A GENETIC ALGORITHM TO SOLVE THE MINIMUM-COST PATHS TREE PROBLEM

Ahmed Y. Hamed[1] and M. R. Hassan[2]

[1]College of Applied Studies and Community Services, University of Dammam, KSA.

[2]Computer Science Branch, Mathematics Department, Faculty of Science, Aswan University, Egypt.

ABSTRACT

*One of the important steps in routing is to find a feasible path based on the state information. In order to support real-time multimedia applications, the feasible path that satisfies one or more constraints has to be computed within a very short time. Therefore, the paper presents a genetic algorithm to solve the paths tree problem subject to cost constraints. The objective of the algorithm is to find the set of edges connecting all nodes such that the sum of the edge costs from the source (root) to each node is minimized. I.e. the path from the root to each node must be a minimum cost path connecting them**.** The algorithm has been applied on two sample networks, the first network with eight nodes, and the last one with eleven nodes to illustrate its efficiency.*

KEYWORDS

Computer networks; Minimum-cost paths tree; Genetic algorithms.

1. INTRODUCTION

The shortest paths tree rooted at vertex s is a spanning tree *T* of *G*, such that the path distance from root *v* to any other vertex *u* in T is the shortest path distance from v to u in G,[1]. In the case of single link failure, [2], proposed an algorithm to solve the optimal shortest paths tree. When considering multicast tree, [3], the authors presented an algorithm to find the Shortest Best Path Tree (SBPT). Based on labeling techniques, Ziliaskopoulos et al. in [4], proposed an algorithm to solve the shortest path trees. Also, The shortest paths tree problem has been solved by an efficient modified continued pulse coupled neural network (MCPCNN) model, [5].

Heuristic and approximate algorithms for multi-constrained routing (MCR) are not effective in dynamic network environment for real-time applications when the state information of the network is out of date, [6]. The authors in [6] presented a genetic algorithm to solve the MCR problem subject to transmission delay and transmission success ratio. Younes in [7] proposed a genetic algorithm to determine the k shortest paths with bandwidth constraints from a single source node to multiple destinations nodes. Liu et al. in [8] presented an oriented spanning tree (OST) based genetic algorithm (GA) for solving both the multi-criteria shortest path problem (MSPP) and the multi-criteria constrained shortest path problems (MCSPP). Also, in [9] the genetic algorithm is used to find the low-cost multicasting tree with bandwidth and delay constraints.

The paper presents a genetic algorithm to solve the paths tree problem under cost constraint. The algorithm reads the connection matrix and the cost matrix of a given network. Also, given the source (root) node *s*, then the genetic operations are executed to search the minimum cost paths that construct the minimum cost paths tree rooted at the source node *s*.

DOI : 10.5121/ijcnc.2015.7405

The rest of the paper is organized as follows: Section 2 presents notations. The problem description in section 3. The proposed GA and its components are given in section 4. Section 5 provides the pseudo code of the entire GA. Section 6 shows the illustrative examples. Finally, section 7 presents conclusions.

2. Notations

G	A network graph.
N	The number of nodes in G.
E	The number of edges in G.
e_{ij}	An edge between node i and node j in G.
c_e	The cost of an edge e.
M	The connection matrix of the given network.
CM	The cost matrix of the given network.
np	The number of paths from node *s* to *t*
T_s	The shortest path rooted at node s

3. THE PROBLEM DESCRIPTION

Given a specified vertex *s*. Let $P^i_{(s,t)}$ be a path number *i* from s to *t*. Let $C^i(P_{(s,t)})$ be the cost of the path $P^i_{(s,t)}$, $i = 1,2, \ldots, np$. The path $P^k_{(s,t)}$ has a minimum cost among all the *(s, t)*-paths if:

$$C^k(\mathrm{P}_{(\mathrm{s,t})}) = \min_{\mathrm{i}} \mathrm{C}^{\mathrm{i}}(\mathrm{P}_{(\mathrm{s,t})})$$

Where

$$\mathrm{C}^{\mathrm{i}}(\mathrm{P}_{(\mathrm{s,t})}) = \underset{e\in P_{(s,t)}}{} c_e$$

Therefore, the minimum-cost paths tree T_s is the collection of minimum cost paths from the source (root) node *s* to the destination nodes t_i. I.e.

$$T_s = \underset{k}{} C^k(\mathrm{P}_{(\mathrm{s,t_i})})$$

The presented method in this paper depend on reading both the connection and cost matrices of a given network, and then find the minimum-cost paths tree rooted at the source node.
Consider the following network with five nodes, shown in Figure 1.

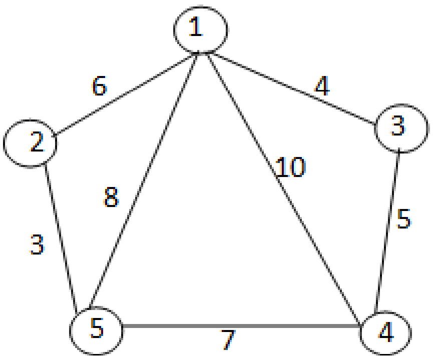

Figure 1. Five nodes network.

The connection matrix (a square matrix of dimension N x N that represents a connection between each node-pairs) of the Figure 1 network is:

$$M = \begin{matrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{matrix}$$

Figure 2. The connection matrix of network in Figure 1

The cost matrix *CM* for the network shown in Figure 1 is in the following form:

$$CM = \begin{matrix} 0 & 6 & 10 & 4 & 8 \\ 6 & 0 & 0 & 0 & 3 \\ 10 & 0 & 0 & 5 & 0 \\ 4 & 0 & 5 & 0 & 7 \\ 8 & 3 & 0 & 7 & 0 \end{matrix}$$

Figure 3. The cost matrix network in Figure 1.

In Figure 4, we show that the minimum-cost paths tree rooted at node 1 with the minimum cost equals to 23.

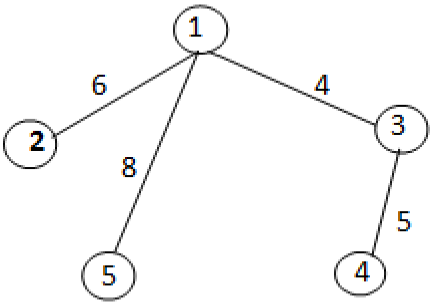

Figure 4. Minimum-cost paths tree rooted at node 1

4. The Proposed Genetic Algorithm (GA)

In the proposed GA, each candidate path is represented by a binary string with length N that can be used as a chromosome. Each element of the chromosome represents a node in the network topology. So, for a network of N nodes, there are N string components in each candidate solution x. Each chromosome must contain at least two none zero elements.

For example if N = 8, the path of Figure 5 is represented as a chromosome as shown in Figure 6. Figure

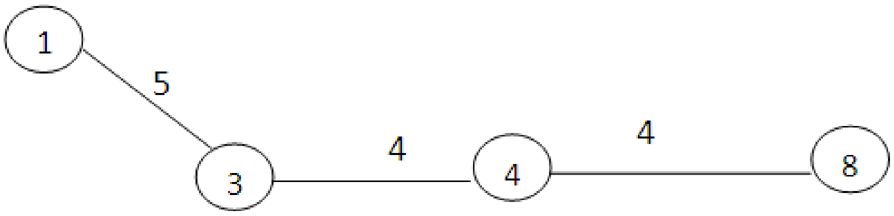

Figure 5. A candidate Path.

1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	1

Figure 6. The chromosome corresponding to the path given in Figure 2.

In the following subsections we give an explanation of different components (operations) of the presented genetic algorithm.

4.1. Initial Population

The generated chromosome in initial population must contain at least two none zero elements to be a real candidate path. The following steps show how to generate *pop_size* chromosomes of the initial population:

1. Randomly generate a chromosome x.
2. Check if x represents a real candidate path, i.e. contains at least two non zero elements.
3. Repeat steps 1 to 2 to generate *pop_size* chromosomes.

4.2. The objective function

The cost of the candidate path is used as objective function to compare the solutions and find the best one. The cost of the candidate path is calculated when it satisfies the following conditions:

- The chromosome must contain at least two none zero elements.
- The chromosome contains a connected candidate path. I.e. each node in the path connects at least one another.

4.3. Genetic Crossover Operation

In the proposed GA, we use the single cut point crossover to breed a new offspring from two parents. The crossover operation will be performed if the crossover ratio (Pc=0.90) is verified. The cut point is randomly selected. Figure 7 shows the crossover operation.

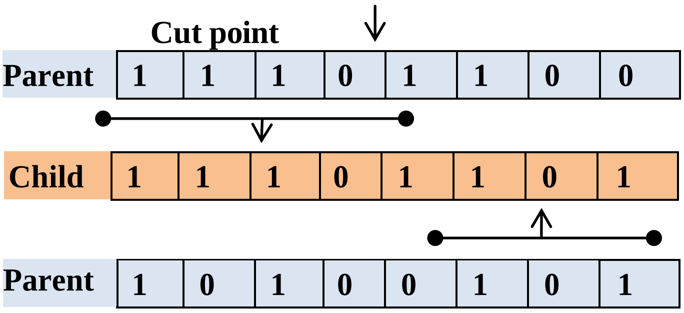

Figure 7. Example of the crossover operation.

4.4. Genetic Mutation Operation

The mutation operation is performed on bit-by-bit basis. In the proposed approach, the mutation operation will be performed if the mutation ratio (Pm) is verified. The Pm in this approach is chosen experimentally to be 0.02. The point to be mutated is selected randomly. The offspring generated by mutation is shown in Figure 8.

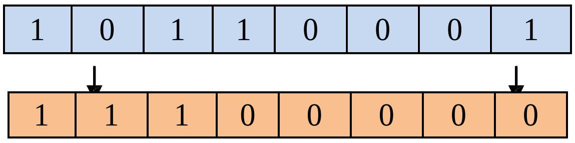

Figure 8. An example of the mutation operation.

5. THE ENTIRE ALGORITHM

The following pseudocode illustrates the use of our different components of the GA algorithm to generate the minimum-cost paths tree of a given network.

Algorithm Find minimum-cost paths tree

```
Input  : Set the parameters: pop_size, max_gen, $P_m$, $P_c$.
Output : Minimum-cost paths tree
  1.  Set j = 2, the destination node.
  2.  Generate the initial population according to the steps in Section 0.
  3.  $gen \leftarrow 1$.
  4.  **While** ($gen <= max\_gen$) **do {**
  5.        **P** ← 1
  6.        **While** ($P <= pop\_size$) **do {**
  7.        Apply Genetic operations to obtain new population
  7.1.  Apply crossover according to $P_c$ parameter ($P_c >=0.90$) as described in section 4.3.
  7.2.  Apply Mutation as shown in section 4.4.
  7.3.  Compute the total cost of the candidate path according to Section 3.
  8.        P ← P+1.
  9.      }
  10. Set gen =gen + 1
```

11. if gen > max_gen then **stop**
12. **}**
13. Save the candidate path for the destination j that has the minimum cost (the shortest path between the root node and the destination node j).
14. Set j = j + 1
15. If j <= N Goto Step 2, otherwise stop the entire algorithm and print out the minimum-cost paths tree.

5. EXPERIMENTAL RESULTS

The proposed algorithm is implemented using Borland C++ Ver. 5.5 and the initial values of the parameters are: population size (pop-size=20), maximum generation (max_gen=50), Pc=0.90, and Pm=0.02. The technique reads both the connection and cost matrices of the given network. Then it generates the shortest paths tree of the network that posses the minimum cost. Two Examples are used to test and validate the proposed technique.

5.1 Eight nodes example

In this section, we illustrate the results of applying the presented GA on an eight nodes network example, as shown in Figure 9. The final output o the GA is shown in Table 1. Figure 10 shows the shortest paths tree rooted at node 1.

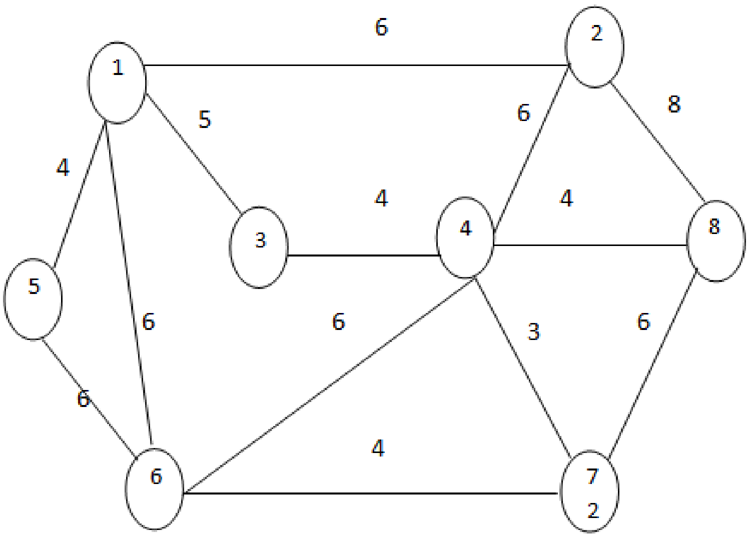

Figure 9. Eight nodes network.

Table 1: The final output of the GA.

The chromosome	The shortest paths set	The cost
(1 1 0 0 0 0 0 0)	{1, 2}	6
(1 0 1 0 0 0 0 0)	{1, 3}	5
(1 0 1 1 0 0 0 0)	{1, 3, 4}	9
(1 0 0 0 1 0 0 0)	{1, 5}	4
(1 0 0 0 0 1 0 0)	{1, 6}	6
(1 0 0 0 0 1 1 0)	{1, 6, 7}	10
(1 0 1 1 0 0 0 1)	{1, 3, 4, 8}	13

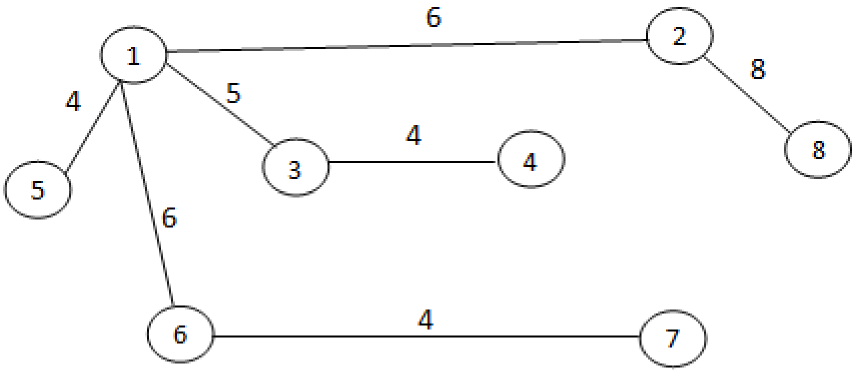

Figure 10: The shortest paths tree

6.2. Eleven nodes example

In this section, the GA is applied on eleven nodes example as shown in Figure 11. The final output of the GA is shown in Table 2. Figure 12 shows the minimum-cost paths tree rooted at node 1.

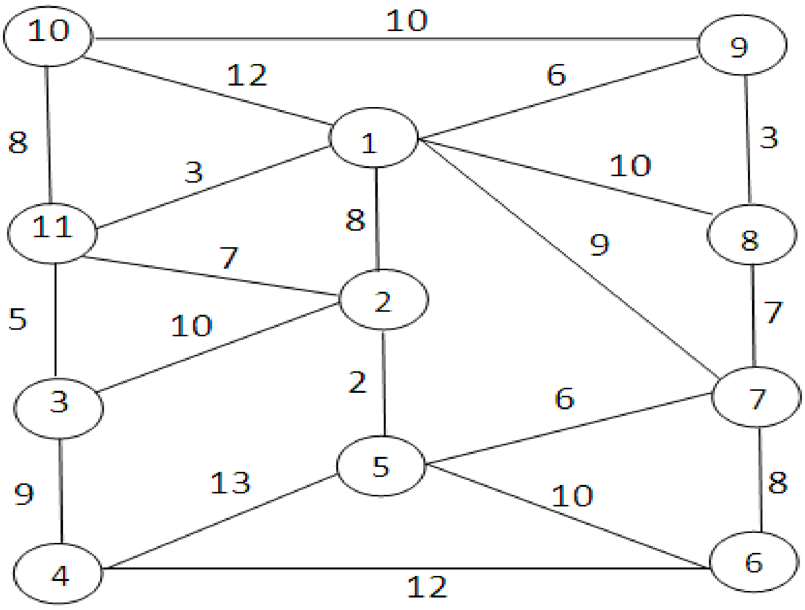

Figure 11: Eleven nodes network.

Table 2. The final output of the GA.

The chromosome	The shortest paths set	The cost
(1 1 0 0 0 0 0 0 0 0 0)	{1, 2}	8
(1 0 1 0 0 0 0 0 0 0 1)	{1, 11, 3}	8
(1 0 1 1 0 0 0 0 0 0 1)	{1, 11, 3, 4}	17
(1 1 0 0 1 0 0 0 0 0 0)	{1, 2, 5}	10
(1 0 0 0 0 1 1 0 0 0 0)	{1, 7, 6}	17
(1 0 0 0 0 1 0 0 0 0 0)	{1, 7}	9
(1 0 0 0 0 0 0 1 1 0 0)	{1, 9, 8}	9
(1 0 0 0 0 0 0 0 1 0 0)	{1, 9}	6
(1 0 0 0 0 0 0 0 0 1 1)	{1, 11, 10}	11
(1 0 0 0 0 0 0 0 0 0 1)	{1, 11}	3

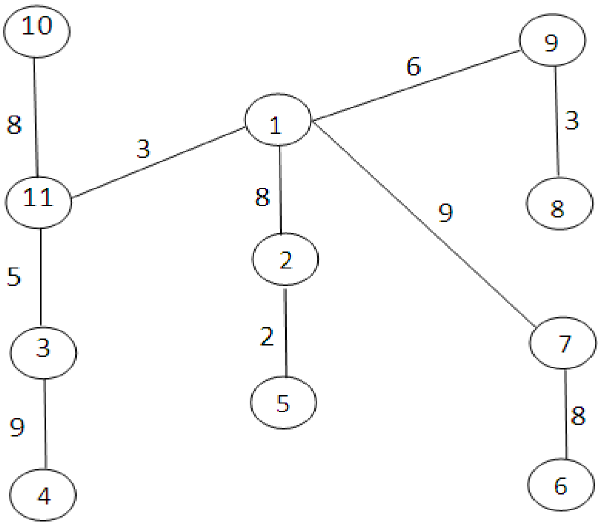

Figure 12: The minimum-cost paths tree rooted at node 1.

6.3. Sixteen nodes example

Also, the GA is applied on sixteen nodes example as shown in Figure 13. The final output of the GA is shown in Table 3. Figure 14 shows the minimum-cost paths tree rooted at node 1.

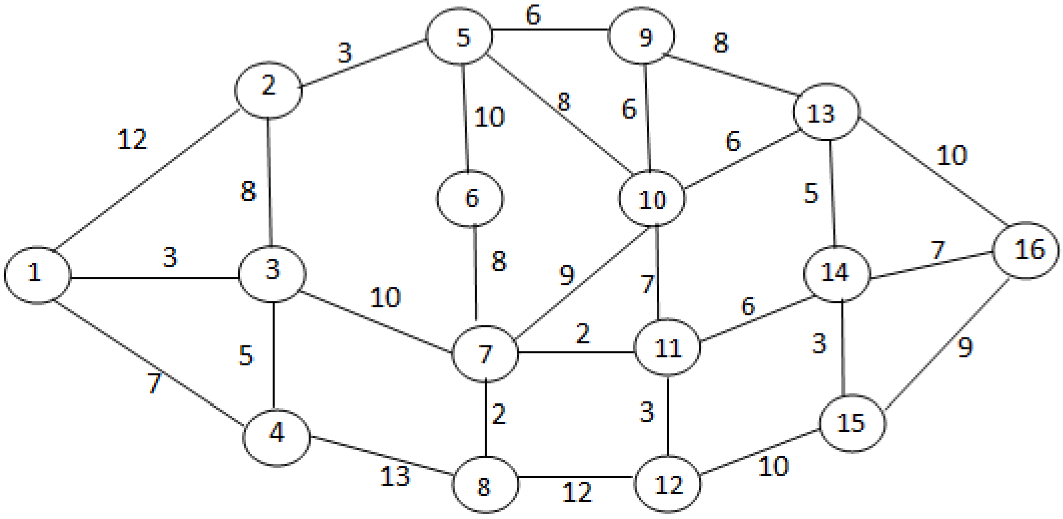

Figure 13: Sixteen nodes network

Table 3. The final output of the GA.

The chromosome	The shortest paths set	The cost
(1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0)	{1, 3, 2}	11
(1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0)	{1, 3}	3
(1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0)	{1, 4}	7
(1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0)	{1, 3, 2, 5}	14
(1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0)	(1, 3, 7, 6}	21
(1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0)	(1, 3, 7}	13
(1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0)	(1, 3,7, 8}	30
(1 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0)	(1, 3, 7, 10, 9}	28
(1 1 1 0 1 0 0 0 0 1 0 0 0 0 0 0)	(1, 3, 7, 10}	22
(1 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0)	(1, 3, 7, 11}	15
(1 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0)	(1, 3, 7, 11, 12}	18
(1 0 1 0 0 0 1 0 0 0 1 1 1 0 0)	(1, 3, 7, 11, 14, 13}	26
(1 0 1 0 0 0 0 0 0 0 1 1 0 1 1 0)	(1, 3, 11, 12, 15, 14}	31
(1 0 1 0 0 0 1 0 0 0 1 0 0 1 1 0)	(1, 3, 7, 11, 14, 15}	24
(1 0 1 0 0 0 1 0 0 0 1 0 0 1 0 1)	(1, 3, 7, 11, 14, 16}	28

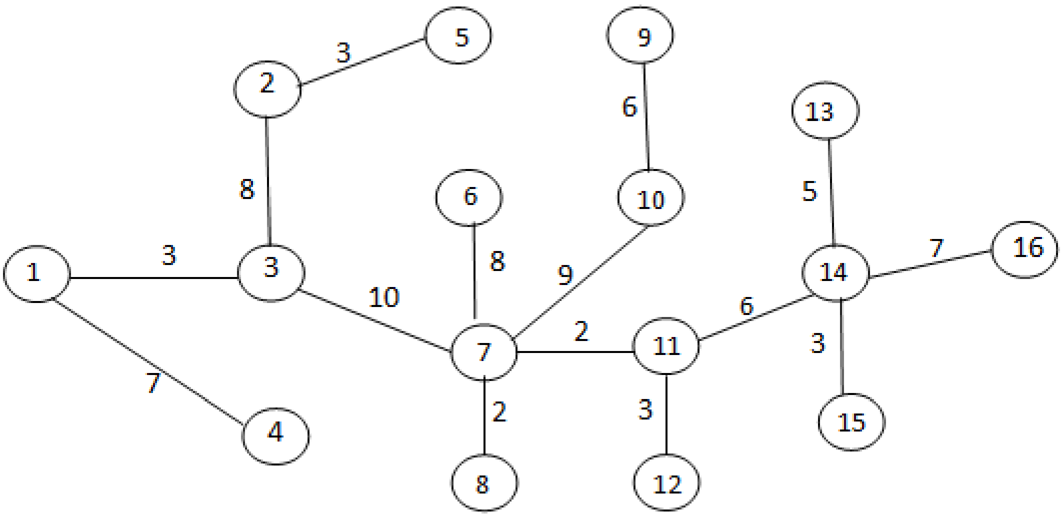

Figure 14: The minimum-cost paths tree rooted at node 1.

7. CONCLUSIONS

The paper addressed the minimum-cost paths tree problem and presented an efficient GA to solve this problem. The algorithm reads both the connection and cost matrices of a given network, then search the minimum-cost paths that construct the minimum-cost paths tree rooted at a given node *s*. The GA has been applied on two examples, the results proved that the efficiency of the proposed GA. For the future work, the GA can be extended to solve multi-constrained paths tree problem.

References

[1] Pierre Hansen and Maolin Zheng, “Shortest shortest path trees of a network”, Discrete Applied Mathematics, Vol. 65, Issues 1–3, March 1996, pp.275–284.

[2] Yueping Li, Zhe Nie and Xiaohong Zhou, “Finding the Optimal Shortest Path Tree with Respect to Single Link Failure Recovery”, Fourth International Conference on Networked Computing and Advanced Information Management, 2008, NCM '08, Vol. 1, pp. 412 – 415.

[3] Hiroshi Fujinoki and Kenneth J. Christensen, “The New Shortest Best Path Tree (SBPT) Algorithm for Dynamic Multicast Trees”, Conference onLocal Computer Networks, 1999. LCN '99. pp. 201-211.

[4] Athanasios K. Ziliaskopoulos, Fotios D. Mandanas, and Hani S. Mahmassani,”An extension of labeling techniques for finding shortest path trees”, European Journal of Operational Research, Vol. 198 (2009), pp. 63–72.

[5] Hong Qua, Simon X. Yang, Zhang Yi, and Xiaobin Wanga, “A novel neural network method for shortest path tree computation”, Applied Soft Computing, Vol. 12 (2012), pp. 3246–3259.

[6] Ting Lu and Jie Zhu, “A genetic algorithm for finding a path subject to two constraints”, Applied Soft Computing, Vol. 13, Issue 2, February 2013, pp. 891-898.

[7] A. Younes , “A genetic algorithm for finding the k shortest paths in a network”, Egyptian Informatics Journal, Vol. 11, Issue 2, December 2010, pp. 75-79.

[8 Linzhong Liu, Haibo Mu, Xinfeng Yang, Ruichun He, and Yinzhen LiAn, “oriented spanning tree based genetic algorithm for multi-criteria shortest path problems”, Applied Soft Computing, Vol. 12, Issue 1, January 2012, pp. 506-515.

[9] A. Younes, “Multicast routing with bandwidth and delay constraints based on genetic algorithms”, Egyptian Informatics Journal, Vol. 12, Issue 2, July 2011, pp. 107-114.

AUTHORS

Ahmed Younes Hamed received his PhD degree in Sept. 1996 from South Valley University, Egypt. His research interests include Artificial Intelligence and genetic algorithms; specifically in the area of computer networks. Recently, he has started conducting a research in the area of Image Processing. Currently, he works as an Associate Professor in University of Dammam, KSA. Younes always publishes the outcome of his research in international journals and conferences.

Moatamad Hassan holds a PhD of Computer Science in June 2006 from Aswan University, Faculty of Science, Aswan, Egypt. He is currently an assistant professor at the Department of Mathematics, Computer Science Branch, Faculty of Science, Aswan University, Aswan, Egypt. His work deals with QoS, Reliability, and Computer Network Design problems.