Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2020.012663
Article

Tech Science Press

A Genetic Algorithm Optimization for Multi-Objective Multicast Routing

Ahmed Y. Hamed[1], Monagi H. Alkinani[2] and M. R. Hassan[3,*]

[1]Faculty of Computers and Information, Department of Computer Science, Sohag University, Sohag, 82524, Egypt
[2]Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah, 21959, Saudi Arabia
[3]Faculty of Science, Department of Mathematics and Computer Science, Aswan University, Aswan, 81528, Egypt
*Corresponding Author: M. R. Hassan. Email: mr.hassan@sci.aswu.edu.eg
Received: 08 July 2020; Accepted: 12 August 2020

Abstract: Many applications require to send information from a source node to multiple destinations nodes. To support these applications, the paper presents a multi-objective based genetic algorithm, which is used in the construction of the multicast tree for data transmission in a computer network. The proposed algorithm simultaneously optimizes total weights (cost, delay, and hop) of the multicast tree. Experimental results prove that the proposed algorithm outperforms a recently published Multi-objective Multicast Algorithm specially designed for solving the multicast routing problem. Also, the proposed approach has been applied to ten-node and twenty-node network to illustrate its efficiency. In addition, the execution time is reported for each studied case and the obtained results are compared with the results obtained by the previously based ant colony algorithm presented recently to solve the same problem. Finality, summing up the three objectives (cost, delay, and hop) to be one objective called the weight of the tree to speed up the searching process by using the proposed algorithm to find the best solutions.

Keywords: Multimedia communication; multicast routing; genetic algorithm; cost; delay; hop

Nomenclature

G: A network graph.
N: The number of vertices in G.
E: The number of edges in G.
e_{ij}: A link between node i and node j in G.
D(e): The delay of a link e.
C(e): The cost of a link e.
H(e): The hop of link e.

1 Introduction

The multicast routing problem is a well-known problem in combinatorial optimization. It is defined as finding the route between two nodes in the weighted graph where that path is the shortest, and shortest means the path with a minimum summation of weights, where an edge between any two nodes always has a certain weight. The problem is to find accordingly the shorter path between a source and a destination in computer networks.

Gen et al. [1] considered the problem of searching the shortest paths with two conflicting objectives of minimizing cost and maximizing flow, as a bicriteria network design problem. They proposed a multi-objective hybrid genetic algorithm (GA) to solve it.

Granat et al. [2], presented an interactive method to analyze the multicriteria shortest path problem by the reference point approach. The multi-objective problem was converted into a parametric single-objective problem. The algorithm succeeded to find the Pareto-optimal shortest path according to the specified preferences.

There are many applications such as multimedia conferencing, distant learning, and video on demand to encourage the network service provider to adapt their network to support additional multicast traffic. The multicast routing problem is the problem of searching a multicast tree that spans all vertices in a communication network [3]. Searching low-cost multicast tree or low delay multicast tree are discussed in [3–5].

To serve the penalty number of users and satisfy quality-of-service (QoS) in real-time applications, this problem is taken into consideration as NP-Complete [6]. Many optimization algorithms based on GA have been proposed to solve the QoS multicast routing (QMR) problem (with different types of QoS constraints) [6–10].

Authors in [11–13] discussed and solved the QoS with multiple constraints like bandwidth, delay, and packet loss rate. An ant colony based heuristic presented by Chu et al. [14] to search minimum cost multicast tree within the case of considering QoS metrics, like bandwidth, delay, delay jitter, and packet loss rate. While, Huang et al. [15], discussed low-cost multicast tree problem subject to delay constraints and ASDLMA (Ant system for delay-constrained low-cost multicast routing algorithm) was constructed to solve it.

It is known that GA is one of the heuristic algorithms that can solve many problems, network design problems [16], real road network problems [17], and unicast routing [18]. Also, GAs used to solve the multicast routing problem [19,20]. In addition, there is a constrained QoS problem [21–27] and [4].

In the case of considering more than one constraint like the cost of the tree, hop count, bandwidth utilization, the problem is considered as a Multi-Objective Problem (MOP) [28].

Ant colony optimization (ACO) is a meta-heuristic approach that has been applied to QoS multicast routing problems [29,30].

Younes et al. [29] presented an AC based algorithm to search a multicast tree with low-cost, minimum delay, and a minimum number of hops. The problem is treated as a multi-objective multicast tree problem.

In this paper, an algorithm based on GA is proposed to solve the multi-objective multicast tree problem. The experimental results prove that the solutions found by the proposed GA are better than those obtained by using AC presented by Younes et al. [30].

The rest of the paper is organized as follows: Section 2 presents the problem description and formulation. Sections 3 describe GA components. The entire GA algorithm is given in Section 4. Studied cases are presented in Section 5. Section 6 gives the conclusion.

2 Problem Description and Formulation

Let $G = (N, E)$ is a weighted directed graph with N vertices and E edges represents a network with $|N|$ nodes and $|E|$ links. The multicast tree from the source node n_0 to the set of destination nodes $U = \{u_1, u_2, \ldots, u_m\}$ denotes a set of destination nodes. Let $X = \{n_0, u_1, u_2, \ldots, , u_m\} \in N$ be a set of from source to destination nodes of the multicast tree. Multicast tree $T = (N_T, E_T)$, where $N_T \subseteq N$ and $E_T \subseteq E$, there exists the path $P_T(n_0, d)$ from source node n_0 to each destination node $d \in U$ in T. Three non-negative real value functions are associated with each link e ($e \in E$): $C(e)$, $D(e)$, and $H(e)$. The link cost function, $C(e)$, may be either monetary cost or any measure of resource utilization. The link delay functions, $D(e)$, define the criteria. The link hop is the number of hops, $H(e) = 1$.

The cost of the path P_T *is* defined as the sum of the cost of all links in that path and can be given by

$$C(P_T) = \sum_{e \in P_T} C(e) \tag{1}$$

The total cost of the tree T is defined as the sum of the cost of all links in that tree and can be given by

$$C(T) = \sum\nolimits_{e \in E_T} C(e) \tag{2}$$

The total delay of the path $P_T(n_0, d)$ is simply the sum of the delay of all links along with $P_T(n_0, d)$:

$$D(P_T) = \sum\nolimits_{e \in P_{T(n_0, d)}} D(e), \quad d \in U \tag{3}$$

The delay of multicast tree T is the maximum value of delay in the path from source node n_0 *to* each destination node $d \in U$.

$$D(T) = \max\left(\sum_{e \in P_{T(n_0, d)}} D(P_T), \quad d \in U\right) \tag{4}$$

The hop of the path P_T is defined as the sum of the hop of all links in that path and can be given by

$$H(P_T) = \sum_{e \in P_T} H(e) \tag{5}$$

The hop of multicast tree is defined as the sum of the hop of all links in that tree and can be given by:

$$H(T) = \sum_{e \in T} H(e) \tag{6}$$

The vector $SW(P_T)$ of the path P_T consists of the vector sum of the vectors corresponding to arcs.

$$SW(P_T) = C(P_T) + D(P_T) + H(P_T); \tag{7}$$

The objective of the presented problem is to find a multicast routing tree (T) such that minimizes the cost $C(T)$, the delay $D(T)$, and the hop $H(T)$. The problem can be formulated as follows:

$$\text{Minimize } W(T) = \sum_{e \in E_T} (C(T) + D(T) + H(T)) \tag{8}$$

where $W(T)$ is the weight of a multicast routing tree (T). The cost $C(T)$, the delay $D(T)$, and the hop $H(T)$ are defined as follows:

$$C(T) = \sum_{e \in E_T} C(e) \tag{9}$$

$$D(T) = \max\left(\sum_{e \in P_T} D(P_T)\right) \tag{10}$$

$$H(T) = \sum_{e \in T} H(e) \tag{11}$$

3 The Proposed GA

If the given network has N nodes, then the candidate path is represented by a chromosome x of N fields, each field represents a node in the network. At least two fields have none zero values to consider the candidate path to be a real path (we called here the reality condition).

3.1 Initial Population

The generated chromosome in the 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. A chromosome x is generated randomly.
2. Check the reality condition for x.
3. Repeat steps 1 and 2 to generate *pop_size* chromosomes.

3.2 The Objective Function

The objective function (The fitness) is the weight of a multicast routing tree $W(T)$ if the candidate path satisfies the following conditions:

- The reality condition.
- The candidate path is connected. i.e., each node within that path connects at least one another.

3.3 Crossover Operation

In our GA, we adopt the single cut point crossover to obtain a new offspring from two parents that are randomly selected based on *Pc* (*Pc* = 0.90).

3.4 Mutation Operation

The uniform mutation is used here based on *Pm (Pm = 0.02)*. The mutated bit is selected randomly to change its value.

4 The Entire Algorithm

The following steps show how the presented GA solves the multi-objective multicast routing tree problem of a given network.

An algorithm to find the minimum multi-objective multicast routing tree

Input: N, S, U, *pop_size, max_gen,* P_m, P_c.

Output: The minimum multi-objective multicast routing tree

1. *T* = *1*
2. While (T<=10) **do** {
3. For all destination nodes (j = 2 to m) {
4. Generate the initial population according to the steps in Section 3.1.
5. *gen* = 1.
6. **While** (*gen* < = *max_gen)* **do**
7. **{**
8. **P =** 1
9. **While** (*P* <= *pop_size*) **do**
10. **{**
11. Apply crossover and Mutation as described in Sections 3.3 and 3.4 respectively.
12. Compute *SW(Pj)* of the candidate path according to Eq. (7).
13. *P* ← *P* + *1*.
14. **}**
15. *gen* = *gen* + *1*
16. If gen > max_gen then **Stop**
17. **}**
18. Store the minimum S*W(Pj) for* the destination node j.
19. }
20. Compute C(T), D(T), and H(T) according to Eqs. (9)–(11) respectively.
21. Compute *W(T)* according to Eq. (8).
22. Save the candidate tree T and *W(T)* for the set of destinations *U*
23. *T* = *T* + *1*
24. }
25. Print out the minimum W(T) (for T = 1: 10) multi-objective multicasting routing tree

5 Studied Cases

The presented GA is implemented using Borland C++ Ver. 5.5, where *pop_size, max_gen,* P_m, and P_c equals to 25, 50, 0.95 and 0.02 respectively. Two networks with 10 and 20 nodes are studied to show the efficiency of the proposed GA. Also, the results are compared with the AC algorithm presented in Younes et al. [30].

5.1 Ten-Node Network

We applied our GA to the network with 10 nodes. Note that the connection matrix and the links' weight are obtained from Younes et al. [30]. Assuming that $n_0 = 1$ and $U = \{5, 7, 9\}$, Tab. 1 shows the value of $W(T)$ for each candidate T. In addition, the execution time (in seconds) required obtaining each T. The minimum value for $W(T)$ is 32 for tree no. 2. The cost, delay, and hop of that tree equals 21, 7, and 4 respectively.

Table 1: The value of W(T) for each T

T No.	The Candidate (T)	W(T)	Average Delay	CPU Time
1	1 –> 5 1 –> 7 1 –> 9	33	5.67	2.15 s
2	1 –> 5 1 –> 5 –> 7 1 –> 9	32	5.00	2.15 s
3	1 –> 5 1 –> 7 1 –> 9	33	5.67	2.16 s
4	1 –> 5 1 –> 7 1 –> 9	33	5.67	2.16 s
5	1 –> 5 1 –> 7 1 –> 7 –> 9	59	10.00	2.18 s
6	1 –> 5 –> 7 1 –> 9	33	5.67	2.18 s
7	1 –> 5 1 –> 5 –> 7 1 –> 9	32	5.00	2.18 s
8	1 –> 5 1 –> 5 –> 7 1 –> 9	32	5.00	2.19 s
9	1 –> 5 1 –> 9 –> 7 1 –> 9	41	6.00	2.19 s
10	1 –> 5 1 –> 7 1 –> 9	33	5.67	2.21 s

The weight, average delay, and execution time for each tree is shown in Figs. 1–3 respectively.

Here, we compare the results of the proposed GA with that obtained by the AC algorithm, Younes et al. [30] as shown in Tab. 2.

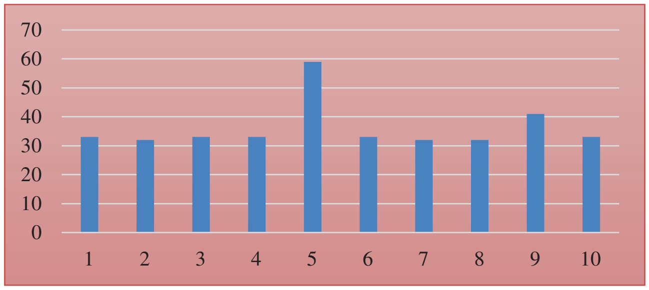

Figure 1: Weight for each tree

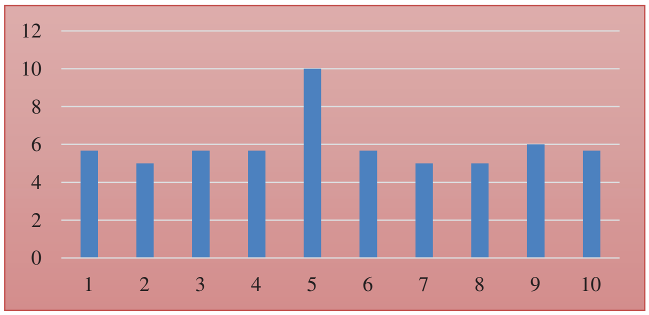

Figure 2: Average delay for each tree

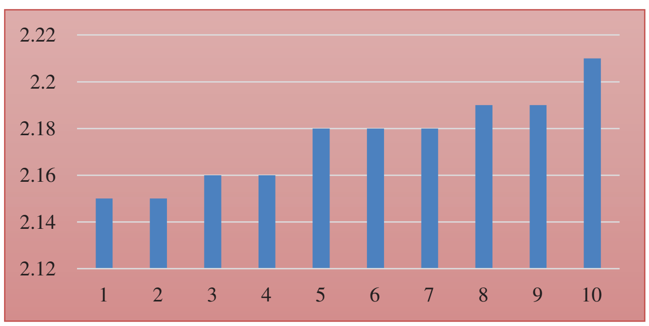

Figure 3: Execution Time for each tree

Table 2: Comparison between the proposed GA and AC presented by Younes et al. [30]

Tree No.	GA Algorithm				AC Algorithm			
	The Candidate Tree (T)	W(T)	Average Delay	CPU Time	The Candidate Tree (T)	W(T)	Average Delay	CPU Time
1	1 –> 5 1 –> 7 1 –> 9	33	5.67	2.15 s	1 –> 2 –> 4 –> 10 –> 3 –> 5 1 –> 6 –> 2 –> 9 –> 7 1 –> 3 –> 7 –> 4 –> 2 –> 9	169	14.33	6.42 s

(Continued)

Table 2 (continued).

Tree No.	GA Algorithm				AC Algorithm			
	The Candidate Tree (T)	W(T)	Average Delay	CPU Time	The Candidate Tree (T)	W(T)	Average Delay	CPU Time
2	1 –> 5 1 –> 5 –> 7 1 –> 9	32	5.00	2.15 s	1 –> 2 –> 9 –> 7 –> 3 –> 5 1 –> 2 –> 9 –> 10 –> 4 –> 7 1 –> 3 –> 7 –> 2 –> 9	156	19.44	6.42 s
3	1 –> 5 1 –> 7 1 –> 9	33	5.67	2.16 s	1 –> 6 –> 10 –> 5 1 –> 6 –> 10 –> 9 –> 7 1 –> 2 –> 7 –> 9	109	27.81	6.43 s
4	1 –> 5 1 –> 7 1 –> 9	33	5.67	2.16 s	1 –> 2 –> 10 –> 8 –> 5 1 –> 3 –> 5 –> 10 –> 4 –> 7 1 –> 2 –> 10 –> 9	123	28.6	6.44 s
5	1 –> 5 1 –> 7 1 –> 7 –> 9	59	10.00	2.18 s	1 –> 3 –> 10 –> 6 –> 5 1 –> 2 –> 10 –> 7 1 –> 3 –> 7 –> 4 –> 10 –> 9	122	29.53	6.45 s
6	1 –> 5 –> 7 1 –> 9	33	5.67	2.18 s	1 –> 2 –> 10 –> 6 –> 5 1 –> 3 –> 10 –> 4 –> 7 1 –> 2 –> 4 –> 10 –> 9	118	27.18	6.45 s
7	1 –> 5 1 –> 5 –> 7 1 –> 9	32	5.00	2.18 s	1 –> 2 –> 6 –> 5 1 –> 2 –> 10 –> 3 –> 7 1 –> 2 –> 10 –> 9	103	27.39	6.46 s
8	1 –> 5 1 –> 5 –> 7 1 –> 9	32	5.00	2.19 s	1 –> 2 –> 7 –> 3 –> 5 1 –> 2 –> 10 –> 7 1 –> 3 –> 7 –> 4 –> 2 –> 9	113	25.13	6.47
9	1 –> 5 1 –> 9 –> 7 1 –> 9	41	6.00	2.19 s	1 –> 2 –> 4 –> 10 –> 6 –> 5 1 –> 2 –> 4 –> 10 –> 7 1 –> 2 –> 6 –> 9	138	23.04	6.47 s
10	1 –> 5 1 –> 7 1 –> 9	33	5.67	2.21 s	1 –> 2 –> 4 –> 10 –> 6 –> 5 1 –> 6 –> 10 –> 4 –> 7 1 –> 6 –> 10 –> 4 –> 2 –> 9	138	26.68	6.48 s

Comparing the results obtained by the proposed GA to those obtained by AC algorithm Younes et al. [30], it is observed that the value minimum W(T) found by the proposed GA is less than that obtained by Younes et al. [30]. Therefore, the proposed GA obtains better optimal solutions. The weight, average delay, and execution time for the best tree found by the proposed genetic algorithm in comparison with Younes, et al. [30] are shown in Fig. 4.

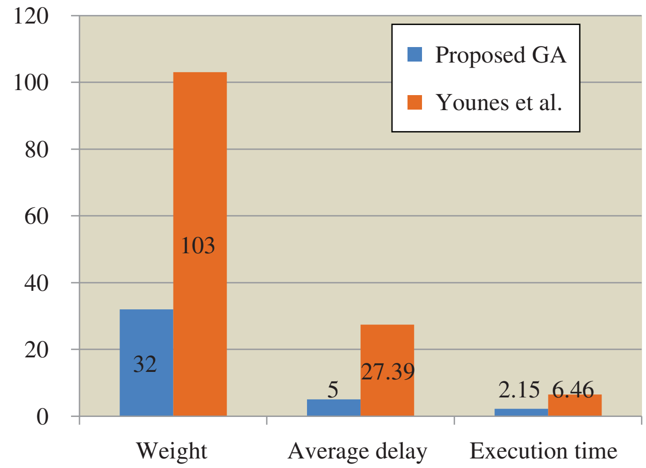

Figure 4: Comparison between the proposed GA and Younes et al. [30]

5.2 Twenty-Node Network

The proposed GA is applied to the twenty-node network, this network along with its information is generated randomly. Also, the connection, cost, hop, and delay matrices are given in Tabs. A1–A4 respectively. Given that n_0 = 1 and $U = \{5, 7, 9, 12, 15, 20\}$, Tab. 3 shows the value of $W(T)$, Average delay, and the execution time (in seconds) for each candidate T. The minimum value for $W(T)$ is 185 for tree no. 6. The cost, delay, and hop of that tree equals 123, 42, and 20 respectively. The weight, average delay, and execution time for each tree are shown in Figs. 5–7 respectively.

Table 3: Candidate route tree from source node 1 to the destination nodes

T No.	The Candidate Tree (T)	W(T)	Average Delay	CPU Time
1	1 –> 5 1 –> 7 1 –> 9 1 –> 5 –> 19 –> 4 –> 14 –> 6 –> 12 1 –> 19 –> 8 –> 2 –> 13 –> 14 –> 15 1 –> 20	198	15.83	14.53 s
2	1 –> 5 1 –> 7 1 –> 9 1 –> 6 –> 14 –> 13 –> 9 –> 20 –> 10 –> 11 –> 12 1 –> 9 –> 12 –> 11 –> 10 –> 13 –> 2 –> 4 –> 20 –> 15 1 –> 20	220	17.83	16.79 s
3	1 –> 5 1 –> 7 1 –> 9 1 –> 20 –> 15 –> 2 –> 3 –> 5 –> 17 –> 6 –> 12 1 –> 20 –> 2 –> 6 –> 13 –> 12 –> 16 –> 15 1 –> 20	226	14.83	19.04 s

(Continued)

Table 3 (continued).

T No.	The Candidate Tree (T)	W(T)	Average Delay	CPU Time
4	1 –> 5 1 –> 7 1 –> 9 1 –> 5 –> 14 –> 13 –> 2 –> 6 –> 20 –> 10 –> 16 –> 12 1 –> 11 –> 4 –> 19 –> 8 –> 17 –> 13 –> 2 –> 20 –> 15 1 –> 20	215	16.33	21.38 s
5	1 –> 5 1 –> 7 1 –> 9 1 –> 11 –> 16 –> 15 –> 20 –> 9 –> 17 –> 2 –> 6 –> 12 1 –> 3 –> 5 –> 14 –> 6 –> 2 –> 17 –> 10 –> 20 –> 15 1 –> 20	212	15.67	23.36 s
6	1 –> 5 1 –> 7 1 –> 9 1 –> 11 –> 4 –> 2 –> 18 –> 15 –> 20 –> 9 –> 12 1 –> 5 –> 14 –> 4 –> 19 –> 16 –> 10 –> 20 –> 15 1 –> 20	185	17.00	25.50 s
7	1 –> 5 1 –> 5 –> 7 1 –> 9 1 –> 2 –> 8 –> 14 –> 5 –> 15 –> 12 1 –> 3 –> 7 –> 19 –> 6 –> 2 –> 17 –> 5 –> 15 1 –> 2 –> 20	237	19.17	27.45 s
8	1 –> 5 1 –> 7 1 –> 9 1 –> 20 –> 7 –> 8 –> 14 –> 5 –> 16 –> 12 1 –> 11 –> 10 –> 13 –> 6 –> 14 –> 20 –> 15 1 –> 20	201	14.50	29.11 s
9	1 –> 5 1 –> 7 1 –> 9 1 –> 14 –> 19 –> 8 –> 7 –> 5 –> 16 –> 11 –> 12 1 –> 5 –> 2 –> 4 –> 18 –> 6 –> 20 –> 15 1 –> 20	207	14.67	30.93 s
10	1 –> 5 1 –> 7 1 –> 9 1 –> 2 –> 4 –> 20 –> 10 –> 13 –> 6 –> 14 –> 12 1 –> 11 –> 10 –> 20 –> 8 –> 7 –> 18 –> 15 1 –> 20	206	15.50	32.90 s

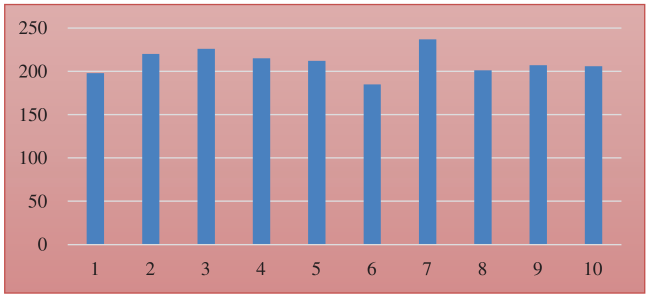

Figure 5: Weight for each tree

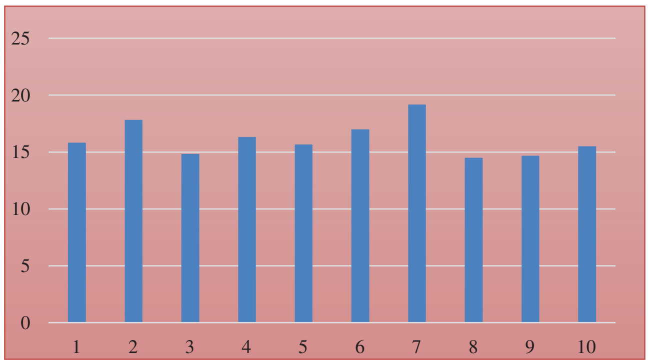

Figure 6: Average delay for each tree

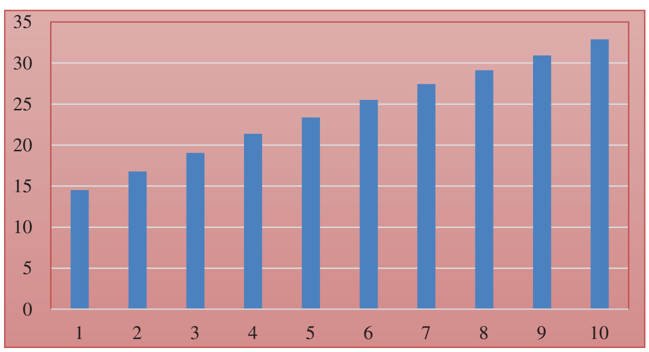

Figure 7: Execution Time for each tree

6 Conclusion

A multi-objective multicast routing problem subject to cost, hop, and delay is presented and formulated as a minimization problem. Furthermore, an approach based on GA is proposed to solve the presented problem. The experimental results illustrated that the proposed GA is efficient in solving this problem and searching the minimum W(T) in a few seconds. In addition, the results obtained by the proposed GA are better than those obtained by AC algorithm presented by Hamed et al. [30].

Acknowledgement: We would like to thank all the parties involved in this research work.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Gen and L. Lin, “Multiobjective hybrid genetic algorithm for bicriteria network design problem,” in *The 8th Asia Pacific Sym. on Intelligent and Evolutionary Systems*, Cairns, Australia, pp. 73–82, 2004.

[2] J. Granat and F. Guerriero, “The interactive analysis of the multicriteria shortest path problem by the reference point method,” *European Journal of Operational Research*, vol. 151, no. 1, pp. 103–111, 2003.

[3] L. H. Sahasrabuddhe and B. Mukherjee, “Multicast routing algorithms and protocols: A tutorial,” *IEEE Network*, vol. 14, no. 1, pp. 90–102, 2000.

[4] B. Wang and J. C. Hou, “Multicast routing and its QoS extension: problems, algorithms, and protocols,” *IEEE Network*, vol. 14, no. 1, pp. 22–36, 2000.

[5] H. F. Salama, D. S. Reeves and Y. Viniotis, “Evaluation of multicast routing algorithms for real-time communication on high-speed networks,” *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 332–345, 1997.

[6] Z. Wang and B. Shi, “Solution to QoS multicast routing problem based on heuristic genetic algorithm,” *Journal of Computer*, vol. 24, no. 1, pp. 55–61, 2001.

[7] A. Younes, “A genetic algorithm for finding the k shortest paths in a network,” *Egyptian Informatics Journal*, vol. 10, no. 2, pp. 75–79, 2010.

[8] Z. Y. Wang, B. X. Wang and E. Zhao, “Bandwidth-delay-constrained least-cost multicast routing based on heuristic genetic algorithm,” *Computer Communications*, vol. 24, no. 7–8, pp. 685–692, 2001.

[9] X. Zhou, C. Chen and G. Zhu, “A genetic algorithm for multicasting routing problem,” *Int. Conf. on Communication Technology*, vol. 2, pp. 1248–1253, 2000.

[10] Z. Wang and J. Crowcroft, “Quality of service for supporting multimedia applications,” *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, 1996.

[11] L. Guo and I. Matta, “QDMR: An efficient QoS dependent multicast routing algorithm,” in *Proc. of the Fifth IEEE Real-Time Technology and Applications Sym.*, Vancouver, BC, Canada, pp. 213–222, 1999.

[12] N. Chen, L. Liand and D. Wushi, “Multicast routing algorithm of multiple QoS based on widest-bandwidth,” *Journal of Systems Engineering and Electronics*, vol. 17, no. 3, pp. 642–647, 2006.

[13] C. Chu, J. Gu, X. Hou and Q. Gu, “A heuristic ant algorithm for solving QoS multicast routing problem,” *Proceedings of the Evolutionary Computation*, vol. 2, pp. 1630–1635, 2002.

[14] L. Huang, H. Han and H. Jian, “Multicast routing based on the ant system,” *Applied Mathematical Sciences*, vol. 1, no. 57, pp. 2827–2838, 2007.

[15] C. W. Ahn and R. S. Ramakrishna, “A genetic algorithm for shortest path routing problem and the sizing of population,” *IEEE Transaction on Evolutionary Computation*, vol. 6, no. 6, pp. 566–579, 2002.

[16] H. Chen and S. Baolin, “Multicast routing optimization algorithm with bandwidth and delay constraints based on GA,” *Journal of Communication and Computer*, vol. 2, no. 5, pp. 63–67, 2005.

[17] G. H. Peng, K. T. Dong and J. S. Su, “A random road network model and its effects on topological characteristics of mobile delay-tolerant networks,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2706–2718, 2013.

[18] L. Atzori and A. Raccis, “Network capacity assignment for multicast services using genetic algorithms,” *IEEE Communications Letters*, vol. 8, no. 6, pp. 403–405, 2004.

[19] R. H. Hwang, W. Y. Do and S. C. Yang, “Multicast routing based on genetic algorithms,” *Journal of Information Science and Engineering*, vol. 16, pp. 885–901, 2000.

[20] R. Bhattacharya, P. Venkateswaran, S. K. Sanyal and R. Nandi, “Genetic algorithm based efficient outing scheme for multicast networks,” in *IEEE Int. Conf. on Personal Wireless Communications*, New Delhi, India, pp. 500–504, 2005.

[21] L. Chen, Z. Yang and Z. Xu, “A degree-delay-constrained genetic algorithm for multicast routing tree,” in *The Fourth Int. Conf. on Computer and Information Technology*, IEEE Computer Society, USA, pp. 1033–1038, 2004.

[22] M. Hamdan and M. E. El-Hawary, “Multicast routing with delay and delay variation constraints using genetic algorithm,” *Canadian Conference on Electrical and Computer Engineering*, vol. 4, pp. 2363–2366, 2004.

[23] L. S. Randaccio and L. Atzori, “A genetic algorithms based approach for group multicast routing,” *Journal of Networks*, vol. 1, no. 4, pp. 1–9, 2006.

[24] M. Mahseur, Y. Meraihi, A. Boukra and A. Ramdane-Cherif, “QoS multicast routing based on a hybrid quantum evolutionary algorithm with firefly algorithm,” in *5th Int. Conf. on Electrical Engineering-Boumerdes*, Boumerdes, pp. 1–6, 2017.

[25] X. X. Zhang, C. X. Shi, H. Y. Chen, J. Yang, P. Wang *et al.,* “A hybrid scatter search algorithm for QoSmulticast routing problem,” in *The 30th Chinese Control and Decision Conf.*, Shenyang, pp. 4875–4878, 2018.

[26] X. Li, Y. Tian, G. Ledwich, Y. Mishra and C. Zhou, “Minimizing multicast routing delay in multiple multicast trees with shared links for smart grid,” *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5427–5435, 2019.

[27] A. Van Veldhuizen, “Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations,” PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, OH, 1999.

[28] M. Dorigo and G. Di Caro, “The ant colony optimization meta-heuristic,” in *New Ideas in Optimization*, D. Corne, M. Dorigo, F. Glover (eds.), London, UK: McGraw Hill, pp. 11–32, 1999.

[29] D. N. Kumar and M. J. Reddy, “Ant colony optimization for multi-purpose reservoir operation,” *Water Resources Management*, vol. 20, no. 6, pp. 879–898, 2006.

[30] A. Y. Hamed, M. H. Alkinani and M. R. Hassan, “Ant colony optimization for multi-objective multicast routing,” *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1159–1173, 2020.

Appendix

Table A1: Connection matrix of twenty-node network

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	1	1	0	1	0	1	0	1	1	0	1	0	1	1	0
2	0	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1
3	0	0	0	0	1	1	1	1	1	0	1	1	0	0	1	0	1	1	1	0
4	0	1	0	0	1	1	1	0	1	1	0	0	0	0	1	0	0	1	1	1
5	0	1	1	1	0	1	1	1	1	0	0	1	1	0	0	0	1	1	0	1
6	1	1	1	1	1	0	1	1	0	0	1	0	1	1	1	0	1	0	1	0
7	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	0	0	0	1	1
8	0	0	1	0	1	1	1	0	0	1	0	0	1	0	0	0	1	0	1	0
9	1	1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	0	1
10	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1
11	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1	0	0	0	0	1
12	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1
13	1	0	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	1

(Continued)

Table A1 (continued).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
14	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1
15	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	1	0	1
16	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	0
17	0	0	1	0	1	1	0	1	0	1	0	0	0	1	1	1	0	0	0	0
18	1	0	1	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	0	1
19	1	1	1	1	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	1
20	0	1	0	1	1	0	1	0	1	1	1	1	1	1	1	0	0	1	1	0

Table A2: Cost matrix of twenty-node network

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	16	19	0	1	0	4	0	6	15	0	4	0	7	7	0
2	0	0	0	12	1	14	0	0	8	17	0	0	0	0	0	0	0	0	9	11
3	0	0	0	0	8	20	17	8	5	0	2	2	0	0	3	0	15	8	12	0
4	0	12	0	0	4	9	13	0	16	14	0	0	0	0	20	0	0	1	7	15
5	0	1	8	4	0	15	8	8	3	0	0	13	5	0	0	0	14	1	0	16
6	16	14	20	9	15	0	4	15	0	0	17	0	4	16	2	0	7	0	15	0
7	19	0	17	13	8	4	0	6	10	0	0	0	0	15	0	0	0	0	15	4
8	0	0	8	0	8	15	6	0	0	8	0	0	11	0	0	0	11	0	6	0
9	1	8	5	16	3	0	10	0	0	0	2	0	0	0	0	13	0	17	0	8
10	0	17	0	14	0	0	0	8	0	0	0	0	14	0	0	0	17	0	0	9
11	4	0	2	0	0	17	0	0	2	0	0	2	3	0	7	0	0	0	0	1
12	0	0	2	0	13	0	0	0	0	0	2	0	0	0	0	11	0	1	0	19
13	6	0	0	0	5	4	0	11	0	14	3	0	0	0	0	16	0	7	8	16
14	15	0	0	0	0	16	15	0	0	0	0	0	0	0	0	0	15	0	0	1
15	0	0	3	20	0	2	0	0	0	0	7	0	0	0	0	0	17	6	0	7
16	4	0	0	0	0	0	0	0	13	0	0	11	16	0	0	0	14	0	0	0
17	0	0	15	0	14	7	0	11	0	17	0	0	0	15	17	14	0	0	0	0
18	7	0	8	1	1	0	0	0	17	0	0	1	7	0	6	0	0	0	0	16
19	7	9	12	7	0	15	15	6	0	0	0	0	8	0	0	0	0	0	0	19
20	0	11	0	15	16	0	4	0	8	9	1	19	16	1	7	0	0	16	19	0

Table A3: Hop matrix of twenty-node network

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	1	1	0	1	0	1	0	1	1	0	1	0	1	1	0
2	0	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1
3	0	0	0	0	1	1	1	1	1	0	1	1	0	0	1	0	1	1	1	0
4	0	1	0	0	1	1	1	0	1	1	0	0	0	0	1	0	0	1	1	1
5	0	1	1	1	0	1	1	1	1	0	0	1	1	0	0	0	1	1	0	1
6	1	1	1	1	1	0	1	1	0	0	1	0	1	1	1	0	1	0	1	0
7	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	0	0	0	1	1
8	0	0	1	0	1	1	1	0	0	1	0	0	1	0	0	0	1	0	1	0
9	1	1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	0	1
10	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1
11	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1	0	0	0	0	1
12	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1
13	1	0	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	1
14	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1
15	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	1	0	1
16	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	0
17	0	0	1	0	1	1	0	1	0	1	0	0	0	1	1	1	0	0	0	0
18	1	0	1	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	0	1
19	1	1	1	1	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	1
20	0	1	0	1	1	0	1	0	1	1	1	1	1	1	1	0	0	1	1	0

Table A4: Delay matrix of twenty-node network

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	3	7	0	6	0	5	0	10	1	0	3	0	2	7	0
2	0	0	0	8	8	3	0	0	4	6	0	0	0	0	0	0	0	0	2	6
3	0	0	0	0	7	8	5	1	6	0	9	8	0	0	2	0	9	2	7	0
4	0	8	0	0	5	1	1	0	7	5	0	0	0	0	8	0	0	9	1	3
5	0	8	7	5	0	8	9	4	5	0	0	9	8	0	0	0	4	2	0	4
6	3	3	8	1	8	0	5	9	0	0	4	0	6	10	3	0	2	0	8	0
7	7	0	5	1	9	5	0	1	10	0	0	0	0	5	0	0	0	0	6	6
8	0	0	1	0	4	9	1	0	0	6	0	0	2	0	0	0	7	0	5	0
9	6	4	6	7	5	0	10	0	0	0	8	0	0	0	0	1	0	5	0	1
10	0	6	0	5	0	0	0	6	0	0	0	0	9	0	0	0	10	0	0	5
11	5	0	9	0	0	4	0	0	8	0	0	10	4	0	3	0	0	0	0	5

(Continued)

Table A4 (continued).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
12	0	0	8	0	9	0	0	0	0	0	10	0	0	0	0	8	0	5	0	1
13	10	0	0	0	8	6	0	2	0	9	4	0	0	0	0	3	0	7	2	10
14	1	0	0	0	0	10	5	0	0	0	0	0	0	0	0	0	2	0	0	10
15	0	0	2	8	0	3	0	0	0	0	3	0	0	0	0	0	7	7	0	6
16	3	0	0	0	0	0	0	0	1	0	0	8	3	0	0	0	10	0	0	0
17	0	0	9	0	4	2	0	7	0	10	0	0	0	2	7	10	0	0	0	0
18	2	0	2	9	2	0	0	0	5	0	0	5	7	0	7	0	0	0	0	8
19	7	2	7	1	0	8	6	5	0	0	0	0	2	0	0	0	0	0	0	7
20	0	6	0	3	4	0	6	0	1	5	5	1	10	10	6	0	0	8	7	0