Fig.1 and Fig.2. If we used Steiner tree cost  $\times$  convergence time as a combined metric to evaluate convergence result and speed in the topology scale of 60 nodes, it is most suitable to set the initial population size as 90.

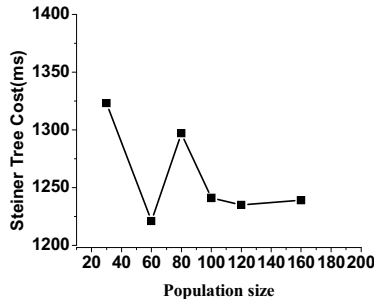


Figure.1. Steiner tree cost with different initial population sizes

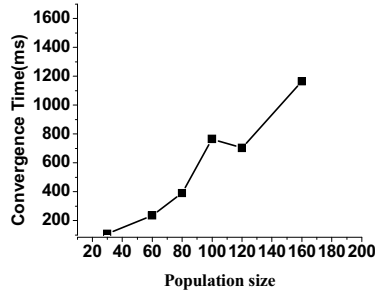


Figure.2. Convergence time with different initial population sizes

#### B. The effect of the destination node scale on convergence speed

Fig.3 and Fig. 4 illustrate the tree cost and iteration while the algorithm converges with different number of destination node in the topology of 80 nodes. It is obvious that the more the destination nodes the higher the cost of the Steiner tree which covers all the destination nodes. With the increase of the number of destination nodes

iteration of the algorithm does not increase in a linear way. This is because the more the destination nodes, the fewer the candidate Steiner trees covering all the destination nodes, and the easier the algorithm converges. But the time increases to find a candidate Steiner tree covering all the destination nodes. And the iteration does not increase continuously. Fig. 4 shows that the number of iteration reaches its maximum value of 20 when the number of destination is 40, i.e. 50% of the total node number.

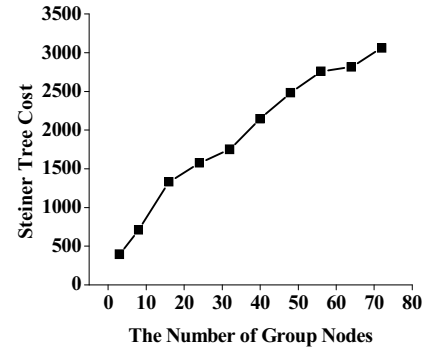


Figure.3 Comparison of tree cost with different node scales

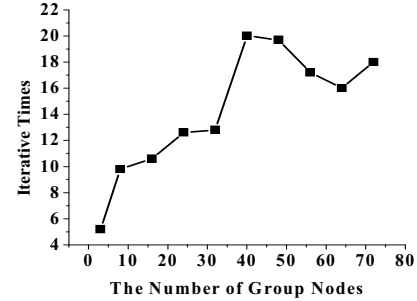


Figure.4. Comparison of iteration with different node scales

#### C. The comparison with other algorithms

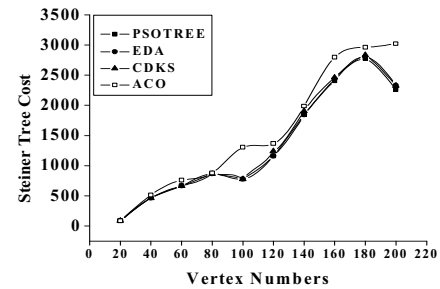


Figure.5 Comparison of cost value on directed graph in different scales

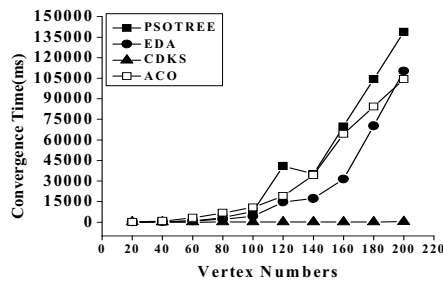


Figure.6 Comparison of convergence time in directed graphs in different scales

We compared with the CDKS and other intelligent algorithms PSOTREE and ACO. The Fig.5 shows that the results of the intelligent algorithms are better than the CDKS owing to they needed more iterations to find the best solution. So more time is needed for the EDA than the CDKS, but it needed less time than the other algorithms, as we can see from the Fig.6.

## VI CONCLUSION

The paper proposed an optimizing Steiner minimal tree algorithm based on EDAs. The algorithm adopted the thought of tree shape change via learning from the probabilistic model to solve Steiner tree problem. This algorithm can be used for undirected Steiner tree. Simulation results show that the results are almost the same as the best one, and this algorithm has better performance in searching and converging speed. It can also be seen that the performance of the algorithm is not much influenced by network scale, indicating that it is more applicable to large-scale topology and provides a better choice for the Steiner problem.

## ACKNOWLEDGMENT

The study is supported by National Natural Science Foundation of China (NSFC No. 60773101), Natural Science Foundation of Shandong Province (Grant No. ZR2011FM021), and Science and Technology Development Program of Jinan (Grant No.201102010), The National Innovating and Enterprising Projects for College Students of china, 2013

## REFERENCES

- [1] M. Bern, R. Graham, The shortest-network problem, *Scientific American* 260 (1989) 84-89.
- [2] H. Kuhn, Steiner's problem revisited, *Studies in Optimization* 10 (1974) 52-70.
- [3] J. Al-Karaki, A. Kamal, Routing techniques in wireless sensor networks: A survey, *IEEE Wireless Communications* 11 (2004) 6-28.
- [4] K. Kalpakis, K. Dasgupta, P. Namjoshi, Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks, *Computer Networks* 42 (2003) 697-716.

- [5] D. Du, *Steiner Tree Problems in Computer Communication Networks*, World Scientific Publishing Company, Singapore, 2008.
- [6] E. Lloyd, G. Xue, Relay node placement in wireless sensor networks, *IEEE transactions on Computers* 56 (2007) 134-138.
- [7] C.-F. Wang, R.-H. Jan, A factoring approach for the Steiner tree problem in undirected networks, *Information Sciences* 177 (2007) 2418-2435.
- [8] H. Wang, Z. Shi, A. Ge, C. Yu, An optimized ant colony algorithm based on the gradual changing orientation factor for multi-constraint QoS routing, *Computer Communications* 32 (2009) 586-593.
- [9] H. Wang, Z. Shi, S. Li, Multicast routing for delay variation bound using a modified ant colony algorithm, *Journal of Network and Computer Applications* 32 (2009) 258-272.
- [10] J. Edmonds, Optimum branchings, *Journal of Research of the National Bureau of Standards* 71B (1967) 233-240.
- [11] M. R. Garey, R. L. Graham, D. S. Johnson, The complexity of computing steiner minimal trees, *SIAM Journal on Applied Mathematics* 32 (1977) 835-859.
- [12] F. K. Wang, Steiner tree problems, *Networks* 22 (1992) 55-89.
- [13] P. Winter, Steiner problem in networks: A survey, *Networks* 17 (1987) 129-167.
- [14] S. Voß, Steiner's problem in graphs: Heuristic methods, *Discrete Applied Mathematics* 40 (1992) 43-72.
- [15] L. Kou, G. Markowsky, L. Berman, A fast algorithm for Steiner trees, *Acta Informatica* 15 (1981) 141-145.
- [16] V. J. Rayward-Smith, On finding Steiner vertices, *Networks* 16 (1986) 283-294.
- [17] H. Wang, Z. Shi, J. Ma, G. Wang, The tree-based ant colony algorithm for multi-constraints multicast routing, *IEEE The International Conference on Advanced Communication Technology (ICACT)*, 2007, pp. 1544-1547.
- [18] S. Chakraverty, A. Batra, A. Rathi, Directed convergence heuristic: A fast & novel approach to Steiner Tree Construction, *2006 IFIP International Conference on Very Large Scale Integration*, 2006, pp. 255-260.
- [19] S. Gosavi, S. Das, S. Vaze, G. Singh, E. Buehler, Obtaining subtrees from graphs: An ant colony approach, in: *Swarm Intelligence Symposium, SIS '03*, 2003, pp. 160-166.
- [20] W. Yang, T. Guo, An ant colony optimization algorithm for the minimum steiner tree problem and its convergence proof, *ACTA Mathematicae Applicatae Sinica* 29 (2006) 352-361.
- [21] S. Baluja, *Population-Based Incremental Learning: A method for Integrating Genetic Search Based Function Optimization And Competitive Learning*. Technical report CMU-CS-94-163. Carnegie Mellon University, 1994.
- [22] M. Pelikan, A. K. Hartmann, Searching for ground states of Ising spin glasses with hierarchical BOA and cluster exact approximation, in: M. Pelikan, K. Sastry, E. Cantú-Paz (Eds.), *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, Springer, 2006, pp. 333-349.
- [23] T.-L. Yu, S. Santarelli, D. E. Goldberg, Military antenna design using a simple genetic algorithm and hBOA, in: M. Pelikan, K. Sastry, E. Cantú-Paz (Eds.), *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, Springer, 2006, pp. 275-289.
- [24] R. Shah, P. Reed, Comparative analysis of multiobjective evolutionary algorithms for random and correlated instances of multiobjective d-dimensional knapsack problems,

- European Journal of Operational Research 211(3)(2011)466-479.
- [25] R.Arst, B.S. Minsker, D.E. Goldberg, Comparing advanced genetic algorithms and simple genetic algorithms for groundwater management, in: Proceedings of the American Society of Civil Engineers, ASCE, Environmental & Water Resources Institute, EWRI, 2002 Water Resources Planning & Management Conference, Roanoke, VA, 2002.
  - [26] M.S. Hayes, B.S. Minsker, Evaluation of advanced genetic algorithms applied to groundwater remediation design, in: Proceedings of the American Society of Civil Engineers, ASCE, Environmental & Water Resources Institute, EWRI, World Water & Environmental Resources Congress 2005 & Related Symposia, Anchorage, AK, 2005.
  - [27] J. Bacardit, M. Stout, J.D. Hirst, K. Sastry, X. Llorà, N. Krasnogor, Automated alphabet reduction method with evolutionary algorithms for protein structure prediction, in: Genetic and Evolutionary Computation Conference, GECCO-2007, 2007, pp. 346–353.
  - [28] Pelikan, M. & Sastry, K. *Scalable optimization via probabilistic modeling: From algorithms to applications*. Vol. 33 (Springer Verlag, 2006).
  - [29] Kollat, J. B., Reed, P. & Kasprzyk, J. A new epsilon-dominance hierarchical Bayesian optimization algorithm for large multiobjective monitoring network design problems. *Advances in Water Resources* **31** (2008), 828-845.
  - [30] Shah, R. & Reed, P. Comparative analysis of multiobjective evolutionary algorithms for random and correlated instances of multiobjective d-dimensional knapsack problems. *European Journal of Operational Research* **211** (2011), 466-479.
  - [31] Baluja, S. Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning. (DTIC Document, 1994).