

# The Generalized Approaches of Genetic Algorithms on Constrained Minimum Spanning Tree Problems\*

Gengui Zhou, Yihua Zhu, Xiaobin Weng, Feng Ye

College of Business Administration  
Zhejiang University of Technology  
Hangzhou 310032, P.R. China

genguiz@yahoo.com

**Abstract** - This paper explores the generalization of problem-solving by Genetic Algorithms (GAs) for a class of constrained Minimum Spanning Tree (MST) problems. These constrained MST problems are quite different from each other in constraints and usually NP-hard but of high practical importance. The paper emphasizes that it is possible for GAs to generalize the problem-solving as long as the problems at hand can be generalized in data structure of solutions. The proposed method adopts only one solution encoding and one algorithm to deal with three constrained MST problems. Numerical experiments show that the proposed GAs approach can respectively compete with the existing heuristic algorithms on these constrained MST problems.

**Index Terms** - Genetic Algorithms, Minimum Spanning Tree, Degree Constraint, Leaf Constraint, Capacity Constraint.

## I. INTRODUCTION

Human beings have been pursuing the generalized method to solve varieties of problems while formulating new problems and creating new methods in the real-world. However, the real-world is so versatile and intricate that it is really not easy for the traditional optimisation techniques to deal with. For different problems, even if they only differ a little in objective functions or constraints, different methods have to be designed to deal cope them. Genetic Algorithms (GAs) simulates the natural evolutionary process of human beings by starting from a set of candidate solutions of the problem in coding space instead of a single one in solution space and improving them step by step through biological evolutionary process such as genetic inheritance and Darwinian strife for survival, and obtains the optimal or near-optimal solutions of the problem at hand. This new idea has really brought about a revolutionary development on optimisation techniques in all fields [1,2]. But the merit of GAs does not only lie in its new idea in algorithm design. The most important of it may rest with its potential power on the generalization of problem-solving. In this paper we try to develop it to some extent on a class of constrained Minimum Spanning Tree (MST) problems.

The classical MST problem has been used to help solve many network optimisation problems such as transportation, telecommunication network design, distribution systems and so on [3,4]. On the base of the MST, many extended problems of MST have been formulated and studied in the last decades as they are more realistic representative for the practical network optimisation problems in the real-world[5]. Among them, the most common extended MST problems are the degree-constrained minimum spanning tree (dc-MST) problem by Narula and Ho (1980)[6], the leaf-constrained minimum spanning tree (lc-MST) problem by Fernandes and L. Gouveia (1998)[7], and the capacitated-constrained minimum spanning tree (cc-MST) problem by Chandy and Lo (1973)[8].

Compared with the simple MST problems, those constrained formulations of MST are generally *NP-hard* and no polynomial-time algorithms for them exist [9]. We can not directly apply those fine polynomial-time MST algorithms to deal with them, such as those suggested by Kruskal [10], Prim [11], and so on. Actually, only some heuristic algorithms exist for the constrained MST problems. However, those heuristic algorithms are quite different in algorithms when solving these constrained MST problems even if not considering their efficiency and effectiveness.

In this paper a new approach by using GAs is put forward to deal with all these constrained MST problems. As all these constrained MST problems have the same data structure in solution characterized as a tree, regardless of their differences in objective functions and constraints, they can be generally approached by one genetic algorithm which adopts only one solution encoding and one algorithm. In encoding a spanning tree, a tree-based genetic representation is adopted as it is capable of representing all possible trees [12,13,14]. In designing the algorithm, the procedure only differs in evaluating individuals as different constrained MST problems have different constraints. Numerical experiments show the high effectiveness of the proposed GAs approach over the existing heuristic algorithms and the possibility of GAs on the

\* This work is partially supported by supported by National Nature Science Foundation of China (No. 60074035) and Zhejiang Provincial Nature Science Foundation (No. 600042).

generalization of problem-solving techniques for some combinatorial optimisation problems.

## II. PROBLEM FORMULATIONS

Consider a connected undirected graph  $G = (V, E)$ , where  $V = \{v_1, v_2, \dots, v_n\}$  is a finite set of *nodes* representing terminals or telecommunication stations *etc.*, and  $E = \{e_1, e_2, \dots, e_m\}$  is a finite set of *edges* representing connections between these terminals or stations. Each edge has an associated positive real number denoted with  $W = \{w_1, w_2, \dots, w_m\}$  representing distance, cost and so on.

Let  $x = (x_1, x_2, \dots, x_m)$  be defined as follows:

$$x_i = \begin{cases} 1, & \text{if edge } e_i \text{ is selected in a spanning tree,} \\ 0, & \text{otherwise.} \end{cases}$$

Then a spanning tree of graph  $G$  can be expressed by the vector  $x$ . Let  $T$  be the set of all such vectors corresponding to spanning trees in graph  $G$ , the well-known MST problem can be formulated as:

$$\text{Min } \{z(x) = \sum_{i=1}^m w_i x_i \mid x \in T\} \quad (1)$$

The degree-constrained MST problem means that there is a degree constraint on each node such that, at each node  $v_j$ , the corresponding degree value  $d_j$  is at most a given value  $b_j$ , i.e., there is a additional constraints on the vector  $x$ ,

$$d_j \leq b_j, \quad v_j \in V, \quad x \in T.$$

The leaf-constrained MST problem means that there are exactly  $k$  leaves (degree 1) on a tree solution. If we add the binary variables ( $u_j = 1, 2, \dots, n$ ) such that  $u_j = 1$  if node  $v_j$  has degree 1 in the solution and  $u_j = 0$ , otherwise. The constraint on the number of leaves can be written as:

$$\sum_{j=1}^n u_j = k.$$

The capacitated-constrained MST problem means that all nodes or items of unit size can always be packed in  $\lceil |S|/\kappa \rceil$  sub-trees or bins if the sub-tree capacity is denoted  $\kappa$ . The additional constraint is as follows:

$$\sum_{v_j \in S} x_i \leq |S| - \left\lceil \frac{|S|}{\kappa} \right\rceil, \quad S \subseteq V \setminus \{v_1\}, |S| \geq 2,$$

where  $v_1$  could be a root node. The above three constrained MST problems are all quite different in mathematical formulation and generally are *NP-hard* so that only quite

different heuristic algorithms have been proposed to deal with them. But whatever the objectives or constraints of all these constrained MST problems are, the solutions for them are ever a tree. Therefore these three constrained MST problems have the same data structure in solution. It makes it possible for GAs to deal with all of them by using only one algorithm as GAs is capable of coping with any objectives and any constraints only if the data structure of the solved problem is determined.

## III. GENETIC ALGORITHMS APPROACH

### A. Genetic Representation

As with the MST problem, two main factors should be taken into consideration if we want to keep its tree topology structure in the genetic representation: one is the connectivity among nodes; the other is the degree value of each node. The intuitive idea of encoding a tree solution is to use a two-dimension structure for its genetic representation. One dimension encodes an ordering of the nodes of a spanning tree; another dimension encodes the degree value of each node. Therefore, we may have a  $2 \times n$  matrix to represent a chromosome for an  $n$ -node tree. The genes in node dimension take a permutation with integers from 1 to  $n$  exclusively; the genes in degree dimension take the integers from 1 to  $b$  inclusively ( $b$  is the largest degree value for all nodes). We denote this genetic representation as tree-based permutation.

On an undirected tree like the MST problem, we can take any node (say, node 1) to be the root node. All other nodes can be regarded as being connected to it hierarchically. For any node (current node), the node incident to it on the upper hierarchy is called its predecessor node and the node incident to it on the lower hierarchy is called its successor node. Obviously, the root node has no predecessor node and a leaf node has no successor node. Based on this observation, the tree-based permutation of such a tree can be encoded as the following procedure.

**procedure:** tree-based permutation encoding

**step 1:** Select node 1 (root node) as the current node in a labeled tree  $T$ , put it as the first digit in the node dimension of the permutation and its degree value as the first digit in the degree dimension.

**step 2:** Check all successor nodes of the current node from the left branch to the right branch. If there are successor nodes, let the leftmost successor node be the current node, then go to **step 3**. Otherwise, go to **step 4**.

**step 3:** Put the label digit of the current node in the node dimension and its degree value in the degree dimension (here we build the permutation by appending digits to the right), then go to **step 2**.

**step 4:** Delete the current node and its adjacent edge from the

tree, let its predecessor node be the current node.  
**step 5.** If all nodes have been checked, stop; otherwise, go to **step 2**.

Fig.1 illustrates an example of this tree-based permutation.

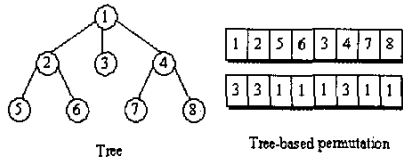


Fig. 1 A rooted tree and its tree-based permutation

For the initial population, each chromosome can be generated randomly. However, in order to keep the connectivity between nodes, the genes in the degree dimension need to satisfy the following conditions: The degree value of each node in degree dimension should be no less than 1 and no greater than a limited upper bound, which guarantees that the total degree value summed over all nodes is  $2(n-1)$  for an  $n$ -node tree.

Any encoding randomly generated in the above way can be decoded into a tree. Suppose that the node dimension for an individual  $P$  is represented as  $P_1(k)$ ,  $k = 1, 2, \dots, n$  and the degree dimension for an individual  $P$  as  $P_2(k)$ ,  $k = 1, 2, \dots, n$ . The decoding procedure for each individual is as follows (for the convenience of the procedure operation, the first gene value in the degree dimension should be increased by one):

**procedure:** tree-based permutation decoding

**step 1:** Set  $k \leftarrow 1$  and  $j \leftarrow 2$ .

**step 2:** Select the node  $r = P_1(k)$  and the node  $s = P_1(j)$ , add the edge from  $r$  to  $s$  into a tree.

**step 3:** Let  $P_2(k) \leftarrow P_2(k) - 1$ ,  $P_2(j) \leftarrow P_2(j) - 1$ .

**step 4:** If  $P_2(k) = 0$ , let  $k \leftarrow k - 1$ , otherwise, go to **step 6**.

**step 5:** If  $j = n$ , stop, otherwise, go to **Step 4**.

**step 6:** If  $P_2(j) \geq 1$ , let  $k \leftarrow j$ ,  $j \leftarrow j + 1$ , go to **Step 2**, otherwise,  $j \leftarrow j + 1$ , go to **Step 2**.

Obviously, any spanning tree can be encoded by this representation scheme and each encoding represents a spanning tree. The relation between the encoding and the spanning tree is not in general a one-to-one mapping because different chromosomes may represent the same spanning tree. But it is possible to represent all possible spanning trees on a complete graph. It is important to point out that this encoding keeps the structure of a tree, so it possesses the locality in the sense that small changes in the representation (such as the mutation operation) make small changes in the tree.

### B. Genetic Operations

Genetic operations are used to alter the genetic composition of chromosomes or individuals. Order crossover,

exchange mutation and insertion mutation are adopted in this paper.

Order crossover is only operated on the node dimension, not on the degree dimension. The operation on any two individuals can be illustrated by Fig. 2.

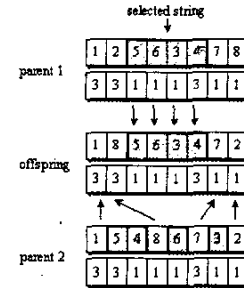


Fig. 2 Order crossover on nodes

Exchange mutation selects two genes (nodes) at random and then swaps the genes (nodes). This mutation is essentially a 2-opt exchange heuristic. The operation can be illustrated by Fig. 3.

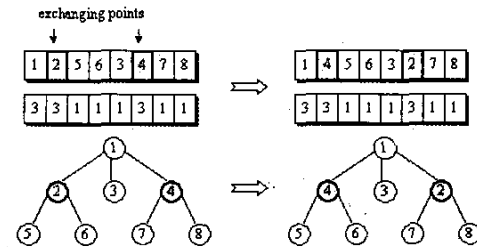


Fig. 3 Exchange mutation on nodes

Insertion mutation selects a string of genes (branch) at random and inserts it on a random gene (node). The operation on any individual can be illustrated by Fig. 4. Obviously, if the string of genes is inserted on a leaf node, it will decrease the number of leaves.

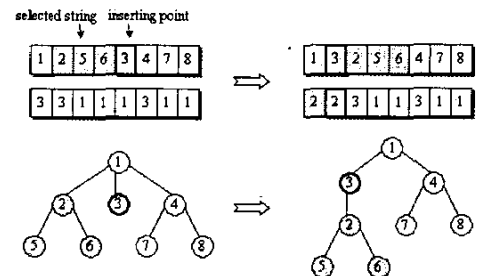


Fig. 4 Illustration of insertion mutation

### C. Modification

For the constrained MST problem, there is the degree-constraint, leaf-constraint, and capacitated-constraint for all spanning tree solutions. Here, we may randomly select a string (branch) and insert it on another randomly selected leaf

to decrease the degree of nodes, the number of leaves, or capacity on each sub-tree as Fig. 5 shows. If we select a string and insert it on an intermediate node, it may increase the number of leaves. Repeat this operation until the constraints are satisfied.

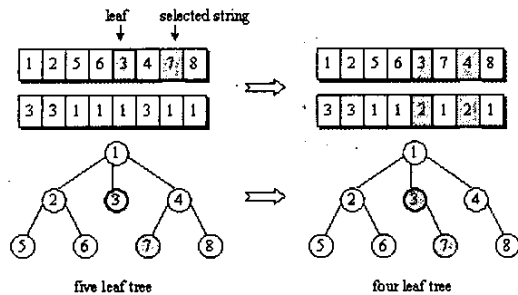


Fig. 5 Illustration of leaves modification

#### D. Evaluation and Selection

Evaluation is the association of each individual with a fitness value which reflects how good it is. We simply take the objective value of Eq. (1) for each individual's fitness value after its decoding from genotypic representation to phenotypic representation.

As to selection, we adopt  $(\mu + \lambda)$  - selection strategy. In order to avoid the premature convergence of the evolutionary process, our selection strategy only selects  $\mu$  different best individuals from this evolving pool. If there are no  $\mu$  different individuals available, the vacant pool of population is filled with renewal individuals.

Finally, using the evaluation-selection-reproduction cycle consists of the whole procedure of the GAs approach on the constrained MST problem.

#### IV. NUMERICAL ANALYSIS

In testing our GAs approach on the dc-MST problem, five randomly generated numerical examples with 10 to 50 nodes were tested. The weights defined on each edge are integers which are randomly generated and uniformly distributed over [10, 100] and the constrained degree value is  $b = 3$ . Here we do not try to find the optimal solution as the problem scale is a little larger. Compared with their lower bounds (LB) which can be directly got by the Prim's MST algorithm without considering the degree constraint, the results by our GAs approach are within less than 7% of their lower bounds as showed in Table 1.

TABLE 1

COMPERISON BETWEEN RESULTS BY GA AND THEIR LB				
Problem (nodes)	Lower Bound (LB)		Genetic Algorithms (GAs)	
	min_value	degree	min_value	%

10	117	5(2)	123	5.13
20	233	4(2)	237	1.72
30	316	4(4)	327	3.48
40	419	7(3)	449	7.16
50	513	6(4)	554	7.99

%, percentage of results by GAs to LB.

All the results are tested on HP9000 Model 715/100 workstation. From Table 1 we can conclude that they are much close to the optimal solutions. The figures in parenthesis are the number of nodes violating the degree constraint. If we arbitrarily modify those vertices violating the degree constraint, the results are always worse than those got by the GAs approaches.

We design the following numerical examples with different leaf-constraints from a class of randomly generated instances from 10 nodes to 30 nodes and test them with the proposed GAs approach on lc-MST problem. The cost matrix was taken as the integer uniformly and randomly distributed in the range of [10, 80]. All the results are listed in Table 2.

TABLE 2  
NUMERICAL RESULTS OF THE lc-MST PROBLEM

Problem Scale (nodes)	k	Lower Bound (Prim's algorithm)	Upper Bound (Fernandes's heuristic)	GAs Results
10	3	174 (5)	186 (3)	186 (3)
20	3	266 (8)	279 (3)	277 (3)
20	5	266 (8)	272 (5)	272 (5)
30	5	367 (12)	408 (5)	402 (5)
30	8	367 (12)	393 (8)	389 (8)

The figures in parenthesis represent the number of leaves. Table 2 shows that the GAs results are all within the interval of their lower and upper bounds. When the problem scales are not larger, there is no difference between GAs approach and Fernandes & Gouveia's heuristic. But, with the increase of the problem scale, we can simply obtain a better solution of the lc-MST problem by the proposed GAs approach than Fernandes & Gouveia's heuristic.

In order to illustrate the effectiveness of the proposed GAs approach on the cc-MST problem, we present a numerical example given out by Gavish [15]. The example consists of a cc-MST problem with 16 nodes, a unit traffic between each node and node 1, and a capacity restriction  $\kappa=5$ . The cost matrix for example is presented in Table 3.

Gavish adopted an augmented Lagrangean-based algorithms to solve this problem and got the optimal solution 8526. By the proposed GAs, we also got the optimal solution 8526 and its corresponding topology of a tree.

#### V. CONCLUSION

Variety of constrained MST problems, which are usually *NP-hard*, were studied by different heuristic algorithms. This paper developed a new approach by using GAs approach and applied it to three different constrained MST problems. The proposed GAs only adopts one solution encoding and one algorithm successfully to solve these constrained MST problems. The numerical analysis shows that the proposed GAs approach is effective to all these constrained MST problems and capable of competing respectively with the existing heuristic algorithms.

From the research work we can get concluded that this new GAs approach is further capable of dealing with other constrained MST problems, such as the MST with fuzzy parameters or multiple criteria. On the whole, the research work shows that different problems can be approached by GAs in the sense of one algorithm if they have the same data structure of solutions, which lights on the potential power of GAs on problem-solving, particularly the generalization of problem-solving.

TABLE 3.  
THE COST MATRIX FOR THE cc-MST PROBLEM ( $n = 16, \kappa = 5$ )

$i/j$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1616	1909	246	622	829	1006	2237	399	1717	632	1191	2116	824	1336	1519
2		2996	1419	2217	1213	2046	3753	1516	1180	1997	552	3622	2423	1367	862
3			1893	1543	1792	2785	1362	1667	3556	2332	2446	1248	1508	3233	3287
4				799	593	1188	2369	242	1670	857	962	2243	1004	1348	1425
5					1230	1253	1625	761	2301	801	1748	1509	206	1873	2119
6						1758	2597	480	1883	1449	663	2463	1420	1701	1573
7							2703	1399	1470	454	1849	2612	1350	960	1470
8								2238	3922	2304	3231	137	1442	3476	3743
9									1889	1029	1009	2108	959	1586	1628
10										1693	1437	3808	2480	511	340
11											1685	2206	909	1205	1603
12												3098	1952	1429	1100
13													1331	3368	3624
14														2038	2309
15															578

## REFERENCES

- [1] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*, 2<sup>nd</sup> ed. Springer-Verlag, Berlin, 1994.
- [2] D. Fogel, *Evolution Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, Piscataway, NJ, 1995.
- [3] A. Kershenbaum, *Telecommunications Network Design Algorithms*, McGraw-Hill, Inc., 1993.
- [4] R.L. Graham and P. Hell, "On the history of the minimum spanning tree problem," *Annals of the History of Computing*, vol. 7, pp. 43-57, 1985.
- [5] D. Narsingh and K. Nishit, "Computation of constrained spanning trees: a unified approach," in *Network Optimization*, P.M. Pardalos, D. Hearn and W. Hager, Eds., Springer-Verlag, Berlin, pp. 194-220, 1997.
- [6] S.C. Narula and C.A. Ho, "Degree-constrained minimum spanning tree," *Comput. & Ops. Res.*, vol. 7, pp. 239-249, 1980.
- [7] L.M. Fernandes and L. Gouveia, "Minimal spanning trees with a constraint on the number of leaves," *European Journal of Operational Research* vol. 104, pp. 250-261, 1998.
- [8] K.M. Chandy and T. Lo, "The capacitated minimum spanning tree," *Networks*, vol. 3, pp. 173-182, 1973.
- [9] M.R. Garey and D.S. Johnson, *Computers and intractability: a guide to the theory of NP-Completeness*, W.H. Freeman and Co., San Francisco, 1979.
- [10] J.B. Kruskal Jr, "On the shortest spanning subtree of a graph and the travelling salesman problem," *Proceeding of ACM*, vol. 7, pp. 48-50, 1956.
- [11] R.C. Prim, "Shortest connection networks and some generalizations," *Bell Sys. Tech. J.*, vol. 36, pp. 1389-1401, 1957.
- [12] G. Zhou and M. Gen, "A note on genetic algorithm approach to the degree-constrained spanning tree problems," *Networks*, vol. 30, pp. 105-109, 1997.
- [13] G. Zhou and M. Gen, "A genetic algorithm approach on tree-like telecommunication network design problem," *Journal of Operational Research Society*, vol. 54, no. 3, pp. 248-254, 2003.
- [14] G. Zhou and Y. Zhu, "A new approach on the optimization of a capacitated tree-structured network design problem," *Proceedings of the Second International Conference on Information and Management Science*, Chengdu, China, pp. 145-149, 2003.
- [15] B. Gavish, "Augmented Lagrangean-based algorithms for centralized network design," *IEEE transaction on Commun.*, COM-33, pp.1247-1257, 1985.