A Note on Genetic Algorithms for Degree-Constrained Spanning Tree Problems

Gengui Zhou, Mitsuo Gen

Department of Industrial and Systems Engineering, Ashikaga Institute of Technology, Ashikaga 326, Japan

Received 13 August 1996; accepted 2 April 1997

Abstract: The degree-constrained spanning tree problem is of high practical importance. Up to now, there are few effective algorithms to solve this problem because of its *NP-hard* complexity. In this paper, we present a new approach to solve this problem by using genetic algorithms and computational results to demonstrate the effectiveness of the proposed approach. © *1997 John Wiley & Sons, Inc.* Networks **30:** 91–95, 1997

Keywords: minimum spanning tree; degree constraint; genetic algorithms

1. INTRODUCTION

The minimum spanning tree (MST) problem has a venerable history [10] in combinatorial optimization and computer science. Given a finite connected graph, the problem is to find a minimum-cost subgraph spanning all vertices. Consider a connected undirected graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a finite set of *vertices* representing terminals or telecommunication stations, etc., and $E = \{e_1, e_2, \ldots, e_m\}$ is a finite set of *edges* representing connections between these terminals or stations. Each edge has an associated positive real number denoted by $W = \{w_1, w_2, \ldots, w_m\}$ representing distance, cost, and so on.

Let vector $\mathbf{x} = (x_1 x_2 \cdots x_m)$ be defined as

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

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

$$\min\{z(\mathbf{x}) = \sum_{i=1}^{m} w_i x_i \,|\, \mathbf{x} \in T\}. \tag{1}$$

If we assume that there are some degree constraints on each vertex such that, at each vertex v_j, the degree value d_j (the number of edges incident to each vertex) is at most a given value b_j, the problem (1) is formulated as the following form:

$$\min\{z(\mathbf{x}) = \sum_{i=1}^{m} w_i x_i \,|\, d_j \le b_j,\, v_j \in V,\, \mathbf{x} \in T\}. \tag{2}$$

Correspondence to: M. Gen; e-mail: {zhou gen}@genlab.ashitech.ac.jp

Contract grant sponsor: Ministry of Education, Science and Culture of the Japanese Government.

Contract grant number: 07045032 (1995.4-1998.3)

 CCC 0028-3045/97/020091-05

TABLE I. Edge weights of the 9-vertext dc-MST problem

j	1	2	3	4	5	6	7	8	9
1	—	224	224	361	671	300	539	800	943
2		—	200	200	447	283	400	728	762
3			—	400	566	447	600	922	949
4				—	400	200	200	539	583
5					—	600	447	781	510
6						—	283	500	707
7							—	361	424
8								—	500
9									—

The above MST problem with degree constraints was denoted as the degree-constrained minimum spanning tree problem (dc-MST) by Narula and Ho (1980) [12]. The problem may arise, for instance, when designing a road system in which at most four roads are allowed to meet at a crossing. In communication networks, a degree constraint limits the vulnerability in case of a drop of a crossing. Therefore, the dc-MST is a more realistic representation of the practical problem than is the MST in this situation.

However, to find the solution of the dc-MST problem, it is difficult to directly apply those fine polynomial-time algorithms such as Dijkstra's [4], Kruskal's [10], Prim's [13], and Sollin's [16]. One intuitive idea is using heuristic algorithms: (1) finding the MST without degree constraints, and (2) modifying the MST combined with the degree constraints. But it is not easy to operate effectively as the problem scale gets larger. Actually, the dc-MST problem can be regarded as a generalization of the traveling salesman problem when at most only two edges are allowed incident to each vertex in the MST problem. Because of this complexity, as the polynomial-time algorithms do not exist, only a few heuristic algorithms to solve this problem were suggested by Narula and Ho [12] and Savelsbergh and Volgenant [15]. In this paper, we put forward a new approach to solve the dc-MST problem by using genetic algorithms (GAs).

2. GENETIC ALGORITHMS APPROACH

Since Holland first proposed GAs in the early 1970s as computer programs that mimic the evolutionary processes in nature [8], the GAs have been demonstrating their power by successfully being applied to many practical optimization problems in last decade [6, 7, 11, 18, 19]. Generally, any evolutionary algorithms designed for a particular problem should address two main factors: genetic representation of solutions to the problem and genetic operators that would alter the genetic composition of individuals during the evolutionary process.

In designing the genetic representation for the dc-MST problem, we adopt the *Prüfer number* to encode a spanning tree [5, 14]. This genetic representation is not only capable of equally and uniquely representing all possible spanning trees for a graph but also explicitly contains the information of vertex degree that any vertex with degree d will appear exactly $d - 1$ times in the Prüfer number. Therefore, if any solution violates the degree constraints, it is easy to modify the number of vertex appearing in its corresponding Prüfer number. As to the genetic operators, we adopt uniform crossover and perturbation mutation operators. To guide the GAs approach to evolve toward the optimal or near-optimal solution of the dc-MST problem, we use the $(\mu + \lambda)$-selection strategy in the evolutionary process [2]. The overall procedure for the dc-MST problem is outlined as follows:

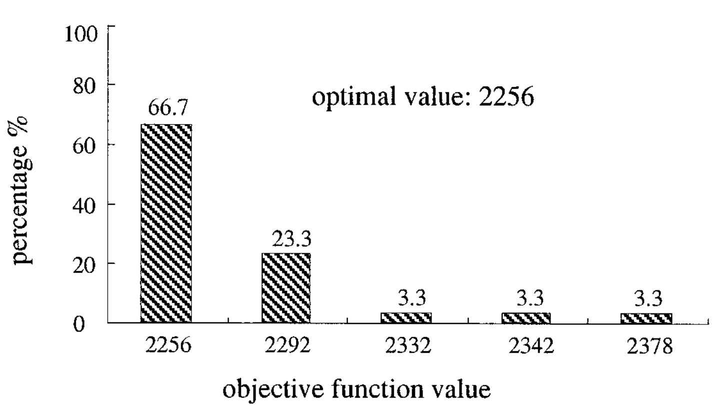

Fig. 1. Solution distribution for the dc-MST using GAs.

TABLE II. Comparison between results by the GAs and their lower bounds

Problem size (vertices)	LB		GAs		Percentage GAs to LB
	min_val	Degree value	min_val	Degree value	
10	117	5 (2)	123	3	5.13
20	233	4 (2)	237	3	1.72
30	316	4 (4)	327	3	4.48
40	419	7 (3)	449	3	7.16
50	513	6 (4)	554	3	7.99

LB: lower bounds obtained by Prim's MST algorithm [13]; min_val: minimal value.

procedure: GAs for dc-MST problem

begin
  $t \leftarrow 0$;
  initialize $P(t)$;
  modify degree $P(t)$;
  evaluate $P(t)$;
  while (**not** termination condition) **do**
    recombine $P(t)$ to yield $C(t)$;
    modify degree $C(t)$;
    evaluate $C(t)$;
    select $P(t + 1)$ from $P(t)$ and $C(t)$;
    $t \leftarrow t + 1$;
  end
end

where $P(t)$ and $C(t)$ are, respectively, the population of parents and offspring in current generation t.

3. COMPUTATIONAL RESULTS

The first numerical example was given by Savelsbergh and Volgenant who solved it using heuristic algorithm denoted as *edge exchanges* [15] and the optimal solution is 2256. It is a 9-vertex undirected complete graph and is given in Table I.

The parameters for the proposed GAs approach are set as follows: population size, *pop_size* = 50; crossover probability, $p_c = 0.5$; mutation probability, $p_m = 0.01$; maximum generation, *max_gen* = 500; the constrained degree value for all vertices $b = 3$; and run 30 times.

By our GAs approach, the optimal solution 2256 can be reached at most times. Figure 1 clearly illustrates that the optimal solutions can be reached with great probability.

In further testing our GAs approach on the dc-MST problem, five randomly generated numerical examples with 10–50 vertices were tested. The weights defined on each edge are integers that are randomly generated and uniformly distributed over [10, 100]. All parameters for the GA are set as before and the constrained degree value for all vertices is $b = 3$. If all degree constraints on the vertices are different, it is still easy for the proposed GAs approach to cope with such a case. Here, we do not try to find the optimal solution because the problem scale is a little larger. Compared with their lower bounds which can be directly found by the MST algorithm without considering the degree constraints, the results by our GAs approach are within less than 8% of their lower bounds, as shown in Table II. The figures in parentheses in the table are the number of vertices 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 approach. Therefore, this shows that our GAs approach has a fine mechanism on operation in a simple way to get the optimal or near-optimal solutions and also verifies the effectiveness of Prüfer number as a genetic representation in the dc-MST problem.

To further verify the effectiveness of the proposed GAs approach, we also applied it to other MST problems, such as the stochastic MST (s-MST) by Ishii et al. (1981)

TABLE III. The expected active cost for the p-MST problem

Test Cases	20 vertices		30 vertices	
	Greedy	GAs	Greedy	GAs
I	54.09	52.50	94.29	95.77
II	67.61	61.11	110.56	109.17
III	62.57	56.97	109.29	100.17
IV	87.09	69.53	122.97	99.95
V	66.94	60.96	108.58	95.76
VI	57.78	53.05	92.82	99.44
VII	82.95	62.47	115.58	107.78
VIII	69.38	64.57	100.81	99.21
IX	60.58	58.10	106.28	103.26
X	65.77	60.96	114.95	120.37

TABLE IV. The best results from heuristic algorithms and GAs approach

Problems size (vertices)	HG	GAs	Improved percent
6	344	302	7.36
7	467	445	4.71
8	563	547	2.84
9	720	630	12.50
10	925	834	9.45
11	1118	979	12.43
12	1209	1095	9.43
13	1392	1284	9.24
14	1777	1549	12.84
15	2078	1823	12.27
16	2292	2010	12.30
17	2615	2389	8.64
18	2888	2475	11.98
20	3554	3233	9.03
30	8500	7742	8.92

HG: heuristic algorithm [17]; improved percent: improved degree on results by GAs to HG.

[9], the probabilistic MST (p-MST) by Bertsimas (1990) [3], and the quadratic MST (q-MST) by Xu (1995) [17]. Here, we only need to relax the degree constraints in the proposed algorithm.

The test experiment on the p-MST problem was based on the research of Abulai et al. [1]. Compared with the greedy algorithm, our GAs approach was tested on 10 data sets with 20 and 30 vertices. The cost of each edge is the Euclidean distance. The probability associated with each vertex in all cases was fixed at 0.1. Table III shows that our GAs approach can get better results at most cases.

Since the crisp equivalent problem of s-MST is the same as the q-MST in formulation, it can be demonstrated together with the q-MST problem. To test the performance of the proposed GAs approach on the q-MST problem, we used 15 test problems with size (n) ranging from 6 to 30 vertices. All graphs are complete graphs with n vertices and $m = n(n-1)/2$ edges. The diagonal elements in the intercost matrix for each q-MST problem are integers that are randomly generated and distributed uniformly over (0, 100]. The off-diagonal elements in the intercost matrix for each q-MST problem are also randomly generated integers distributed uniformly over (0, 20].

Table IV gives out the best results for the q-MST problem by the heuristic algorithm (HG) [17] and our GAs approach. By our GAs approach, much better results can be reached. Compared with the results from the heuristic algorithm, the solution to the q-MST problem can be improved by 9.6% on average and at most by 12.83% with our GAs approach.

4. CONCLUSION

This paper developed a new approach to solve the degree-constrained spanning tree problem by using GAs. Quantities of computational results demonstrate that the proposed GAs approach is capable of getting optimal or near-optimal solutions for the dc-MST problem, as well as for other hard-solving MST problems. The research works also show that genetic algorithms and the Prüfer number as a tree encoding are effective to deal with such spanning tree problems.

This research work was partially supported by the International Scientific Research Program No. 07045032 (1995.4-1998.3) Grant-in-Aid for Scientific Research by the Ministry of Education, Science and Culture of the Japanese Government.

REFERENCES

[1] F. N. Abuali, R. L. Wainwright, and D. A. Schoenefeld, Determinant factorization: A new encoding scheme for spanning trees applied to the probabilistic minimum spanning tree problem. *Proceedings of the Sixth International Conference on Genetic Algorithms* (L. J. Eshelman, Ed.). Morgan Kaufmann, San Francisco, CA (1995) 470–475.

[2] T. Bäck, F. Hoffmeister, and H. Schwefel, A survey of evolution strategy. *Proceedings of the Fourth International Conference on Genetic Algorithms* (R. Belew and L. Booker, Eds.). Morgan Kaufmann, Los Altos, CA (1991) 2–9.

[3] D. J. Bertsimas, The probabilistic minimum spanning tree problem. *Networks* **20** (1990) 245–275.

[4] E. W. Dijkstra, A note on two problems in connection with graphs. *Num. Math.* **1** (1959) 269–271.

[5] J. Dossey, A. Otto, L. Spence, and C. Eynden, *Discrete Mathematics.* Harper Collins, New York (1993).

[6] D. Fogel, *Evolution Computation: Toward a New Philosophy of Machine Intelligence.* IEEE Press, Piscataway, NJ (1995).

[7] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design.* Wiley, New York (1997).

[8] J. H. Holland, *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI (1975).

[9] H. Ishii, H. Shiode, and T. Nishida, Stochastic spanning tree problem. *Discr. Appl. Math.* **3** (1981) 263–273.

[10] J. B. Kruskal, Jr., On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. ACM.* **7** (1956) 48–50.

[11] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Program,* 2nd ed. Springer Verlag, New York (1994).

[12] S. C. Narula and C. A. Ho, Degree-constrained minimum spanning tree. *Comput. Op. Res.* **7** (1980) 239–249.

[13] R. C. Prim, Shortest connection networks and some generalizations. *Bell Syst. Tech. J.* **36** (1957) 1389–1401.

[14] H. Prüfer, Neuer beweis eines satzes über permutationen. *Arch. Math. Phys.* **27** (1918) 742–744.

[15] M. Savelsbergh and T. Volgenant, Edge exchanges in the degree-constrained spanning tree problem. *Comput. Op. Res.* **12** (1985) 341–348.

[16] M. Sollin, Le trace de canalisation. *Programming, Games, and Transportation Networks* (C. Berge and A. Ghouilla-Houri, Eds.). Wiley, New York (1965).

[17] W. Xu, On the quadratic minimum spanning tree problem. *Proceedings of 1995 Japan–China Joint International Workshops on Information Systems,* Ashikaga, Japan (M. Gen and W. Xu, Eds.). (1995) 141–148.

[18] G. Zhou and M. Gen, A new approach to the degree-constrained minimum spanning tree problem using genetic algorithm. *Proceedings of 1996 IEEE International Conference on Systems, Man and Cybernetics.* IEEE Press, Bejing, China (1996) 2683–2688.

[19] G. Zhou and M. Gen, An effective genetic algorithm approach to the quadratic minimum spanning tree problem. *Proceedings of the First Asia-Pacific Conference on Simulated Evolution and Learning,* Taejon, Korea (J-H. Kim, X. Yao, and T. Furuhashi, Eds.). (1996) 395–402.