

Intelligent Routing Algorithm Using Genetic Algorithm (IRAGA)

Nibras Abdullah^{1,2(✉)}, Ola A. Al-wesabi^{2,3}, Mahmoud Baklizi⁴,
and Mohammed M. Kadhum¹

¹ National Advanced IPv6 Centre, Universiti Sains Malaysia (USM),
11800 Gelugor, Penang, Malaysia
neprasf@gmail.com

² Faculty of Computer Science and Engineering, Hodeidah University,
Hodeidah, Yemen

³ School of Computer Sciences, Universiti Sains Malaysia (USM),
11800 Gelugor, Penang, Malaysia

⁴ Department of Computer Information and Network Systems,
The World Islamic Sciences and Education University, Amman, Jordan

Abstract. In this study, Intelligent Routing Algorithm using Genetic Algorithm (IRAGA) is proposed to find the feasible path based on multi-constraint by using GA which presented the paths in chromosomes to help the algorithm to find the efficient path among the available paths. The proposed crossover-refining operation keeps the genetic variety of the chromosomes and also improve the ability of searching which obtains the good result and increases the rate of convergence between the chromosomes. The objective of this research work is to decrease the time of route selection and select the most efficient path on the basis of fitness value. IRAGA offers a better solution in a few iteration compared to other genetic algorithm-based routing protocols, such as genetic load balancing routing, adaptive routing method based on GA with QoS, and multi-constraint QoS Unicast Routing using GA, in terms of the time taken to find the feasible path.

Keywords: GA · Crossover · Routing · Multi-constraints

1 Introduction

A mobile ad hoc network (MANET) consists of a collection of mobile and wireless nodes without the need for a specific fixed infrastructure that make it useful and easy to use anywhere such as: disaster area where MANET is useful in rescue effort, also it is useful in war mission areas to communicate between the soldiers and their staff. There are many Manet-based applications and services, such as Military application, mobile conferencing, and emergency services. These applications, need to control and manage the information among the network's entities where the routing protocol taking place. Routing protocols are an essential component of the communication processes in the network, wherein they are used to route data packets and also to guarantee that an efficient path is established between a pair of nodes in the network. Therefore, data packets can be delivered in a timely manner [1–3].

Dynamic change of the network topology is the common phenomenon in MANET as nodes move arbitrarily and unpredictably. Thus, the connection between the nodes might be changed frequently. Moreover, the transmission range could be changed according to the radius of transmission area, which may lead to loss of the connection between the nodes. The information of the current session of the network topology changes frequently during the time that is considered a significant requirement in a MANET routing algorithm. Therefore, the fast decision, based on which a route provides an efficient performance should be considered [4–8]. Recently, Genetic-based routing Algorithm has been used to find the feasible path.

2 Genetic Algorithm

GA as optimization approach is used in MANET routing algorithms [9–13]. GA works in specific principle procedures, starting from random initial population generation, then chromosomes evaluation according to the fitness function to evaluate each individual. Across evaluation procedure, the acceptable paths which presented in chromosomes, are selected for processing in certain common genetic recombination operations, such as standard crossover and mutation operations to improve the next generation. The recombination operations impact in the decision of routing selection. The standard crossover operation is the exchange of selected genes based on the crossover point through selected individuals' chromosomes which assigned as parents to produced new offspring. Then, the mutation operation induces changes in single individual gene. The unacceptable paths presented in chromosomes will be replaced by a new offspring that may be better than the existing paths. However, these recombination operations do not care for the connectivity between the parts of genes which may generate invalid paths. These processes are repeated until terminal condition is reached, during which an optimum solution is found or the maximum number of iterations is obtained in the early stage [14, 15].

This article present the crossover refining operation instead of the two-standard operation to reduce the consuming time in the recombination stage and avoid the invalid paths.

3 GA-Based-Routing Algorithm

Genetic algorithm has been used in several research works in the routing protocol to find the optimal location for router, and find the feasible path for wireless networks [16–18].

The selection of paths is very complex network optimization problem. This problem can be solved by efficient solutions are much sought out because this kind of solutions could lead to considerable monetary savings and better utilization of the networks. However, several researchers have adopted GA for routing optimization in computer networks, the crossover operation and the mutation operation was not taken into account the topology of the network. Thus, many of invalid chromosomes which present the invalid paths are generated, that greatly decrease the efficiency of GA [18].

In this section selected GA-based routing algorithms are reviewed. Genetic Load Balancing Routing (GLBR), Adaptive Routing method based on GA with QoS (ARGAQ), and Multi-constraint QoS Unicast Routing Using GA (MURUGA).

3.1 GLBR

The GLBR algorithm utilizes genes in determining the code by which the genes have been encoded into chromosomes in the same order as the nodes have formed the path of communication [19]. Accordingly, the individuals' lengths of the population chromosomes are varied. A control condition must be produced to approve any of the new courses that are recognized in the overhead count. The algorithm begins with an irregular initial population and after that the genetic operations, crossover, and mutations, are applied. The crossover operator arbitrarily selects a node as a crossing site to exchange sub-paths among the combination of paths. The new offsprings need to verify and validate which causes the increase in processing overhead. In order to improve the new offsprings, the path mutation operator arbitrarily selects a mutation node that may help to find a path to the destination node by mutating Dijkstra's shortest path selection. These operations of GLBR and its procedures are repeated until available path is found or if the generation range exceeds a certain determined value. The aim of GLBR algorithm is to balance the load in the network. GLBR sending inquiries packet to check the current path if it is loaded or not. Based on the threshold value of the response time to this query from the destination, the genetic algorithm operations will be repeated to recompute the path in case the response time is greater than the threshold value. GLBR algorithm performs better compared with Shortest Path First (SPF) and Routing Information Protocol (RIP) algorithms. On the other hand, GLBR algorithm is operated based on Delay Time as a single constraint, which does not satisfy the QoS routing. The network overload is caused by frequently sending the delay query packets. Moreover, the variation in size of the individuals chromosomes leads to a complicated genetic algorithm operations, especially in crossover operation which can produce path with loops. In addition to this, the algorithm has to validate each path every time when a new path is identified, which results in computation overhead.

3.2 ARG AQ

Routing algorithms on high speed network for multimedia applications should be enhanced which taken in account certain QoS parameters [9]. Thus, ARG AQ is improved into ARG AQ by using two QoS parameters. ARG AQ scheme utilizes two constraints to evaluate the path selection, delay time and transmission success ratio. However, the two QoS parameters are considered as one metric. Delay time denoted the time it takes for packet to move from node to another node. The main aim of this method is to find the optimal path from the source node to the destination node. The ARG AQ uses QoS Routing Search Engine (RSE) structure to find the feasible paths to be used in GA as individual populations. This engine has two search sub-engines: Cache Search Engine (CSE) and Tree Search Engine (TSE). These two

sub-engines are working independently, but during the update of routing information they work together.

When a request for a path is received in RSE, the request will be forwarded in parallel to CSE and TSE. The CSE and TSE responsibility are to find a satisfied route from the source to the destination that meet the QoS requirements. The route information is saved in the cache database. CSE seeks for the route in the database, if the satisfied route is there, it will send the route information to RSE. The route which does not meet the QoS requirements is sent to the genetic population pool. If the route is not found by CSE, it will be looked up by TSE and then, send the route information to RSE. TSE is working based on GA to find the route in its domain.

The route information has to be updated because the network changes dynamically. As mentioned above, CSE and TSE are working together to update the database route information. TSE sends the good route information to the cache database; thus, it can be found faster by CSE. If the route in the cache does not fulfill the QoS requirements, it will be deleted from the database. Moreover, the satisfied routes are given higher priority which makes the next search faster. In addition, when TSE receives the QoS route request from RSE, it transforms the network into a tree network model and makes the source node as the root of the tree [9]. ARG AQ is working on the tree network model as ARG A which reduces the tree junctions and encoding the junctions in genes; thus, every chromosome would present a single path from source to destination. The crossover operation is applied in single point between the pair of individuals, and the mutation is applied on one gene randomly. The fitness function combined the two constraints in single mixed metrics. ARG AQ gives a rough estimation of the counting path for single mixed metrics. Thus, multimedia applications need a QoS route that fulfills a number of constraints.

3.3 MURUGA

MURUGA algorithm was presented in [2]. MURUGA was proposed to locate the shortest path between two nodes (source and destination). This method comprises five modules which increased the computation time. In this algorithm, the full path between the source and destination nodes is presented in the chromosome. The chain of neighboring nodes is illustrated by genes. So, all potential paths have appeared in the chromosome structure. MURUGA is depended on QoS, and it controls genes by utilizing the tree structure to avoid loops.

The network connectivity in [20] was formed via a connectivity matrix.

The MURUGA is working as the following:

- Initial population generation.
- Chromosomes evaluation.
- Ranking the population based on their fitness value.
- Termination condition testing.
- If the termination fail: applied GA on the lower-ranked chromosomes to create new genes, else: repeated the steps until a feasible path is found or the termination condition is met.

MURUGA algorithm avoid looping problem and decreased the research space by utilized the connectivity matrix. So, the available paths that are found by this algorithm satisfy the minimum requirements.

However, MURUGA has selected the path according to the minimum required value for the constraint. So, the selected path does not necessary to be the efficient path. Moreover, MURUGA algorithm does not offer high end-to-end performance.

4 Proposed Method

In this section, a new intelligent approach based on multi-constraint GA is proposed.

4.1 Initialization

A representation of the network connectivity in a symmetric matrix, where 1 presents the connectivity, and 0 presents the non-connectivity. For the generation of an initial population, each chromosome starts with the source node as a first active state. Then, the next state is randomly selected from the next state field and is denoted as an active state. This process is repeated for each chromosome until the destination node is reached.

4.2 Fitness Function Calculation

The Fitness function is responsible to evaluate each chromosome in the population. According to the constraints values for each available path the population is evaluated. Based on the fitness value, the efficiency path is selected. The constraints parameters which considered in this work are cost and delay. The Eq. 1 used for calculation weight for each path

$$w_i(p) = \sum_{(u \rightarrow v) \in p} (w_i(u \rightarrow v) \leq K_i \text{ for all } i = 1, \dots, l). \quad (1)$$

Where: l : total number of constraints; p : path; K_i : maximum constraint value; and w_i : the path weights,

From (1) the fitness function is calculated as the following:

$$F(P) = \text{Max}(w_1(p), w_2(p), w_3(p), \dots, w_l(p)) \quad (2)$$

4.3 New Population Generation

Before generating the new population, the current population should be evaluated using the fitness function given in Eq. (2). Populations will then be ranked to eliminate the worst case, which will be replaced by new generated chromosomes. The worst cases comprise the chromosomes that fall outside of the limited constraints values. Thus, the

best chromosomes will have the best values and are thus the most efficient path constraint values.

4.4 Crossover Refining Operation

Crossover refining is applied on pair of selected chromosomes by exchanging the genes values at the crossover point onward to generate new chromosomes. However, the crossover point is selected among the top ranked chromosome. After that, the process continues to find the nearest partner chromosome that is based on the contents of the crossover point to the destination node which guarantee the connectivity between the parts exchanged. The new generated paths are replaced with the worst ones. The crossover-refining is applied according to the crossover point which calculated by Eq. (3) based on the chromosome length.

$$Crossover\text{-}point = \text{floor_off} (Chromosome_length/2) \quad (3)$$

The steps of crossover refining operation as follows:

- Rank the available paths for the current generation based on the fitness value.
- Select the best path among the available paths according to the fitness value, and it is considered as a main factor in the combination processing of a pair of chromosomes.
- Determine the crossover point in the selected chromosome using Eq. (3). Based on the content value of the crossover point, the chromosomes are exchanged the genes to generate valid new chromosomes.
- As the content of crossover point is important to confirmed the connection between the exchanged parts of the chromosomes, if the selected point not found in the current chromosome, have to find in the following ranked chromosomes.
- Apply the recombination between the selected chromosomes by exchanging the genes parts between them according to the content of the crossover point which present the node ID in the available path.
- Replace the new combined chromosomes with the worst fitness chromosomes.
- Move to the fitness evaluation, which is followed by condition termination.

5 Simulation and Analysis

The simulation scenarios were performed on different network topologies of 50 nodes. Based on the random nature of the network, the parameters setting for each scenario is done accordingly. The network topology information represented in graph is inserted into adjacency matrix and the maximum generation number for the termination is 15 iterations.

This algorithm is running until the feasible path is found or the maximum number of iteration is exceeded.

The results of comparison of iteration numbers taken to find the feasible path is shown in Table 1.

Table 1. Comparison of iteration number.

Nodes	Method	Number of iterations
50	GLBR	>14
	ARGAQ	13
	MURUGA	4
	IRAGA	2

As shown in Fig. 1, it is clear that IRAGA finds the possible path at 2nd generation, while MURUGA has managed to find it after 3 generations. ARG AQ needs 12 generations to find the possible path, and GLBR needs more than 14 generations to find the possible path.

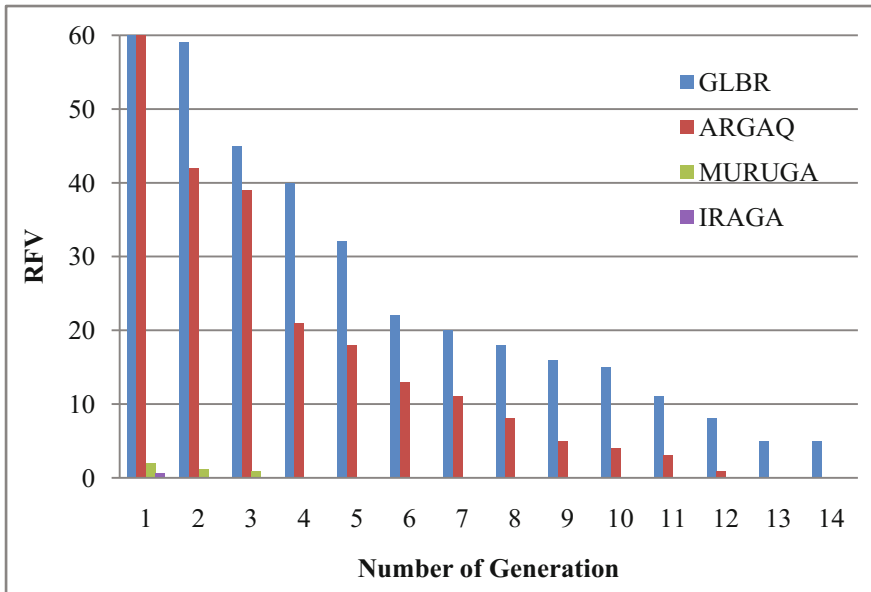


Fig. 1. Performance of GLBR, ARG AQ, MURUGA, and IRAGA.

The result of this study shows that MURUGA, GLBR and ARG AQ need more genetic operations and computation time to find the possible path. Thus, it can be deduced that GRAPC is the efficient method that outperforms MURUGA, GLBR and ARG AQ. The proposed routing algorithm performs better in terms of number of generation, which means less computation time to find the efficient path compared with the other methods.

The use of crossover refining has helped in selecting the right the populations, and avoiding infeasible paths selection. Moreover, the refining crossover operation is used instead of using two successive operations (crossover and mutation), which reduces the time to find the target quality.

6 Conclusion

In this paper, IRAGA was proposed on the basis of multi constraint to find a feasible path, and minimum time of routing selection in addition provide alternative path according to the quality of constraint. In IRAGA, the population is generated according to connectivity matrix to avoid looping routing and invisible paths. Moreover, IRAGA utilizes GA by using crossover refining to reduce computation time which uses the contents of crossover point instead of the index point. These features avoid the invalid paths, which minimize the time taken to find the feasible path. Finally, the simulation results indicate that the proposed algorithm outperforms GLBR, ARGAQ, and MURUGA in terms of the time taken to find the feasible path.

References

1. Taneja, A.M.S., Kush, A., Bhushan B.: Power management in mobile ad-hoc network. *Int. Trans. J. Eng. Manag. Appl. Sci. Technol.* (2011)
2. Patil, V.P.: Efficient AODV routing protocol for MANET with enhanced packet delivery ratio and minimized end to end delay. *Int. J. Sci. Res. Publ.* **2**(8), 1 (2012)
3. Priya, S., Gopinathan, B.: Reliable data delivery in MANETs using VDVH framework. In: *International Conference on Information Communication and Embedded Systems (ICICES)*, pp. 1–6. IEEE (2014)
4. Shen, H., Zhao, L.: ALERT: an anonymous location-based efficient routing protocol in MANETs. *IEEE Trans. Mob. Comput.* **12**(6), 1079–1093 (2013)
5. Kumar, J., Kulkarni, M., Gupta, D.: Effect of black hole attack on MANET routing protocols. *Int. J. Comput. Netw. Inf. Secur.* **5**(5), 64 (2013)
6. Lu, T., Zhu, J.: Genetic algorithm for energy-efficient QoS multicast routing. *IEEE Commun. Lett.* **17**(1), 31–34 (2013)
7. Razouqi, Q., Boushehri, A., Gaballah, M., Alsaleh, L.: Extensive simulation performance analysis for DSDV, DSR and AODV MANET routing protocols. In: *27th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 335–342. IEEE (2013)
8. Rao, M., Singh, N.: Quality of service enhancement in MANETs with an efficient routing algorithm. In: *International Advance Computing Conference (IACC)*, pp. 381–384. IEEE (2014)
9. Barolli, L., Koyama, A., Sukanuma, T., Shiratori, N.: A genetic algorithm based QoS routing method for multimedia communications over high-speed networks. *IPSN J.* **44**(2), 544–552 (2003)
10. Yang, S., Cheng, H., Wang, F.: Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **40**(1), 52–63 (2010)

11. Abdullah, N., Baklizi, M., Al-wesabi, O., Bin-Salem, A.A., Ramadass, R., Ahmadpour, S.: Video delivery based on multi constraint genetic and tabu search algorithms. *Int. J. Comput. Sci.* **9**(2), 136–140 (2011)
12. Yen, Y.S., Chao, H.C., Chang, R.S., Vasilakos, A.: Flooding-limited and multi-constrained QoS multicast routing based on the genetic algorithm for MANETs. *Math. Comput. Model.* **53**(11), 2238–2250 (2011)
13. Abdullah, N., Baklizi, M., Ramadas, S., Al-wesabi, O.: Multi-constrained routing using genetic algorithm. In: *RFID-TA*, pp. 1–4 (2013)
14. Lin, C.H., Yu, J.L., Liu, J.C., Lee, C.J.: Genetic algorithm for shortest driving time in intelligent transportation systems. In: *International Conference on Multimedia and Ubiquitous Engineering (MUE)*, pp. 402–406. IEEE (2008)
15. Roberge, V., Tarbouchi, M., Labonté, G.: Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Trans. Ind. Inf.* **9**(1), 132–141 (2013)
16. Vyas, N., Shah, R.: Intelligent and efficient cluster based secure routing scheme for wireless sensor network using genetic algorithm. *Int. J. Digit. Appl. Contemp. Res.* **2**, 1–7 (2014)
17. Oda, T., Elmazi, D., Barolli, A., Sakamoto, S., Barolli, L., Xhafa, F.: A genetic algorithm-based system for wireless mesh networks: analysis of system data considering different routing protocols and architectures. *Soft Comput.* **20**(7), 2627–2640 (2016)
18. Yao, G.S., Dong, Z.X., Wen, W.M., Ren, Q.: A routing optimization strategy for wireless sensor networks based on improved genetic algorithm. *J. Appl. Sci. Eng.* **19**(2), 221–228 (2016)
19. Munetomo, M., Takai, Y., Sato, Y.: An adaptive routing algorithm with load balancing by a genetic algorithm. *Trans. IPSJ* **39**(2), 219–227 (1998)
20. Leela, R., Thanulekshmi, N., Selvakumar, S.: Multi-constraint QoS unicast routing using genetic algorithm (MURUGA). *Appl. Soft Comput.* **11**(2), 1753–1761 (2011)