Pergamon

PII: S0305-0548(97)00039-7

Computers Ops Res. Vol. 25, No. 3, pp. 229–237, 1998

0305-0548/98 $19.00+0.00

AN EFFECTIVE GENETIC ALGORITHM APPROACH TO THE QUADRATIC MINIMUM SPANNING TREE PROBLEM

Gengui Zhou‡ and Mitsuo Gen†§

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

(Received July 1996; in revised form May 1997)

Scope and Purpose—The minimum spanning tree (MST) problem is one of the traditional optimization problems. Its linear objective function makes it easy to deal with. However, if we consider the presence of the interaction effects of the cost between pairs of edges, it results in a new minimum spanning tree problem with a quadratic (rather linear) objective function, which is denoted as the quadratic minimum spanning tree (q-MST) problem. This q-MST problem is of high importance in network optimization design such as oil transportation, communication, etc. Because the problem is *NP-hard* and no effective algorithms exist, we have developed a new approach by using a genetic algorithm (GA) to deal with it.

Abstract—In this paper we present a new approach to solve the q-MST problem by using a genetic algorithm. A skillful encoding for trees, denoted by Prüfer number, is adopted for GA operation. On comparing with the existing heuristic algorithms by 17 randomly generated numerical examples from 6-vertex graph to 50-vertex graph, the new GA approach shows its high effectiveness in solving the q-MST problem and real value in the practical network optimization.

1. INTRODUCTION

The minimum spanning tree (MST) problem is one of the traditional combinatorial optimization problems. Given a finite connected graph, the problem is to find a minimum cost subgraph spanning all vertices. Since it was first formulated in 1926 by Boruvka who is said to have learned about it during the rural electrification of Southern Moravia where he gave a solution to find the most economical layout of a power-line network [6], the MST has been widely applied to many combinatorial optimization problems such as transportation problem, telecommunication network design, distribution systems and so on [7].

Recently, a new formulation of the MST problem that involves searching for the spanning tree of minimum cost under a quadratic cost structure was proposed [14]. The quadratic cost structure means the interaction effects of the cost between pairs of edges while considering the spanning tree with minimum cost. The problem arises for instance when transmitting oil from one pipe to another, the cost may depend on the nature of the interface between two pipes. The same pairwise interaction effect arises in the connection of overground and underground cables or in a transportation or road network with turn penalties. In all these cases, the presence of intercosts results in the minimum spanning tree problem with the quadratic (rather linear) objective, which was denoted as the quadratic minimum spanning tree problem (q-MST for short).

The q-MST problem has been proven *NP-hard* [14], we can not directly use polynomial-time MST algorithms proposed by Kruskal [8], Prim [11] and Dijkstra [4]. When the number of vertices in network is more than twelve, it is very difficult to obtain the optimal solution by branch-and-bound algorithm such as leveling algorithm [2]. Because of this complexity, there have been no effective algorithms proposed

† To whom all correspondence should be addressed (email: zhou@genlab.ashitech.ac.jp).

‡ Gengui Zhou is an Associate Professor in the school of Economic & Trade Administration at Zhejiang University of Technology, China. His primary focus is on mathematical programming, network optimization and genetic algorithms. He has published several articles in international journals such as *Networks, Engineering Design and Automation, and Operations Research* (in Chinese). He received his M.S. degree in Systems Engineering from the East China University of Technology. Now he is pursuing his Ph.D. degree in Industrial and Systems Engineering at the Ashikaga Institute of Technology, Japan.

§ Mitsuo Gen is a Professor in the Department of Industrial and Systems Engineering at Ashikaga Institute of Technology, Japan. He received his M.S. and Ph.D. degrees from Kogakuin University, Japan. His research interest includes reliability analysis, fuzzy operations research, network optimization, neural networks and genetic algorithms. He has published several books including the book of *Genetic Algorithms and Engineering Design* with Dr. Cheng by John Wiley & Sons, Inc. He has published numerous articles in several international journals such as *IEEE Trans. on Rel., European Journal of Operational Research, Computers and Operations Research, Computers and Industrial Engineering, International Journal of Systems Science, Fuzzy Sets and Systems, Networks, Engineering Design and Automation* and so on. He is also a member of Editorial Board of *International Journal of Computers and Industrial Engineering*, and Associated Editor of *Engineering Design and Automation*.

to solve this problem since it was formulated. Two heuristic algorithms in [14] were suggested but the heuristic algorithms can only get some local optima which are a little far from the optimal solution.

The contribution of this paper is to put forward a new approach to deal with the q-MST problem using genetic algorithm (GA) which has been demonstrating its power by successfully being applied to many combinatorial optimization problems in last decade [5,9]. In the paper we will not probe the exact algorithm to get the exact solution for the q-MST problem. We turn to the GA approach to get the optimal or near-optimal solutions compared with the existing heuristic algorithms. In designing the genetic algorithm for the q-MST, we adopt the Prüfer number to encode a tree, which has been successfully applied in the degree-constrained minimum spanning tree problem [15]. In order to enforce the GA approach to evolve to the optimal or near-optimal solutions, we propose a mixed strategy combined with evolution strategy and *roulette wheel* selection in selection operation. Based on the randomly generated numerical example with vertices from 6 to 50, the results obtained by the GA approach show its high effectiveness which can find much better solution than the existing heuristic algorithms and even the optimal solution or near-optimal solution with great probability.

The remainder of this paper is organized as follows. In Section 2 a general description of the q-MST problem is presented. In Section 3 two heuristic algorithms for the q-MST problem are discussed. Our GA approach is in detail discussed in Section 4. Numerical examples and results from both heuristic algorithms and GA approach are demonstrated in Section 5 and the conclusion follows in Section 6.

2. Q-MST PROBLEM

Consider a connected graph $G=(V, E)$, where $V=\{v_1, v_2,\dots,v_n\}$ is a finite set of *vertices* 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 by $W=\{w_1,w_2,\dots,w_m\}$ representing distance, cost and so on.

Let $\mathbf{x}=(x_1x_2\dots x_m)$ be defined as follows:

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

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 spanning trees in graph G, the well-known MST problem can be formulated as:

$$\text{Min}\left\{z(\mathbf{x})=\sum_{i=1}^{m} w_ix_i \mid \mathbf{x}\in T\right\}. \tag{1}$$

If we assume that there is a cost c_{ij} associated with each pair of edges (e_i, e_j) where $i\neq j$ and also there is a direct cost w_i associated with selected edge e_i in the tree, it means that the objective function in (1) is no longer linear and the interactive costs have to be considered in the objective function. Thus, the problem is formulated as the following form:

$$\text{Min}\left\{z(\mathbf{x})=\sum_{i=1}^{m}\sum_{\substack{j=1\\ j\neq i}}^{m} c_{ij}x_ix_j+\sum_{i=1}^{m} w_ix_i \mid \mathbf{x}\in T\right\}, \tag{2}$$

where c_{ij} is also denoted as intercost. The above MST problem with quadratic objective function is denoted as the quadratic minimum spanning tree problem.

In a special case, when only adjacent edges may have nonzero intercosts and non-adjacent edges have no intercosts existing, then the formulation (2) is referred as the adjacent-only q-MST problem, which in general has the same complexity in algorithm. For convenience we only consider the q-MST problem. However, the q-MST problem has been proven *NP-hard* [14], so we can not directly use those polynomial-time MST algorithms to solve it. In the following sections, we will discuss the heuristic algorithms and genetic algorithms for solving the q-MST problem.

3. HEURISTIC ALGORITHMS

Since there are no effective polynomial-time algorithms for the q-MST problem, only two heuristic algorithms were given out in [14].

3.1. *Heuristic algorithm H1 (average contribution method)*

If the edge e_k is singly taken into consideration and all terms are separated from containing index k, then the objective $z(\mathbf{x})$ in (2) can be rewritten as follows:

$$z(\mathbf{x})=\left[w_k+\sum_{j\neq k}(c_{jk}+c_{kj})x_j\right]x_k+z_2(\mathbf{x}),$$

where the term $z_2(\mathbf{x})$ no longer involves x_k. In the summation within bracket there is a total of $(m-1)$ terms, all but $(n-1)$ terms are bound to be zero. Suppose e_k be the edge selected into the solution tree, the average contribution of e_k to the objective can be estimated according to the following formula:

$$p_k=w_k+\frac{(n-1)}{(m-1)}\cdot\sum_{j\neq k}(c_{jk}+c_{kj}),\ 1\leq k\leq m. \tag{3}$$

After evaluating the average contribution p_k for all edges e_k $(k=1,2,\ldots,m)$, the q-MST solution can be obtained by solving the following MST problem:

$$P=\mathrm{Min}\left\{\sum_{k=1}^{m}p_kx_k\mid \mathbf{x}\in T\right\}. \tag{4}$$

3.2. *Heuristic algorithm H2 (sequential fixing method)*

In heuristic algorithm H1, the average contributions of all edges are independently estimated. Suppose that fixing certain edges into the tree would affect the estimated contributions of the remaining edges, the average contributions of the remaining edges are then re-assessed prior to fixing another edge. In this way, the edges in the spanning tree are determined sequentially.

Let U be the index set of the edges which have been spanned into the tree, V be that of the excluded edges whose inclusion would result in a cycle in the tree, and F be that of the free edges which are the candidates for the tree. The average contribution of e_k to the objective in each iteration can be sequentially estimated as follows:

$$q_k=\hat{w}_k+\frac{n_1}{m_1}\cdot\sum_{j\in F,j\neq k}(c_{jk}+c_{kj}), \tag{5}$$

where $m_1=|F|-1$ and $n_1=n-1-|U|$, it means that there are $|F|-1$ terms of which all but $n-1-|U|$ must vanish, and

$$\hat{w}_k=w_k+\sum_{j\in U}(c_{kj}+c_{jk}).$$

The heuristic algorithm can be illustrated as the following procedure:

Procedure: Heuristic H2

Step 1. Let $U\leftarrow\{\phi\}$, $F\leftarrow\{1,2,\ldots,m\}$, $n_1\leftarrow n-1$, and $m_1\leftarrow m-1$.
Step 2. Calculate q_k for all $k\in F$.
Step 3. Select the e_l such that $q_l=\min\{q_k\mid k\in F\}$. Let $x_l\leftarrow 1$, $U\leftarrow U\cup\{l\}$, $F\leftarrow F\backslash\{l\}$ and $n_1\leftarrow n_1-1$. If $n_1=0$, stop.
Step 4. If for any k in F, adding edge e_k results in a cycle with the edges in U, set $x_k\leftarrow 0$ and $F\leftarrow F\backslash\{k\}$. Return to step 2.

4. GENETIC ALGORITHM APPROACH

Unlike conventional combinatorial optimization techniques, the GA requires neither heuristics nor knowledge of key properties of the problems to be solved. By imitating the process of natural selection in nature, the GA starts from a set of candidate solutions of the problem instead of a single one, improves them step by step through biological evolutionary process like crossover, mutation, etc., and gets to the optimal solutions in most occasion. The whole process is operated randomly not deterministic. Thus, the GA is algorithm in performance of population and probability search in optimization and capable of dealing with any kind of objective and constraint functions. In this section, we focus on the design of the GA approach for the q-MST problem.

4.1. Chromosome representation

The q-MST problem is the extension of MST with quadratic cost structure in objective function, so the solution or the chromosome for it should represent a spanning tree. In order to encode a tree, three kinds of encodings have been proposed as follows:

- *edge encoding*;
- *vertex encoding*;
- *edge and vertex encoding.*

These three kinds of tree encodings have both advantages and disadvantages in encoding a tree for GA operation, which has been in detail discussed in [5]. In this paper we focus on the Prüfer number encoding.

One of the classical theorems in graphical enumeration is Cayley's theorem that there are $n^{(n-2)}$ distinct labeled trees on a complete graph with n vertices. Prüfer provided a constructive proof of Cayley's theorem by establishing a one-to-one correspondence between such trees and the set of all strings of $n-2$ digits [12]. This means that we can use only $n-2$ digits permutation to uniquely represent a tree with n vertices where each digit is an integer between 1 and n inclusive. This permutation is usually known as the Prüfer number.

For any tree there are always at least two leaf vertices. By leaf vertex is meant that there is only one edge connected to the vertex. Based on this observation we can easily construct the encoding according to the following procedure:

Procedure: Encoding

Step 1. Let vertex i be the smallest labeled leaf vertex in a labeled tree T.
Step 2. Let j be the first digit in the encoding if vertex j is incident to vertex i. Here we build the encoding by appending digits to the right, and thus the encoding is built and read from left to right.
Step 3. Remove vertex i and the edge from i to j, we have a tree with $n-1$ vertices.
Step 4. Repeat the above steps until one edge is left and produce the Prüfer number or encoding with $n-2$ digits between integer 1 and n inclusive.

It is also possible to generate a unique tree from a Prüfer number via the following procedure:

Procedure: Decoding

Step 1. Let P be the original Prüfer number, $\bar{P}$ be the set of all vertices not included in P, and be designated as eligible for consideration.
Step 2. Let i be the eligible vertex with the smallest label in $\bar{P}$. Let j be the leftmost digit of P. Add the edge from i to j into the tree. Remove i from $\bar{P}$ and j from P. If j does not occur anywhere in the remaining part of P, designate j as eligible and put it into $\bar{P}$. Repeat the process until no digits are left in P.
Step 3. If no digits remain in P, there are exactly two vertices, r and s, still eligible for consideration. Add the edge from r to s into the tree and form a tree with $n-1$ edges.

An example is given to illustrate this encoding. The Prüfer number [2 5 4 5] corresponds to a spanning tree on a six-vertex complete graph represented in Fig. 1. The construction of the Prüfer number is described as follows: locate the leaf vertex having the smallest label. In this case, it is vertex 1. Because

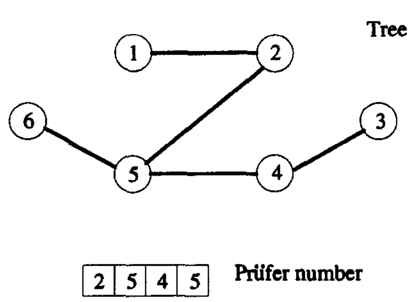

Fig. 1. A tree and its Prüfer number.

vertex 2 (the unique vertex) is incident to vertex 1 in the tree, assign 2 to the first digit in the Prüfer number, then remove vertex 1 and the edge (1,2). Repeat the process on the subtree until the edge (5,6) is left and the Prüfer number of this tree with four digits is finally produced.

Conversely, for the Prüfer number with $P=[2\ 5\ 4\ 5]$, the vertices 1, 3 and 6 are eligible and $\bar{P}=\{1,3,6\}$. Vertex 1 is the eligible vertex with the smallest label. Vertex 2 is the leftmost digit in P. Add edge (1,2) to the tree, remove vertex 1 from $\bar{P}$ for further consideration, and remove the leftmost digit 2 of P leaving $P=[5\ 4\ 5]$. Because vertex 2 is now no longer in the remaining P, it becomes eligible and put into $\bar{P}=\{2,3,6\}$. Vertex 2 is now the eligible vertex with the smallest label and the vertex 5 is the leftmost digit in remaining P. Then add edge (2,5) to the tree, remove vertex 2 from $\bar{P}$ for further consideration, and remove the first digit 5 from P leaving $P=[4\ 5]$. Repeat the process until $P=[5]$ and only vertices 4 and 6 are eligible. Add edge (4,5) to the tree, remove the last digit of P, and designate vertex 4 as not eligible and remove it from $\bar{P}$. Vertex 5 is now eligible as it is no longer in the remaining P and put into $\bar{P}=\{5,6\}$. P is now empty and only vertices 5 and 6 are eligible. Thus add edge (5,6) to the tree and stop. The tree in Fig. 1 is formed.

It is clear that the Prüfer number is more suitable for encoding a spanning tree. This encoding is capable of equally and uniquely representing all possible trees and any initial population or offspring from crossover and mutation operation still keep a tree. As to the q-MST problem, we can directly and easily evaluate the quadratic objective from each Prüfer number or chromosome without any key property. Therefore, the Prüfer number encoding is adopted as the chromosome representation.

4.2. Crossover and mutation

Crossover and mutation are two crucial factors in the biological evolutionary process. In the GA approach they guarantee that the population or chromosomes have a great chance to be evolved to the optimal solution. Because Prüfer number encoding can always represent a tree after any crossover or mutation operations, simply we use the uniform crossover operator which has been shown to be superior to traditional crossover strategies for combinatorial optimization problem [13]. Uniform crossover firstly generates a random crossover mask and then exchanges relative genes between parents according to the mask. A crossover mask is simply a binary string with the same size of chromosome. The operation can be illustrated as Fig. 2. Mutation is performed as random perturbation within the permissive integer from 1 to n as illustrated in Fig. 3.

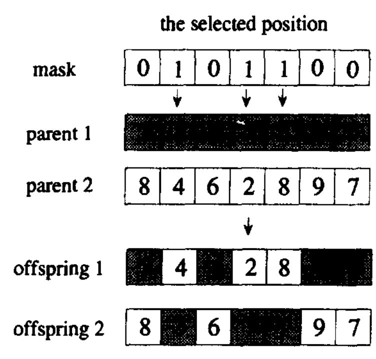

Fig. 2. Illustration of a crossover operation.

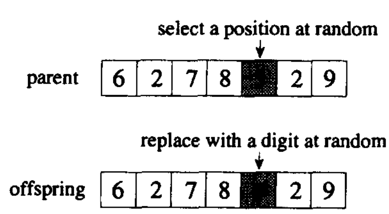

Fig. 3. Illustration of a mutation operation

4.3. Evaluation and selection

The evaluation is to calculate the fitness for each chromosome and the selection is to select chromosomes for the next generation according to their own fitness values. In the GA approach, the evaluation and selection play a very important role. If the crossover and mutation are regarded as the exploration for the GA to make the chromosomes be of diversity, then the evaluation and selection can be regarded as the exploitation for the GA to converge to the optimal or near-optimal solution. In our GA approach for the q-MST problem, the evaluation procedure consists of two steps:

(1) convert a chromosome into a tree, in which includes the transformation of the Prüfer number into a tree in the form of edge set;
(2) calculate the total costs of the tree, in which the fitness for each chromosome can be directly calculated according to the objective in formulation (2).

Let P be a chromosome and $\bar{P}$ is the set of eligible vertices. The evaluation procedure can be illustrated as follows:

Procedure: Evaluation

begin
$T \leftarrow \{\phi\}$;
eval $(T) \leftarrow 0$;
define $\bar{P}$ according to the P;
repeat
select the eligible vertex with the smallest label from $\bar{P}$, say i;
select the leftmost digit from P, say j;
determine the corresponding edge, say e_k;
$T \leftarrow T \cup \{e_k\}$ and $x_k \leftarrow 1$;
remove i from $\bar{P}$
remove j from P;
if j does not occur anywhere in remaining P;
put j into $\bar{P}$;
endif
$k \leftarrow k+1$
until $k \leq n-2$
determine the last edge from remaining two vertices in $\bar{P}$, say e_r;
$T \leftarrow T \cup \{e_r\}$, $x_r \leftarrow 1$;
eval $(T) \leftarrow \sum_{i=1}^{m} \sum_{j=1, j \neq i}^{m} c_{ij} x_i x_j + \sum_{i=1}^{m} w_i x_i$;
end

As to selection procedure, a mixed strategy with $(\mu+\lambda)$-selection [3] and *roulette wheel* selection is used because the $(\mu+\lambda)$-selection can enforce the best chromosomes into the next generation. This evolution strategy has been widely adopted in many combinatorial optimization problems. The mixed strategy in our GA approach selects μ best chromosomes from μ parents and λ offspring. If there are no μ different chromosomes available, the vacant pool of population is filled up with *roulette wheel* selection.

The procedure is shown as follows:

Procedure: Selection

begin
select μ' best different chromosomes;
if $\mu < \mu$
select $\mu - \mu'$ chromosomes by *roulette wheel* selection;
endif
end

4.4. q-MST genetic algorithm

The overall procedure for q-MST problem is outlined as follows:

Procedure: Genetic Algorithm for q-MST

begin
$t \leftarrow 0$;
initialize $P(t)$;
evaluate $P(t)$;
while (**not** termination condition) **do**
begin
recombine $P(t)$ to yield $C(t)$;
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. The termination condition of the algorithm is controlled by a definite number of generation.

5. NUMERICAL EXAMPLE

To test the performance of the GAs approach, we use 17 test problems with size (n) ranging from 6 vertices to 50 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 which are randomly generated and uniformly distributed 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].

The parameters for the GA set as follows: population size $pop\_size=300$; crossover probability $p_c=0.2$; mutation probability $p_m=0.2$; maximum generation $max\_gen=500$; and run by 20 times.

Table 1 gives out the best results from two heuristic algorithms and the GA approach. The "*" in Table 1 indicates the best performers between two heuristic algorithms. According to the results from two heuristic algorithms, there is no apparent comparison between them. However, compared with the results by the GA approach, much better results can be reached. If only the best heuristic results are taken into account, the solution to the q-MST problem can be improved by average 9.6% and at most 12.83% with the GA approach. If we take the run-times of both algorithms into account, the GA approach generally takes more CPU time to get the final solutions than both heuristics. For the largest problem (fifty-vertex graph), the GA approach at least cost 2 h while the heuristic algorithm H2 only takes ten minutes on HP9000 Model 715/100 workstation. However, so far as the effectiveness of the algorithms be concerned, Fig. 4 clearly illustrates the effectiveness of the proposed GA approach over the heuristic algorithms.

Though it is difficult to guarantee if the GA approach has got the optimal solution to the q-MST problem, we still can get some conclusion about the solution by GA approach based on the statistic analysis. As to the solution distribution shown in Table 2, we can be sure, at least in the sense of statistics,

Table 1. The best results from heuristic algorithms and GA approach

Problems size (vertices)	H1	H2	GA	Improved degree
6	344	326*	302	7.36%
7	467*	495	445	4.71%
8	563*	573	547	2.84%
9	720*	766	630	12.50%
10	925	921*	834	9.45%
11	1118*	1241	979	12.43%
12	1209*	1362	1095	9.43%
13	1392	1375*	1248	9.24%
14	1777*	1974	1549	12.83%
15	2078*	2239	1823	12.27%
16	2292*	2517	2010	12.30%
17	2615*	2689	2389	8.64%
18	2888	2812*	2475	11.98%
20	3554*	3749	3233	9.03%
30	8500*	9402	7742	8.92%
50	25182	23487*	22980	2.16%

Note: H1: heuristic algorithm 1; H2: heuristic algorithm 2; GA: genetic algorithm; Improved degree: improved on result by GA to the best heuristic result.
*Best performers between two heuristic algorithms.

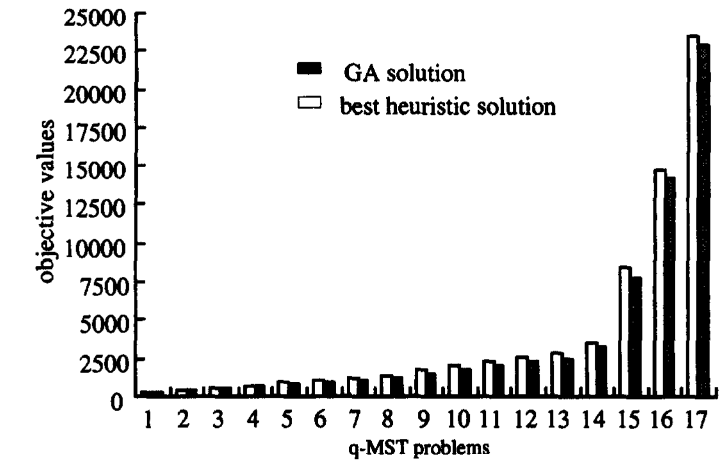

Fig. 4. Comparison between heuristic algorithms and GA.

that the proposed GA approach can get the optimal solution with great probability when the problem scale is not too large. While the problem scale becomes lager, the proposed GA approach can focus on the points near the optimal solution with less standard deviation, from which we can conclude that it has been close to the optimal solution. Therefore, with less relative error and better performance than the heuristic algorithms, the proposed GA approach to the q-MST problem can greatly meet the demands from practical engineering design.

6. CONCLUSION

This paper describes a new formulation in MST problem which was denoted as quadratic minimum spanning tree (q-MST). After introducing two heuristic algorithms to the q-MST problem, we design a new approach to deal with it by using genetic algorithm (GA). This new GA approach gets the optimal or near-optimal solutions through randomly generating a series of chromosomes, which represents a spanning tree in the form of Prüfer number, and being operated in the way of genetic algorithm. The experiments on the q-MST problem show the high effectiveness of the proposed GA approach which can get much better results than the existing heuristic algorithms and also the proposed GA approach can be easily extended to solve the q-MST problem with degree constraints on vertices.

The q-MST problem is a much recently formulated combinatorial optimization problem. Its quadratic objective function with tree-structure solution makes it more difficult to cope with. Further research is both necessary and possible to make further improvement on this problem by other methods. But the research work in this paper shows the great potential power of GA on the q-MST problem and other

Table 2. The solution distribution of the GA approach

Problems Size (vertices)	Min_val	Ave_val	Std_dev	Rel_err	Frequency
6	302	302.0	0.00	0.0000	100%
7	445	445.0	0.00	0.0000	100%
8	547	547.0	0.00	0.0000	100%
9	630	632.0	6.48	0.0098	90%
10	834	835.4	2.91	0.0035	75%
11	979	981.2	4.51	0.0046	80%
12	1095	1100.4	7.71	0.0070	65%
13	1248	1258.8	13.22	0.0105	55%
14	1549	1591.4	20.19	0.0127	55%
15	1823	1850.1	19.83	0.0107	10%
16	2010	2051.8	21.54	0.0105	10%
17	2389	2420.0	21.67	0.0090	10%
18	2475	2562.9	42.17	0.0165	10%
20	3233	3347.7	48.18	0.0144	10%
30	7742	7886.2	82.27	0.0104	10%
40	14275	14516.8	158.30	0.0109	5%
50	22980	23382.6	255.50	0.0109	5%

Note: Min_val: minimal value; Ave_val: average value; Std_dev: standard deviation; Rel_err: ratio of standard deviation to average value; Frequency: percentage of minimal solution got in all 20 trials.

combinatorial optimizations, and also the high practical value to the q-MST problem in the real-world network design.

Acknowledgements—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. Abuali, F. N. Wainwright, R. L. and Schoenefeld, D. A., Determinant factorization: A new encoding scheme for spanning trees applied to the probabilistic minimum spanning tree problem. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, ed. L. J. Eshelman. Morgan Kaufmann Publishers, San Francisco, California, 1995, pp. 470–475.
2. Assad, A. A. and Xu, W., On lower bounds for a class of quadratic 0-1 problem. *Operation Research Letters*, 1985, **4**, 175–180.
3. Bäck T. and Hoffmeister F., Extended selection mechanisms in genetic algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, eds. R. Belew and L. Booker. Morgan Kaufmann Publishers, Los Altos, California, 1991, pp. 92–99.
4. Dijkstra, E. W., A note on two problems in connection with graphs. *Numerische Mathematik*, 1959, **1**, 269–271.
5. Gen, M. and Cheng, R., *Genetic Algorithms and Engineering Design*. John Wiley & Sons, New York, 1997.
6. Graham, R. L. and Hell, P., On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 1985, **7**, 43–57.
7. Kershenbaum, A., *Telecommunications Network Design Algorithms*. McGraw-Hill, Inc., 1993.
8. Kruskal, J. B.Jr., On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the ACM*, 1956, **7**, 1 48–50.
9. Michalewicz, Z., *Genetic Algorithms+Data Structure=Evolution Programs*, 2nd ed. Springer-Verlag, Berlin Heidelberg, 1994.
10. Palmer, C. C. and Kershenbaum, A., An approach to a problem in network design using genetic algorithms. *Networks*, 1995, **26**, 151–163.
11. Prim, R. C., Shortest connection networks and some generalizations. *Bell Systems Technical Journal*, 1957, **36**, 1389–1401.
12. Prüfer, H., Neuer Beweis eines Satzes über Permutationen. *Archives of Mathematical Physics*, 1918, **27**, 742–744.
13. Syswerda, G., Uniform crossover in genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*, ed. J. Schaffer. Morgan Kaufmann Publishers, San Mateo, California, 1989, pp. 2–9.
14. Xu, W., On the quadratic minimum spanning tree problem. In *Proceedings of 1995 Japan–China International Workshops on Information Systems*, eds. M. Gen and W. Xu. Ashikaga, 1995, pp. 141–148.
15. Zhou, G. and Gen, M., An approach to the degree-constrained minimum spanning tree problem using genetic algorithm. *Engineering Design and Automation*, 1997, **3**, 2 153–162.