

A genetic algorithm for shortest path with real constraints in computer networks

Fahad A. Alghamdi¹, Ahmed Younes Hamed², Abdullah M. Alghamdi¹, Abderrazak Ben Salah¹,
Tamer Hashem Farag¹, Walaa Hassan¹

¹Management Information System Department, College of Applied Studies and Community Service,
Imam Abdulrahman Bin Faisal University, Dammam, Kingdom of Saudi Arabia

²Computer Science Department, College of Computers and Information, Sohag University, Egypt

Article Info

Article history:

Received May 24, 2021

Revised Jul 27, 2022

Accepted Aug 29, 2022

Keywords:

Combinatorial optimization
genetic algorithms
Computer network
Multi-objective evolutionary
algorithm
Shortest path problem

ABSTRACT

The shortest path problem has many different versions. In this manuscript, we proposed a multi-constrained optimization method to find the shortest path in a computer network. In general, a genetic algorithm is one of the common heuristic algorithms. In this paper, we employed the genetic algorithm to find the solution of the shortest path multi-constrained problem. The proposed algorithm finds the best route for network packets with minimum total cost, delay, and hop count constrained with limited bandwidth. The new algorithm was implemented on four different capacity networks with random network parameters, the results showed that the shortest path under constraints can be found in a reasonable time. The experimental results showed that the algorithm always found the shortest path with minimal constraints.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Tamer Hashem Farag
Management Information system Department, College of Applied Studies and Community Service, Imam
Abdulrahman Bin Faisal University
Dammam 34212, Saudi Arabia
Email: thhanafy@iau.edu.sa

1. INTRODUCTION

The problem of finding the shortest path is a well-known combinatorial optimization problem, where the term shortest is measured by the minimum value of an objective function. It is defined as finding the shortest route between two nodes in a weighted network, where the shortest path is the one with the least sum of weights, given that every edge connecting any two nodes has a weight associated with it, where the shortest path is the one with the least sum of weights, given that every edge connecting any two nodes has a weight associated with it. Many networking problems are usually mapped to the same problem for instance path planning, vehicle routing, and computer network routing [1], [2]. Furthermore, the requirement for the shortest route is essential for the resolution of other connectivity problems [3]. In this century and last century, many algorithms have been tackled several versions of that problem, for instance, the Bellman's dynamic programming algorithm, Dijkstra algorithm, and the Bellman-Ford algorithm [4]. In computer networks, the shortest path problem is to determine which efficient route should packets take to reach their destinations.

Genetic algorithms are indeed good way to find the optimal solutions for several network problems. In our group's previous works, Hamed [5] considered the problem of multicast and that is finding the k shortest paths and constrained his problem with the required bandwidth and that is done by using the genetic algorithm (GA). Younes [6] attempted to figure out the best way for message multicasting by using a

multicasting tree model while keeping bandwidth and delay limitations in consideration, while Lu and Zhu [7] used an evolutionary algorithm to solve the multi-constrained routing (MCR) issue by restricted transmission latency and transmission success ratio.

In this paper, it can be concluded that GA is a common and recommended approach from our point of view to tackle the shortest path with multi-objectives and under conditioned constraints in computer networks. As mentioned before, finding an optimal solution in the network problems with a heuristic and a stochastic algorithm is very common, like in [3]. While the shortest path problem that has been studied in many papers uses many heuristic algorithms. For example, Xi *et al.* [8] used a dynamic direction restricted algorithm based on the Dijkstra algorithm to find the shortest path in traffic networks.

The consequence sections are organized: Section 2 explores related research and covers previous relevant algorithms, as well as identifying the research gap. The mathematical notations are gathered in Section 3. Section 4 describes the proposed method and the GA parameters utilization. Section 5 gives a pseudocode representation of the proposed algorithm. Section 6 illustrates the findings of the simulation and how they were implemented. Section 7 outlines the conclusion

2. RELATED WORKS

Numerous studies and implementations had focused on finding the shortest path. Monita *et al.* [9] implemented the shortest path first routing protocol to find its impact on the software-defined network (SDN). Torad and Hossamel-din [10] implemented (Dijkstra's algorithm) to find the first shortest path to a hospital with the right blood type to reduce the time needed to find the best hospital. Suryana *et al.* [11] combine four algorithms to find the shortest safe path in emergencies in an indoor environment, the network model was using one sensor data. Tarapata [12] uses a fast and modified version of Dijkstra's algorithm and effective data structures to analyze the complexity of a selected multi-objective shortest path problem. That proposed algorithm finds a solution efficiently to the multi-objective shortest path problem. Zhang and Mouftah [13] presented an algorithm based on a combination of informed search and backward routing. The algorithm finds the shortest path between any two pairs of points in the network with one constraint, the constraint is on the available bandwidth for each link through the path. In [14], the shortest path problem has been studied with multi-objectives. The treated problem considered fuzzy weighted arcs and without other constraints, the conversion speed of the algorithm compared with his previous algorithm [15] and the GA [16]. Li *et al.* [17] build a general model for the shortest path problem with three different types of constraints and with type II dynamics, and then they introduced their algorithm based on a multidimensional expression of the greedy search strategy. Sori *et al.* [18] present a fuzzy inference system to find the shortest path with three objectives and one constraint on the hop count, the multi-objective are the cost, time, and risk, which can be considered as corresponding to cost, delay, and interference. Liao *et al.* [19] used LINGO software to solve a constrained shortest path problem. Their objective is to find the path with the minimum cost where delay time does not more than a predetermined delay time. Hughes *et al.* [20] developed a model for the multiple shortest path problem that did take conflict arcs into account.

Many algorithms use artificial intelligence algorithms to find a solution for the shortest path problem. Marinakis *et al.* [21] considered the constrained shortest arc-disjoint path problem, they used a hybrid local search algorithm which is particle swarm optimization algorithm and variable neighborhood search algorithm, their algorithm finds the shortest path with limited delay and minimum cost. Ahmadi *et al.* [22] used the GA to find the shortest path while the arcs have stochastic conditions. Attiratanasunthron and Fakcharoenpholin [23] studied how long it has taken to execute the ant colony optimization algorithm to solve the shortest path optimization problem with positive weights and without constraints. The ant colony optimization algorithm was also used by Ghoseiri and Nadjari [24] to solve the two-objective shortest path problem, that paper dealt with a large-sized network and finds the solution in an acceptable running time. Ebrahimnejad *et al.* [15] had used a practical swarm optimization algorithm to find the shortest path under the constrain of the existence of mixed fuzzy weighted arcs. Again, Ebrahimnejad *et al.* [25] introduce an artificial bee colony algorithm to find the shortest path in wireless sensor networks. Liu *et al.* [26] introduced a GA to resolve the shortest path problem under multi-criteria and conditioned by constraints. In [27], the shortest path problem in an undirected graph with multi-objectives using an evolutionary algorithm has been introduced. That algorithm employed non-dominated solutions and different selection strategies by using an external set. It analyzed the stochastic convergence and proved the efficiency and the availability of the solution through simulations Alani *et al.* [28] proposed a new hybridizing of A* algorithm and ant colony optimization to solve single-source of the shortest path problem without constraints. The objective of the problem is to find a path with minimum time. For a large-scale (10,000 nodes and 10,000 demand) software network, the routing path in [27] considers finding a visible route for a set of demands with optimal bandwidth. The demanding route becomes clear when it passes two constraints: For each demand route path,

there were two calculations: one based on overall latency, and one based on the total number of hops for each demand route path. Xu *et al.* [29] solve the routing with shortest path problem with multi-constraint but for one objective in large scale SDN, where shortest referred as minimum number of hops.

In Table 1 we summarize key parameters from the previously mentioned research. The research that uses GA is illustrated by shaded rows. Because the weights' values are ambiguous or the weights' values are frequently changed, some networks assign fuzzy weights to the arcs.

The list below is the key contributions and improvements of this proposal.

- In comparison to the several studies mentioned above, we consider a multi-objective version for the shortest path problem. The cost, hop count, and delay are the three main kinds of objectives. The cost would be decided in a different way depending on the real network implantations. In many practical cases the cost could be the consumed electricity and in other ones may be considered as consumed bandwidth.
- In comparison to the regular shortest path problem, we looked at the constrained shortest path problem, where the chosen practical restriction is an upper limit for bandwidth, which again reflects a practical constraint, many studies find the solution for the shortest path problem without considering real network limitations.
- An optimization problem with three objectives and limited bandwidth is being handled effectively and efficiently using a novel genetic approach.
- The introduced GA finds the optimal solution in a reasonable computation time, which can be implemented in real routing applications.
- The algorithm results were tested on completely random networks.

Table 1. The key parameter of previous work

Start node	# of Objectives	Objectives	Constraint or not	# of Constraints	Constraints	year
[12]	one	hop count	no	0	none	2006
[27]	muti (k)	multi	no	0	none	2007
[13]	one	hop count	yes	1	bandwidth	2006
[26]	multi	multi	no	0	none	2012
[14]	one	time	no	0	none	2011
[5]	one	hop count	yes	1	bandwidth	2010
[7]	one	hop count	yes	2	TD, TSR	2013
[23]	one	cost	no	0	none	2008
[24]	two	2 types of cost	no	0	none	2010
[15]	one	cost	no	0	fuzzy arcs	2015
[25]	one	cost	no	0	fuzzy arcs	2016
[16]	one	cost	no	0	fuzzy arcs	2013
[21]	one	cost	yes	1	delay	2017
[22]	one	cost	no	0	stochastic weight	2018
[17]	one	cost	yes	3	3 types	2018
[18]	three	cost, time, interference	yes	1	hop count	2020
[19]	one	cost	yes	1	delay	2021
[20]	One with interference	cost	no	0	none	2021
[28]	one	time	no	o	Multi-destinations	2020
[29]	one	hop count	yes	3	Hop count, delay, bandwidth	2019
Our proposal	three	cost, hop count, delay	yes	1	bandwidth	-

3. MATHEMATICAL MODEL

Notations: G is network graph, N is set of nodes, E is set of edges, e_{ij} is the edge between a node i and a node j , $C(e)$ is cost of an edge e , $D(e)$ is delay of an edge e , $B(e)$ is the bandwidth of an edge e , M is connection matrix of G , and T_s is the shortest path which rooted at nodes. To formulate our problem, we use weighted graph $G=(N, E)$, where N is the set nodes (in computer networks represent the processing machines), and E indicates the connections links between computers (if the nodes can connect, we present that connection by edge e in set E). Hence, the shortest path problem is to find the path from one node to another node where the cost and delay are minimized, under a limited bandwidth of the path. We construct the set of all possible paths between any two nodes, and $P(s, d) \in T$ is denoted as the path from node s to node d . For any two nodes i, j , $e(i, j)$ is denoted as the edge between those nodes. The cost of any e is denoted as $C(e)$, and $D(e)$ denote the delay on edge e . we denote $H(P)$ as the hop count in path P .

The cost of the path P is represented as the total cost of whole links $e \in P$ in the path and can be given by (1).

$$C(P) = \sum_{e \in P} C(e) \quad (1)$$

We calculate the total delay in path PT by (2).

$$D(P) = \sum_{e \in P} D(e) \quad (2)$$

The hop count can be calculated by (3).

$$H(P) = |P| \quad (3)$$

The vector $SW(P)$ of path P contains the vector sum of objective values of the path.

$$SW(PT) = C(P) + D(P) + H(P) \quad (4)$$

The bandwidth of the path P from a node n to a node d is calculated by finding the least edge bandwidth in the path, and it is calculated by (5).

$$B(PT) = \min_{e \in P} B(e) \quad (5)$$

In general, the present shortest path optimization issue may be seen as a vector optimization problem, and it is presented in (6):

$$\text{Minimization } \sum_{e \in P} SW(P) (= \sum_{e \in P} C(P) + \sum_{e \in P} D(P) + \sum_{e \in P} H(P)) \quad (6)$$

where bandwidth is limited: $B(P) \geq B$, and B is the bandwidth limit.

4. THE PROPOSED METHOD

The proposed algorithm implementation is based on the GA. As a result, the first step might be to define the GA parameters, such as:

- a. The chromosome: In the case of a network with N nodes, we will represent the chromosome as a vector of N elements since the longest path will pass through all nodes. In each chromosome, the first node and last node are present as the sender node and the destination node, as a result, each chromosome will have two or more non-zero elements to represent a pathway.
- b. Path length: The number of non-zero elements is the path length.
- c. Path sequence: The order of non-zero elements in each chromosome presents the route sequence nodes from source to destination.

4.1. Initial population

The primary population generation method produces a random set of chromosomes (set of paths) without repeated elements. i) Randomly generate a chromosome y with the first element as the source node and the last element as the destination node and the remaining elements as a random number from 0 to N . ii) For non-consequence, repeated elements replace the elements between them with zero. Iii) For all repeated consequence elements replace them with zero except one. Iv) Examine if the updated chromosome y is a valid one. In another way, check if it can present a connected path in the network graph. V) All chromosomes (paths) must meet the bandwidth limitation as shown in (5) and (6). Vi) Repeat the previous steps (a-e) till it produces the aimed population size.

4.2. The fitness/evaluation function

We use the vector $SW(P)$ for each chromosome as its fitness function. To choose between any two paths we compare its fitness function value. For any path, we evaluate its vector $SW(P)$ as (6): i) hop count: number of non-zero elements; ii) path delay: summation of the edges delays in the path defined by the chromosome; iii) path cost: calculated by adding the costs of all edges in the path, where the path is retrieved by the chromosome; and iv) the evaluation function value is equal to the summation of the above their items.

4.3. Genetic crossover operation

In general, crossover operations can be done in multiple cuts in the chromosome, but in this proposed GA, we choose to make a single cut point to generate the new offspring. The selected crossover ratio (P_c) is (0.9), that ratio is verified through many running of the program. The cut point is selected randomly at any point of the chromosome. A validation test is performed after the production of the additional chromosomes, just as it was done during steps 2 to 5 in section 4.1 initial population stage.

4.4. Genetic mutation operation

The mutation process is the change in the elements of the chromosome, to make a mutation in the path, in the proposed GA it is implemented by a one-bit change basis (one element). The chosen mutation ratio (P_m) is (0.02), and the turning point is chosen randomly. Again, a validation test is performed after the production of the changed chromosomes, just as it was done during steps 2 to 5 in Section 4.1 initial population stage.

5. THE ALGORITHM

The GA for our problem is described in the following pseudocode. For practical networks, the algorithm produces a shortest-path tree according to the objective function, where the bandwidth constraint is satisfied.

Algorithm find the shortest paths

Input: P_c , P_m , pop_size , B , n (last node) and max_gen

Output: The multi-objective shortest path (Minimize $SW(P_T)$) from S to last node n

1. The primary population will be generated as described in subsection 4.1.
2. $I=1$.
3. $MSWP=1000$;
4. do
5. {
6. $J=1$
7. While ($J \leq pop_size$) do
8. {
9. Crossover chromosomes with P_c parameter ($P_c \geq 0.90$), see subsection 4.3.
10. apply mutation between every two chromosomes, steps are shown in subsection 4.4.
11. Compute $B(PT)$ of the candidate path by using (5).
12. If $B(PT) < B$ discard PT and repeat from step 7
13. Calculate $SW(PT)$ of the chosen path according to (4).
14. Check $SW(PT)$; if $SW(PT) < MSWP$ then $MSWP=SW(PT)$ and save the candidate path
15. $J=J+1$.
16. }
17. Set $I=I+1$
18. Update the shortest path for the destination node N with the minimum $SW(PT)$.
19. }
20. While ($I > max_iterations$)
21. Print out the multi-objective shortest path.

6. EXPERIMENTAL RESULTS

The suggested algorithm is implemented by using a Borland C++ version 5.5 program, and with primary parameters values: population size ($pop_size=20$), $P_c=0.90$, and $P_m=0.02$. Let the indexed node no.1 be the source node and the last indexed node be the destination node. The cost, delay, and bandwidth values for each given network are all randomly generated. Where C_{ij} , D_{ij} , and B_{ij} distributed uniformly in the scope of [1], [20], and $H_{ij}=1$.

We have generated randomly four different networks with a different number of nodes as 50, 100, 150, and 200 nodes for testing and validation of the proposed algorithm. The proposed algorithm generates the shortest path of the given networks that possess the minimum weight (cost, delay, and hop) with limited bandwidth. The proposed algorithm GA was run ten times to compute the minimal sum weights and the execution time for each network is stored to compute the average execution time. For all networks, we proposed that the bandwidth limitation is 10. Without loss of generalization, for all networks we consider the node with the final index to be the destination aim for all nodes (i.e., node no. 50 for 1st network, node 100 for 2nd network, and so on).

Table 2 shows the best results for the first five nodes in the first network. The execution time is 10.37 for the best network and the average for all executions is 9.89. As shown in Table 2, the shortest path for 5 nodes, the shortest path of nodes no. (6, 11, 12, 13, 19, 23, 26) are also matched with what is shown in the table. For instance, the shortest path between node 6 and node 50 was 6-23-50, which can prove that the algorithm finds the best value of the objective function.

The execution time of the best result of the four networks using the proposed GA is shown in Table 3, and Figures 1 and 2. We can deduct from Table 3 that the execution time increases gradually as the network size increases. Figure 1 represents the highest weight for the four networks, while Figure 2 is demonstrating the maximum bandwidth in the four networks. The findings show that the proposed algorithm can solve the shortest path problem in a reasonable amount of time.

Table 2. The results of the proposed algorithm (GA) for 1st network (50 nodes)

Start node	End node	Shortest Path	Cost	Delay	Hop	Bandwidth
1	50	1-11-50	26	12	2	10
2	50	2-13-26-6-23-50	41	19	5	18
3	50	3-25-12-50	31	22	3	15
4	50	4-19-12-50	27	45	3	12
5	50	5-6-23-50	22	14	3	14

Table 3. The execution time of the proposed algorithm (GA) for four networks

Network	Nodes#	No of the best result archived through the ten executions are	Minimum execution time	Maximum execution time	Maximum achieved bandwidth
1	50	9	8.56	11.87	22
2	100	8	9.78	12.45	26
3	150	10	7.45	10.5	32
4	200	8	12.5	16.7	18

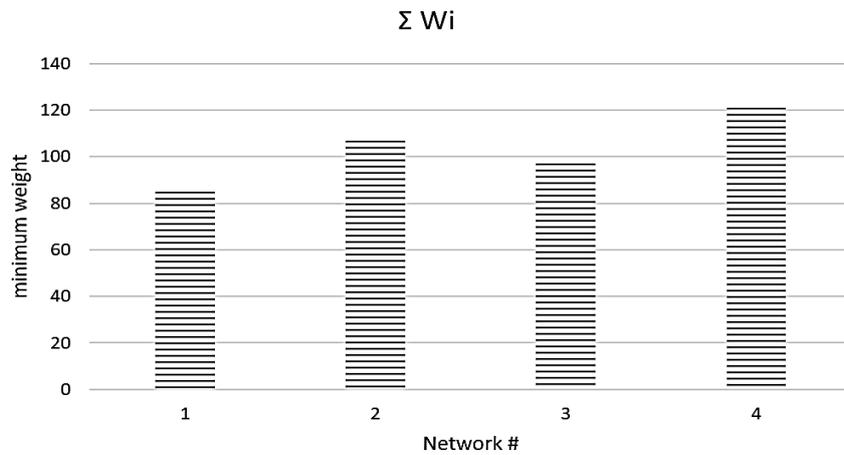


Figure 1. The minimum weight $\sum W$ for the four networks

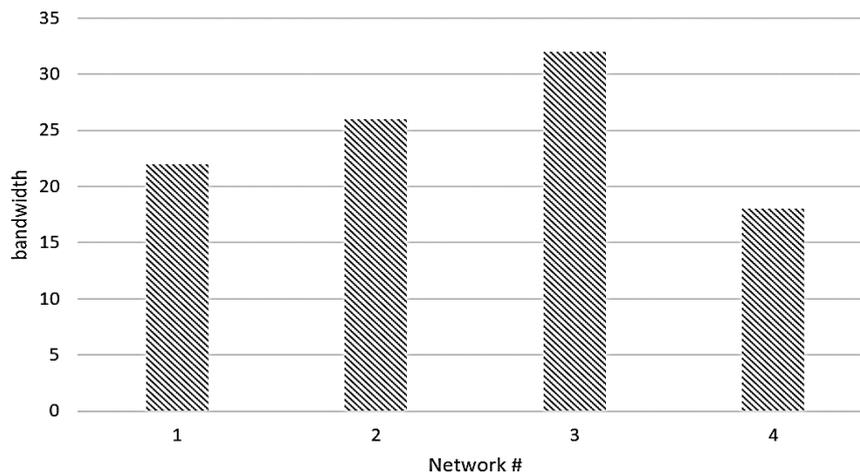


Figure 2. The maximum bandwidth for the four networks

7. CONCLUSION

In this paper, we suggest a new utilization of the GA for solving the shortest path problem with bandwidth constraints. The aim is to find a route path between two nodes that has the fewest hops and meets the practical multi-objectives and practically with bandwidth limitation, as we know in practical life any network has bandwidth limitation. The proposed algorithm is a randomly weighed GA that finds the route

subjected to the minimal sum value of cost, delay, and hop count with a limited bandwidth constraint. In the experimental results, completely random networks (random in links, costs, delays, and edge bandwidths) are used to show the efficiency of the proposed algorithm. The experimental results show that the optimal solution can be found in all cases with efficient execution time.

REFERENCES

- [1] P. V. Mieghem and F. A. Kuipers, "Concepts of exact QoS routing algorithms," *IEEE/ACM Transactions on Networking*, vol. 12, no. 5, pp. 851–864, Oct. 2004, doi: 10.1109/TNET.2004.836112.
- [2] Q. Gao and J. Luo, "A tabu-search-based fast QoS multicast routing optimal algorithm," *Journal of Software*, vol. 15, no. 12, pp. 1877–1884, 2004.
- [3] W. Hassan and T. Farag, "Adaptive allocation algorithm for multi-radio multi-channel wireless mesh networks," *Future Internet*, vol. 12, no. 8, Jul. 2020, doi: 10.3390/fi12080127.
- [4] C. W. Ahn and R. S. Ramakrishna, "A genetic algorithm for shortest path routing problem and the sizing of populations," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 566–579, Dec. 2002, doi: 10.1109/TEVC.2002.804323.
- [5] A. Y. Hamed, "A genetic algorithm for finding the k shortest paths in a network," *Egyptian Informatics Journal*, vol. 11, no. 2, pp. 75–79, Dec. 2010, doi: 10.1016/j.eij.2010.10.004.
- [6] A. Younes, "Multicast routing with bandwidth and delay constraints based on genetic algorithms," *Egyptian Informatics Journal*, vol. 12, no. 2, pp. 107–114, Jul. 2011, doi: 10.1016/j.eij.2011.04.004.
- [7] T. Lu and J. Zhu, "A genetic algorithm for finding a path subject to two constraints," *Applied Soft Computing*, vol. 13, no. 2, pp. 891–898, Feb. 2013, doi: 10.1016/j.asoc.2012.10.018.
- [8] Chen Xi, Fei Qi, and Li Wei, "A new shortest path algorithm based on heuristic strategy," in *2006 6th World Congress on Intelligent Control and Automation*, 2006, pp. 2531–2536, doi: 10.1109/WCICA.2006.1712818.
- [9] V. Monita, I. D. Irawati, and R. Tulloh, "Comparison of routing protocol performance on multimedia services on software defined network," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1612–1619, Aug. 2020, doi: 10.11591/eei.v9i4.2389.
- [10] M. A. Torad and Y. H. Hossamel-din, "Smart ambulance using IoT for blood transfer facilities," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 22, no. 1, pp. 97–103, Apr. 2021, doi: 10.11591/ijeecs.v22.i1.pp97-103.
- [11] N. Suryana, F. S. Utomo, M. F. I. Othman, and M. N. Muhammad, "Voronoi diagram with fuzzy number and sensor data in an indoor navigation for emergency situation," *Telecommunication Computing Electronics and Control (TELKOMNIKA)*, vol. 18, no. 4, pp. 1990–1997, Aug. 2020, doi: 10.12928/telkomnika.v18i4.14905.
- [12] Z. Tarapata, "Selected multicriteria shortest path problems: an analysis of complexity, models and adaptation of standard algorithms," *International Journal of Applied Mathematics and Computer Science*, vol. 17, no. 2, pp. 269–287, Jun. 2007, doi: 10.2478/v10006-007-0023-2.
- [13] B. Zhang and H. T. Mouffah, "Fast bandwidth-constrained shortest path routing algorithm," *IEE Proceedings-Communications*, vol. 153, no. 5, 2006, doi: 10.1049/ip-com:20050509.
- [14] R. K. Ismail and A. Umit, "A genetic algorithm approach for finding the shortest driving time on mobile devices," *Scientific Research and Essays*, vol. 6, no. 2, pp. 394–405, 2011.
- [15] A. Ebrahimnejad, Z. Karimnejad, and H. Alrezaamiri, "Particle swarm optimisation algorithm for solving shortest path problems with mixed fuzzy arc weights," *International Journal of Applied Decision Sciences*, vol. 8, no. 2, 2015, doi: 10.1504/IJADS.2015.069610.
- [16] R. Hassanzadeh, I. Mahdavi, N. Mahdavi-Amiri, and A. Tajdin, "A genetic algorithm for solving fuzzy shortest path problems with mixed fuzzy arc lengths," *Mathematical and Computer Modelling*, vol. 57, no. 1–2, pp. 84–99, Jan. 2013, doi: 10.1016/j.mcm.2011.03.040.
- [17] D. Li, Z. Yu, W. Luo, Y. Hu, X. Che, and L. Yuan, "Optimal route searching with multiple dynamical constraints—a geometric algebra approach," *ISPRS International Journal of Geo-Information*, vol. 7, no. 5, May 2018, doi: 10.3390/ijgi7050172.
- [18] A. Abbaszadeh Sori, A. Ebrahimnejad, and H. Motameni, "The fuzzy inference approach to solve multi-objective constrained shortest path problem," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 4, pp. 4711–4720, Apr. 2020, doi: 10.3233/JIFS-191413.
- [19] X. Liao, J. Wang, and L. Ma, "An algorithmic approach for finding the fuzzy constrained shortest paths in a fuzzy graph," *Complex and Intelligent Systems*, vol. 7, no. 1, pp. 17–27, Feb. 2021, doi: 10.1007/s40747-020-00143-6.
- [20] M. S. Hughes, B. J. Lunday, J. D. Weir, and K. M. Hopkinson, "The multiple shortest path problem with path deconfliction," *European Journal of Operational Research*, vol. 292, no. 3, pp. 818–829, Aug. 2021, doi: 10.1016/j.ejor.2020.11.033.
- [21] Y. Marinakis, A. Migdalas, and A. Sifaleras, "A hybrid particle swarm optimization-variable neighborhood search algorithm for constrained shortest path problems," *European Journal of Operational Research*, vol. 261, no. 3, pp. 819–834, Sep. 2017, doi: 10.1016/j.ejor.2017.03.031.
- [22] E. Ahmadi, G. A. Süer, and F. Al-Ogaili, "Solving stochastic shortest distance path problem by using genetic algorithms," *Procedia Computer Science*, vol. 140, pp. 79–86, 2018, doi: 10.1016/j.procs.2018.10.295.
- [23] N. Attiratanasunthron and J. Fakcharoenphol, "A running time analysis of an ant colony optimization algorithm for shortest paths in directed acyclic graphs," *Information Processing Letters*, vol. 105, no. 3, pp. 88–92, Jan. 2008, doi: 10.1016/j.ipl.2007.08.013.
- [24] K. Ghoseiri and B. Nadjari, "An ant colony optimization algorithm for the bi-objective shortest path problem," *Applied Soft Computing*, vol. 10, no. 4, pp. 1237–1246, Sep. 2010, doi: 10.1016/j.asoc.2009.09.014.
- [25] A. Ebrahimnejad, M. Tavana, and H. Alrezaamiri, "A novel artificial bee colony algorithm for shortest path problems with fuzzy arc weights," *Measurement*, vol. 93, pp. 48–56, Nov. 2016, doi: 10.1016/j.measurement.2016.06.050.
- [26] L. Liu, H. Mu, X. Yang, R. He, and Y. Li, "An oriented spanning tree based genetic algorithm for multi-criteria shortest path problems," *Applied Soft Computing*, vol. 12, no. 1, pp. 506–515, Jan. 2012, doi: 10.1016/j.asoc.2011.08.015.
- [27] F. He, H. Qi, and Q. Fan, "An evolutionary algorithm for the multi-objective shortest path problem," 2007, doi: 10.2991/iske.2007.217.
- [28] S. Alani, A. Baseel, M. Maad Hamdi, and S. Abduljabbar Rashid, "A hybrid technique for single-source shortest path-based on A* algorithm and ant colony optimization," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 2, pp. 356–363, Jun. 2020, doi: 10.11591/ijai.v9.i2.pp356-363.
- [29] C. Xu, L. Tao, H. Wu, D. Ye, and G. Zhang, "Multiple constrained routing algorithms in large-scaled software defined networks," Feb. 2019.

BIOGRAPHIES OF AUTHORS



Fahad A. Alghamdi    is currently an assistant at the Faculty of Applied Studies and Community Service, Department of Management of Information Systems, Imam Abdurrahman Bin Faisal University, Saudi Arabia. He received his BS degree in computer science in 2002 from Imam Abdurrahman Bin Faisal University, Saudi Arabia, his MSc degree in computer science in 2009 from University Wollongong, Australia, and his Ph.D. in Computer Science in 2017 from Aberystwyth University in UK. His research interests include network, applied computer software, cloud computing and online learning. He is currently the Vice Dean of Faculty of Applied Studies and Community Service, Imam Abdurrahman Bin Faisal University, Saudi Arabia. He can be contacted at faghamdi@iau.edu.sa.



Ahmed Younes Hamed    holds a Ph.D. in computer science from South Valley University, Egypt (1996). he is currently a full professor at College of Computers & Information, Sohag University, Egypt. He was as associated professor at Dammam university, an assistant professor at King Faisal university, and assistant professor at South Valley university. His research interest focused on the artificial intelligence. He can be contacted at email: ayhamed@sohag.fci.edu.eg.



Abdullah M. Alghamdi    (BSc'02, MSc'09, Ph.D'17) is an Assistant Professor at Imam Abdulrahman bin Faisal University (IAU), Dammam, Saudi Arabia. Received his Ph.D. (2017) degree in E-learning Information Technologies from the University of Lincoln, UK, MSc (2009) from SJU, USA in Computer information Science 2009, BSc (2002) in Education in Computer from IAU, Dammam, Saudi Arabia. Alghamdi's current research focuses on human-computer-interaction, human identification, deep learning, ICT in education, user experience, and acceptance. He can be contacted at amghamdi@iau.edu.sa.



Abderrazak Ben Salah    is currently an assistant professor at the Faculty of Applied Studies and Community Service, Department of Management of Information Systems, Imam Abdurrahman Bin Faisal University, Dammam KSA. He received his Engineer degree in Software engineering in 1991 from Wroclaw University of Science and Technology, Poland, his MSc degree in Computers Networks in 1992 from Wroclaw University of Science and Technology, Poland, and his Ph.D. in Applied Computer Science in 1995 from Wroclaw University of Economics and Business, Poland. He has published many papers in the Applied Computer Science areas. His research interests include applied computer science, e-commerce security, networks and artificial intelligence. He can be contacted at asalah@iau.edu.sa.



Tamer Hashem Farag    holds a Ph.D. in Computer Science from Okayama University, Japan, 2009. He is currently an assistant professor at the Imam Abdulrahman bin Faisal University, Saudi Arabia. His research interest focused on the combinatorial optimization field, and his publication related to computer networks, image processing, management information technology, technology and sustainability, technology, and entrepreneurship. He can be contacted at thhanafy@iau.edu.sa.



Walaa Hassan    received a B.Sc. in mathematics and computer science in 1999, and M.Sc. in computational science in 2005 from Cairo University, Cairo, Egypt. She received her Ph.D. in September 2010 in network security from Okayama University, Japan. She is currently an assistant professor at Imam Abdulrahman bin Faisal University (IAU), Dammam, Saudi Arabia. Hassan's current research focuses on wireless network security, deep learning, and digital signature. She can be contacted at wahassan@iau.edu.sa.