Research Article

A memetic optimization algorithm for multi-constrained multicast routing in ad hoc networks

Rahab M. Ramadan1*, Safa M. Gasser2*, Mohamed S. El-Mahallawy2*, Karim Hammad2‡, Ahmed M. El Bakly1‡

1 Department of Basic and Applied Sciences, Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt, 2 Department of Electronics and Communications, Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt

* These authors also contributed equally to this work.
‡ These authors also contributed equally to this work.
* rhabmz@aast.edu

Abstract

A mobile ad hoc network is a conventional self-configuring network where the routing optimization problem—subject to various Quality-of-Service (QoS) constraints—represents a major challenge. Unlike previously proposed solutions, in this paper, we propose a memetic algorithm (MA) employing an adaptive mutation parameter, to solve the multicast routing problem with higher search ability and computational efficiency. The proposed algorithm utilizes an updated scheme, based on statistical analysis, to estimate the best values for all MA parameters and enhance MA performance. The numerical results show that the proposed MA improved the delay and jitter of the network, while reducing computational complexity as compared to existing algorithms.

Introduction

A Mobile Ad hoc NETwork (MANET) is a collection of arbitrarily located nodes, thus the interconnections between them are dynamically changing [1]. The absence of infrastructure and the dynamic nature of these networks make them adequate for operating in extreme situations, such as battlefield and disaster recovery [2, 3]. Developing routing protocols for MANETs is highly challenging, due to; continuous variation of the network topology, limited resources in mobile nodes and rapid change of radio propagation conditions [4]. Therefore, recent studies have focused on designing protocols for effective routing, mobility management, data transport, and quality of service (QoS) provisioning [1, 4, 5].

There is a growing requirement to handle real-time applications in MANETs to support their deployment in public domains (e.g., highways, airports, etc.). Numerous routing protocols have been proposed in an attempt to satisfy various objectives, such as efficient utilization of bandwidth and battery capacity [6, 7], optimization of metrics (e.g., throughput and end-to-
end delay), fast route convergence, and elimination of loops [8, 9]. QoS routing in ad hoc networks is required in many applications [10], yet is challenging to maintain because the network topology may continuously change and the available state information for routing is inherently imprecise [11]. QoS routing may be based on various parameters, such as end-to-end delay, delay jitter, packet loss probability, etc. [9].

Routing problems are classified into unicast and multicast routing. The unicast problem is to find a feasible route from a single source to a single destination. Several routing protocols have been developed to solve the QoS unicast routing problem from different aspects [6–8]. Some works [12, 13] have addressed the problem by offering memetic-based optimization techniques. However, only the delay constraint was considered in these works.

On the other hand, the multicast routing problem aims to find a tree structure, which is used for efficiently delivering the same data stream to different destinations of a network [9]. Different considerations are usually involved when solving the QoS multicast routing problem, such as minimization of routing cost [14], maximization of throughput [15], and minimization of link delay. Therefore, the QoS multicast routing problem is defined as a Non-deterministic Polynomial (NP) hard problem for large-scale and wide area networks [16]. This makes traditional methods of polynomial complexity such as Dijkstra’s algorithm, Bellman-Ford algorithm and Floyd-Warshall algorithm unsuitable for solving this kind of problems for real time applications [12]. Therefore, researchers have studied several metaheuristic methods to solve this problem, such as Ant Colony Optimization (ACO) [17], Particle Swarm Optimization (PSO) [18], Bee Life-based Algorithm (BLA) [19] and Genetic Algorithms (GA) [9, 20, 21]. In those algorithms, first, the fitness of tree structures is analyzed based on the whole network topology, followed by optimization of the tree structures using some metaheuristic operators.

However, weak robustness, low convergence speed and low search efficiency pose limitations to the performance of those algorithms.

Interesting results from previous works [12, 13] in solving the unicast routing problem using memetic algorithm, encouraged us to investigate the use of memetic algorithm in solving the QoS multicast routing problem which is a non-deterministic polynomial hard problem. In [22], Bäck and Schütz suggested a time-dependent mutation rate to solve difficult combinatorial optimization problems (multiple knapsack and maximum independent set). The performance of their self-adaptation method was significantly better than the performance of other conventional algorithms in terms of convergence reliability and velocity. We have extended their work on GA to develop a new MA that uses an adaptive mutation parameter to solve the multicast routing problem and support QoS-featured routing, based on four different QoS parameters. The performance of the proposed MA is analyzed with regard to three important QoS constraints: the end-to-end delay, the bandwidth and the delay jitter. The QoS constraints and the network cost of the multicast tree constitute the objective function that our algorithm strives to minimize. To the best of the authors’ knowledge, no previous study has simultaneously analyzed all MA parameters, including population size, maximum iteration number, mutation, and crossover probabilities. The present study proposes a new taxonomy of MA parameters and performs an extensive analysis of these components to determine the best levels for all parameters in the context of MANET routing.

The rest of this paper is organized as follows: Section 2 presents a description of the QoS multicast routing problem. Section 3 explains the development of MA. In Section 4, the algorithm complexity analysis is presented. Section 5 discusses the formulation and testing of the
experimental design and demonstrates simulation results. Finally, Section 6 discusses the results obtained and concludes the study.

**QoS multicast routing problem**

An *ad hoc* network model is presented as an undirected weighted graph $G = (M, L)$ where $M$ denotes the set of nodes labeled as $(1, 2, 3, \ldots, m)$, and $L$ denotes the set of communication links connecting two neighboring nodes. Each of these communication links indicates the existence of a radio interface for each node, with a common wireless channel, and all of the links have weights representing link transmission cost, delay values, link transmission delay variation (jitter), and estimated link bandwidth. Fig 1 shows an example of an *ad hoc* network consisting of 16 nodes. As an example, the link between node 0 and node 2 has a link cost = 267, delay = 123ms, jitter = 1.75ms and bandwidth = 797.4 Kbit/s.

In order to transfer a data message from the source $s$ (where $s \in M$) to a group of destinations $D$ (where $D \in \{M - s\}$), a multicast tree $T(s, D)$ is generated from the nodes of graph $G$. A multicast tree $T(s, D)$ represents a group of different paths from node $s$ to all destinations $D$. Therefore, the objective of a multicast routing problem is to find a multicast tree $T(s, D)$ that has a set of paths with acceptable QoS metrics. In the present study, four QoS constraints are considered in solving a multicast routing problem: transmission cost, end-to-end delay, delay jitter, and available bandwidth. According to [19], the total cost of $T(s, D)$ is given by

$$\text{Cost}(T(s, D)) = \sum_{l \in T(s, D)} C_l,$$

where $C_l$ is the cost of the communication link ($l$). The delay of a route ($R$) from the source node ($s$) to a destination node ($d$) is equal to the sum of the delay values of all links that form this route

$$\text{Delay}(R(s, d)) = \sum_{l \in R(s, d)} D_l,$$

where $D_l$ is the transmission delay on the communication link ($l$). Similarly, the end-to-end delay variation (jitter) is given by

$$\text{Jitter}(R(s, d)) = \sum_{l \in R(s, d)} J_l,$$

where $J_l$ is the delay jitter on the communication link ($l$).

The bandwidth of $R(s, d)$ is the minimum link bandwidth in the entire route, so bandwidth

$$\text{Bandwidth}(R(s, d)) = \min \{\text{bandwidth}(l) \mid l \in R(s, d)\}.$$
Therefore, the current QoS multicast routing problem is formulated around four objectives and the fitness function is to identify a multicast tree \( T(s, D) \) that minimizes the weighted combination of cost, delay, jitter, and bandwidth. This can therefore be defined as follows [19]:

\[
\text{Minimize } F(T(s, D)) = w_1 \text{Cost}(T(s, D)) + w_2 \text{Delay}(R(s, d)) + w_3 \text{Jitter}(R(s, d)) + w_4 \text{bandwidth}(R(s, d))
\]  

Subject to:

\[
\text{Delay}(R(s, d)) \leq Q_d
\]

\[
\text{Jitter}(R(s, d)) \leq Q_j
\]

\[
\text{bandwidth}(R(s, d)) \geq Q_b
\]

Where \( w_1, w_2, w_3 \) and \( w_4 \) are the objective weighting coefficients used to evaluate the problem based on the relative importance of these four objectives. Furthermore, \( Q_d \) is the upper bound of the delay, \( Q_j \) is the upper bound of the jitter and \( Q_b \) is the lower bound of the bandwidth of every route.

**Adaptive mutation based MA scheme**

Although many studies have evaluated the improvement of the QoS of MANET routing problems using genetic algorithms (GA) [1, 5, 9, 20, 21], yet few of them employed MA in their applications. In addition, several studies [23, 24] showed that MA outperforms GA in many applications. Since, MA includes a local search process; it is helpful in effectively solving different optimization problems [23].

Although MA generally appears to be a single robust algorithm, which contains the same operators and holds the same algorithmic logic, the algorithm itself is, in fact, significantly different for fully-defined domains [25]. The main reason behind this is that the Evolutionary Algorithm (EA) has several parameters, and any combination of these parameters has different impact on the performance of the EA [26, 27]. These parameters are classified into two categories: structural and numerical. Structural parameters are the main factors affecting the EA performance, and the most complex set of parameters to be addressed in an EA application [28]. As the name of the category suggests, they are concerned with the structure of an EA, hence any change in a structural parameter value calls for significant alterations in the coding pattern of the EA. The entire algorithm must then be reformulated, including the coding scheme, operator types, and stopping criterion. Numerical parameters constitute the second category of the taxonomy. The initial population, generation number, population size, crossover, and mutation probabilities are the primary variables recognized in this category. These parameters are easy to manage when the coding structure is established. Although modifications in the levels of these parameters do not require extensive coding, a different combination leads to drastic changes in EA performance [29, 30]. Several previous studies have analyzed numerical parameters; however, none made general conclusions regarding computing optimum values for these parameters [31]. Moreover, these studies analyzed the parameters separately, ignoring the interaction between different parameters, which introduces a gap in the literature when it comes to numerical parameters.

DeJong [32] and Fogarty [33] attempted to identify the optimal control parameters for the EA on a chromosomal representation, and concluded that if the mutation rate is too high,
then the search is random, irrespective of other parameter settings. They suggested optimal population size values in the range of 50–100, a mutation probability of 0.001, and a single-point crossover with a rate in the range of 0.6–0.95. Grefenstette [34] reported that, in small populations in the range of 20–40, good performance is generally coupled with either a high crossover rate and a low mutation rate, or a low crossover rate and a high mutation rate. [34] also concluded that a mutation rate above 0.05 is generally detrimental to the optimal performance of EAs, when optimal control parameters with a population size of 30 individuals, a mutation rate of 0.01, and a two-point crossover rate of 0.95 are deployed. Schaffer et al. [35] observed that EA performance has a greater sensitivity to mutation rate than to crossover rate, using the same optimal parameter settings, suggested by Grefenstette [34]. In a population size of 20–30 individuals, the optimal mutation rate was established as between 0.005 and 0.01, and the optimal crossover rate in a range of 0.75–0.95. However, Alander [36] reported that an approximation value between n and 2n is optimal for the population size, where n equals the number of network nodes. A typical size of 10–40 population is common in MAs, because the computational complexity of the local search does not permit the evolution of much larger populations in real time applications [23, 24]. As per the results reported in [34–36], the levels of population size are taken as 10, 20, and 40.

Several previous studies [33, 37, 38] showed that mutation probability should be decreased during the convergence stage, so that each chromosome has the crossover probability $p_c$ and the mutation probability $p_m$ necessary to undergo crossover and mutation, respectively. During the execution of EAs, $p_m$ is adapted based on generation progress, with probabilities being large when the mean fitness is near the initial process, and small for particular chromosomes with near optimum finesses. Mühlenbein [39] investigated the optimal mutation rate for a single parent that generates offspring with improved survival and fitness for the next generation. For example, Mühlenbein [39] recommends an optimal approximation of the mutation rate of $p_m = 1/n$ (where n is the number of nodes in network) to improve the objective function value. This setting yields surprisingly good results for a variety of NP-hard combinatorial optimization problems, such as the maximum independent set problem [40], the multiple knapsack problem [41], and the minimum vertex cover problem [42]. Bäck and Schütz [22] suggested a time-dependent mutation rate $p_m$ as follows:

$$p_m = (2 + \frac{2}{T-1})^{-1}$$

Where $t \in \{0, 1, \ldots, T-1\}$ denotes the generation counter, and $T$ is a given maximum number of generations. From the range of $p_m = \frac{1}{2} - \frac{1}{n}$, the adapted intelligent mutation rate described in Eq (5) can easily satisfy that range. During the execution process, a new mutation rate value is created for each generation within that range, so it is certainly plausible to state that Eq (5) better serves as a general parameter setting rule than does $p_m = \frac{1}{n}$.

The present study examines the effects of MA primary parameters on performance, with regard to the optimal fitness identified for ad hoc network routing problems, and proposes guidelines for parameter selection for this particular problem domain. Test problems of the ad hoc network, which are extracted from previous studies, are selected for simulation [19]. The present study has two main aims; first, to propose a new taxonomy for MA parameters, and second, to present an extensive analysis of these parameters to draw a general conclusion applicable to an ad hoc network domain.

An MA provides a near-optimal solution by evolving a group of successive generations. As shown in Fig 2, the first generation is randomly selected from the total population pool. In
Fig 2. Memetic algorithm flow chart.

https://doi.org/10.1371/journal.pone.0193142.g002
addition, the total population should be very high so that it is not possible to computationally evaluate every solution in the whole pool. Otherwise, if the population is not large enough then evaluating each solution separately and selecting the best would be more efficient than using the memetic algorithm.

Parent selection is made by tournament selection, where two individuals are randomly selected, and one-point crossover is carried out. The crossover point is selected randomly, but only at the valid boundaries, since the main concept of crossover is to pass on the genetic information [43]. After completion of the crossover operation, the two new offspring are evaluated. If the fitness of a single offspring is superior to the fitness of either parent, the new offspring is selected. This operation is carried out to reflect the contribution of the genes to the improvement in fitness. The proposed MA is explained in the following Pseudocode.

**PseudoCode 1: Proposed MA**

1. Create a random population of \( n = 40 \) multicast trees (chromosomes).
2. Evaluate the fitness function \( F(i) \) of each chromosome \( i \) in the population based on Eq (1).
3. While (end condition not satisfied) do {
   a. Perform crossover for the selected parents with probability \( p_c = 0.85 \), to form new offspring.
   b. Execute mutation on selected chromosomes with probability described in Eq (5).
   c. Conduct a local search procedure (a mixture of greedy and steepest approaches) for selected chromosomes.
   d. Place the new accepted offspring (feasible solutions) to form a new population.
   e. [Replace old population] Use the newly generated population for a further run of the algorithm.
   f. [Test] If the end condition is satisfied (iteration number was set to 50), stop, and display the optimal routes in the current population.
}

A chromosome (single solution) is a multicast tree that contains the different routes from source node to each one of the multicast group members (destinations) via a set of intermediate nodes. A route is represented as a string of nodes that link the source to one of the destinations. Then, the population is a set of multicast trees (feasible solutions).

Finally, the local search implemented is a mixture of greedy and steepest approaches. First, the procedure begins with the random selection of one gene (node) removal position \( i \) (the greedy approach). The location of this gene is then removed. Second, the local search picks an insertion position \( j \) for the removed location. The optimal insertion position is then selected from all of those possible (the steepest approach). If the gene insertion at the optimal position results in an overall improvement of the objective function (fitness value), accounting for both changes due to the removal and the insertion, the shift move is performed; otherwise, the process is switched back.
Complexity analysis of the proposed scheme

In order to describe the computational resource running time of the MA, the complexity is computed as follows. Let N and M be the number of nodes in the network, and the number of chromosomes (solutions), respectively. The MA starts off using \( O(M \times (n - 1) \times \log(n - 1)) \) time units to develop randomly M chromosomes. The MA iterates by letting \( p_c \) be the crossover probability and \( p_m \) be the value of the mutation probability. The number of offspring developed by crossover consumes \( O(N \times p_c \times [M \times (N + 1)]) \) time units, while the number of mutated offspring consumes \( O(p_m \times [M \times (N + 1)]) \) time units. Since adaptive mutation rate described in Eq (5) is used, we consider \( p_m = \frac{1}{2} \) as the upper bound value. This concerns the reproduction phase. Finally, a local search is introduced which spends \( O(N) + N \times O(1) \) \( O(N) \) time units. Thus, the total time complexity for each iteration is recorded as:

\[
O((M \times (n - 1) \times \log(n - 1)) + (N \times p_c \times [M \times (N + 1)]) + (1/2 \times [M \times (N + 1)] + N))
\]

From Eq (6), it is clear that the MA has a linear complexity that is reasonable in time and in developing the lowest-cost network trees. In addition, the population size of MAs is typically smaller when compared to GAs [24]. This small population size allows the MA to have a faster convergence rate.

Computational complexities of EAs with different coding schemes and operators are hardly described uniformly. Hence, we just focus on the computational complexities of crossover operators. As in [9, 19], the complexity of BLA at the cross over stage for \( (p_c = 0.85) \) is in the order of \( (1.5 \times M) \), while that of ISGSA and EEGA is of order \( (M^2) \). However, in the proposed MA algorithm, the cross over stage requires a complexity in the order of \( (0.85 \times M) \).

Proposed algorithm simulation

The main aim of the proposed MA is to solve the multicast routing problem by finding a multicast tree of paths from source to different destinations that offers optimal fitness value. This optimal fitness is the minimized value of the weighted combination between the optimization metrics (cost, delay, jitter and bandwidth) calculated by Eq (1). All experiments were implemented on MATLAB 2016a, with an Intel \textsuperscript{\textregistered} Core \textsuperscript{\textregistered} i5-4200M CPU, and 6 GB RAM.

A. Simulation parameters setup

A test problem from a previous study is used [9] to determine the optimal MA parameters, based on the simulation-optimization technique, and the significance of different values of these parameters using factorial design and Analysis Of Variance (ANOVA). The optimal fitness value obtained is the performance measure considered throughout the analysis. The objective function value (the fitness value), one-point crossover with a probability of \( p_c \) (using roulette wheel selection), self-adaptive mutation with a probability of \( p_m \), and maximum rotation (iteration) numbers as a stopping criterion, were employed in the experimental analysis. The roulette wheel selection procedure was also used for the reproduction process. If unfeasible solutions are reached after the operations of both crossover and mutation, then these chromosomes are discarded, and new ones are generated from the beginning. The system under consideration is an ad-hoc network consisting of three separate network models with varying node content (15, 20, and 50 nodes, respectively).

The performance of MA is highly dependent on the initially generated population. Therefore, it is prejudicial to state that one memetic parameter is better than another by simply
comparing two solutions with different initialization. To neutralize the effect of the initial population, 10 trials were conducted for each condition in the present study with randomly selected, initial populations. All of the memetic parameters for each evaluation are given in Table 1 except for the mutation rate probability, which is set on according to Eq (5).

Different static crossover rates are used to examine the variety of results, as shown in Table 1, in order to analyze the impact of variation of crossover rate and to identify the value of the crossover rate that yields the best solution. In addition, an adaptive mutation is used, such that the mutation rate is high in early generations and decreases as generations increase. The selected, adaptive mutation rate is a function of the generation number, and is obtained by Eq (5), where $p_m$ refers to the mutation probability and $n$ refers to the number of generations. Since there are three distinct numerical parameters, a three-way factorial design is employed. For each parameter level, 10 replication results, gathered from running the simulation with the assigned parameters, are obtained. ANOVA is used to determine the significance of each effect on the fitness value, using Statistical Package for the Social Sciences Software, version 22 (http://www.spss.com). For each initialized population size, a random search heuristic of 10, 20, and 40 random populations are drawn from all populations. Factors identified as significant are further analyzed using Duncan’s test to differentiate between the factor levels. In addition, if a higher interactive effect is found, a simple effect technique [44] is used to demonstrate the type of effect at each factor level. The dependent variable is the fitness value of the problem and the independent variables are the number of rotations, population size, and crossover rate. The Shapiro and Wilk [45] test is implemented to test the normality of the data, and statistical significance is set at a confidence level of 95%.

After examining data normality, tests for significance are carried out to compare the three assigned parameter levels, based on the evaluation parameters in Table 2a, 2b and 2c. For the 15 nodes network, only population size is significant, as shown in Table 2. Using the Duncan

Table 1. The proposed algorithm settings.

| Parameters                                      | Values                                      |
|-------------------------------------------------|---------------------------------------------|
| Number of generations for a trial (e.g., iterations) | 20 and 50                                    |
| Number of individuals (e.g., chromosomes per generation) [Population size] | 10, 20, and 40                               |
| Crossover rates                                  | 0.6, 0.75, 0.85, and 0.95                   |
| Mutation rate                                    | The adapted mutation rate described in Eq (5) |
| Number of genes (nodes) in each individual (e.g., three different problem sets) | 15, 20, and 50                               |

Table 2. Analysis of variance for the 15 nodes network.

| Source                     | Type III Sum of Squares | df | Mean Square | F     | Sig. |
|----------------------------|-------------------------|----|-------------|-------|------|
| Rotation                   | 22.204                  | 1  | 22.204      | 1.473 | 0.226|
| Pop_Size                   | 1508.108                | 2  | 754.054     | 50.031| 0.000|
| Cross_over                 | 7.946                   | 3  | 2.649       | 0.176 | 0.913|
| Rot*Pop_Size               | 34.058                  | 2  | 17.029      | 1.130 | 0.325|
| Rot’Cross_over             | 9.079                   | 3  | 3.026       | 0.201 | 0.896|
| Pop_Size * Cross_over      | 30.592                  | 6  | 5.099       | 0.338 | 0.916|
| Rot * Pop_Size * Cross_over| 8.508                   | 6  | 1.418       | 0.094 | 0.997|
| Error                      | 3255.500                | 216| 15.072      |       |      |
test with a 5% significance level to differentiate between the population size levels, the fitness of the 40 generations (i.e., with a mean value and a standard deviation of 18.675±2.385) was superior to the fitness of both the 20 and 10 generations (i.e., with a mean value and a standard deviation of 24.713±4.401 and 20.725±4.195, respectively).

Population size is also a significant factor for the 20 nodes network, as shown in Table 3. The fitness of both the 40 and 20 generations (with mean values and standard deviations of 246.464±0.0 and 246.464±0.0, respectively) are superior to the fitness of the 10 generations (254.16±29.993).

Population size is also a significant factor for the 50 nodes network, as shown in Table 4. The fitness of the 40 generations (with a mean value and standard deviation of 2.457±0.109) is superior to the fitness of the 20 generations (2.619±0.166), and the fitness of the 20 generations is superior to the fitness of the 10 generations (2.846±0.277). In addition, rotation by crossover interaction is significant; the simple effect technique [44] is employed to describe this interaction. The analysis revealed that the fitness of 50 rotations with a crossover rate of 0.85 (with a mean value and standard deviation of 2.606±0.035) is superior to the fitness of 20 rotations with a crossover rate of 0.85 (with a mean value and standard deviation of 2.728±0.035) using $t(29) = 2.453$, $p < 0.02$. The other comparisons are not significant at the $p < 0.05$ level.

On the basis of the statistical results described, it is concluded that the following parameters give the best fitness value for ad hoc network domains: 50 rotations, population size of 40 chromosomes, adaptive mutation rate, and 0.85 crossover rate, irrespective of the ad hoc network size (e.g., number of nodes included).

### B. Simulation results

The efficacy and the validity of the proposed MA are tested via comparing it to other EA techniques. These techniques are; the bee life-based algorithm (BLA) [19], the bees algorithm (BA)
[46], the marriage in honey bees optimization algorithm (MBO) [47], the genetic algorithm (GAMRA) [20], the energy efficient genetic algorithm (EEGA) [48], the genetic simulated annealing algorithm (ISGSA) [21], and their updated versions, PM-GAMRA, PM-EEGA, and PM-ISGSA [9]. The simulation is conducted using two datasets. These datasets are available in the Appendix of the study conducted by Liang et al. [9].

The first dataset is a 15-node random graph, which is constructed on the basis of Lu and Zhu [48], where the costs of links are generated in the range between 2 and 39, and delay values are between 1 and 7. Node 0 represents the source node \( s \), and nodes \{2, 5, 6, 8, 14\} represent the multicast group members (destination nodes). A chromosome is chosen as a multicast tree that contains five paths, with each one being a string of nodes from the source \( s \) (node = 0) to one of each of the destination nodes (nodes 2, 5, 6, 8, and 14). The second dataset is 20 nodes, and is generated using a network simulator (http://www.isi.edu/nsnam/ns/), developed by Salim et al. [19]. Node 0 represents the source node \( s \), and nodes \{4, 9, 14, 19\} represents the multicast group members (destination nodes). A chromosome is chosen as a multicast tree that contains four paths, with each one being a string of nodes from the source \( s \) (node = 0) to each of the destination nodes (nodes 4, 9, 14, and 19). In order to evaluate the fitness of each individual (tree structure), Eq (1) is applied, in which the QoS weights are; \( w_1 = 10, w_2 = 1, w_3 = 10, \) and \( w_4 = 1 \). The weight values reflect the importance assigned to each one of the four considered QoS parameters [19].

Several measures are introduced to compare the effectiveness and performance of the proposed MA against the aforementioned techniques. These measures are; \( R_{\text{min}}, R_{\text{mean}} \) and \( R_{\text{variance}} \), denoting the minimum, mean, and variance of the results, respectively. The results are all based on K times repeated trials. \( R_{\text{min}} \) is computed as \( \min \{ R_{i, \text{step}(j)}, i = 1, 2, \ldots, K \} \), where \( R_{i, \text{step}(j)} \) is the optimal solution to the routing problem in the \( j \) step for the \( i \)th time computation up to \( K \). A similar procedure is followed to compute the \( R_{\text{mean}} \) and \( R_{\text{variance}} \) measures.

Fig 3 shows the \( R_{\text{min}}, R_{\text{mean}}, \) and \( R_{\text{variance}} \) of the results obtained via PM-GAMRA, PM-ISGSA [9], and the proposed MA, using the first dataset. It is clear from the figure that the
proposed MA provides mean cost = 86 and mean delay = 18. On the other hand, the former two GAs provide mean cost = 108 and mean delay = 18. Moreover, the $R_{\text{variance}}$ of the proposed MA is zero value which is less than those reported for the former GAs, which indicates that the proposed MA is more robust.

Both PM-EEGA and PM-ISGSA generate near optimum solutions in the evolution, yet their costs and fitness results, did not reach the level of success of the proposed MA results. Fig 3a and 3b show that the $R_{\text{min}}$, $R_{\text{mean}}$, and $R_{\text{variance}}$ of the MA fitness value are clearly less than those of PM-EEGA and PM-ISGSA. These results indicate that the MA scheme improved searchability as to the optimal solution.

Fig 4 shows the optimal multicast trees reached by the proposed MA, BLA, BA, and MBO, respectively, using the second dataset. The MA fitness result is compared in terms of common QoS constraints (cost, delay, jitter, and bandwidth) with the other meta-heuristic algorithms; MBO [47], BA [46], BLA [19], PM-EEGA, and PM-ISGSA [9]. The fitness values of these algorithms (as listed in Table 5) are induced from [9].

The results in Fig 5 show that the fitness values of the proposed MA scheme are lower than those of the other algorithms. This indicates that the proposed MA scheme can offer a high level of robustness for ad hoc networks by constantly identifying optimal solutions, supported by a 72% success rate.

Fig 4. The optimal multicast trees for: (a) MA, (b) BLA, (c) BA, and (d) MBO for the second dataset.

https://doi.org/10.1371/journal.pone.0193142.g004
In order to compare the execution time of MA against other approaches (BLA, BA, and MBO), the improvement progress for the algorithms is summarized in Fig 6 and Table 6. This figure illustrates the stagnation of the optimal fitness solution for each generation (iteration times) developed by each algorithm. It is observed that the proposed MA algorithm reaches the optimal fitness (18,822.5) in a reasonable number of generations (the 9th generation). This shows that a reasonable convergence speed is associated with the low fitness solution compared to other approaches. For example, the MBO algorithm reaches its stagnation state at the 16th generation, with a fitness value of 18987.34. However, BA algorithm reaches its stagnation state

Table 5. The fitness and property values of the optimal results generated by the different algorithms on the applied dataset. The bold numbers emphasize the optimal values of properties.

| Algorithms | MA       | BLA      | BA       | MBO      | PM-EEGA   | PM-ISGSA  |
|------------|----------|----------|----------|----------|-----------|-----------|
| Fitness    | 18822.5  | 18905.29 | 20266.64 | 18987.34 | 18905.29  | 18822.64  |
| Cost       | 1746.19  | 1740.38  | 1876.53  | 1762.65  | 1740.38   | 1746.29   |
| Delay      | 493.14   | 616.34   | 616.13   | 493.14   | 616.34    | 493.14    |
| Jitter     | 7        | 8.75     | 8.75     | 7        | 8.75      | 7         |
| Bandwidth  | 797.5    | 797.54   | 797.65   | 797.57   | 797.54    | 797.54    |

https://doi.org/10.1371/journal.pone.0193142.t005

Fig 5. The optimal fitness solutions identified by applying different algorithms on the second dataset.

https://doi.org/10.1371/journal.pone.0193142.g005
at the 7th generation, but with a poor fitness value (20,266.64). In this case, BA dropped early in local optimal value. Also, BLA reached optimal fitness (18,905.29) in the 13th generation.

To sum up, the above results showed the effectiveness of the proposed adaptive mutation-based memetic algorithm (MA) which combines the advantages of both genetic algorithm and local search method. Combining both techniques in solving the multi-cast routing problem outperformed MBO [47], BA [46], BLA [19], and the PM-modified Physarum network model [9]. The use of local search technique allows the algorithm to avoid premature convergence to suboptimal solutions. According to the obtained results, the best fitness is the fitness obtained by MA (18,822.5), achieving lowest delay (493.14ms) and lowest jitter (7ms) with a reasonable transmission cost value. Other algorithms that achieved the same delay or jitter performance, obtained similar performance at a higher transmission cost. As a result, the proposed

![Fitness Value versus Generations](https://doi.org/10.1371/journal.pone.0193142.g006)

To sum up, the above results showed the effectiveness of the proposed adaptive mutation-based memetic algorithm (MA) which combines the advantages of both genetic algorithm and local search method. Combining both techniques in solving the multi-cast routing problem outperformed MBO [47], BA [46], BLA [19], and the PM-modified Physarum network model [9]. The use of local search technique allows the algorithm to avoid premature convergence to suboptimal solutions. According to the obtained results, the best fitness is the fitness obtained by MA (18,822.5), achieving lowest delay (493.14ms) and lowest jitter (7ms) with a reasonable transmission cost value. Other algorithms that achieved the same delay or jitter performance, obtained similar performance at a higher transmission cost. As a result, the proposed

Table 6. A comparison of optimal fitness solutions in generations (iteration times).

| Generation | MA         | BLA         | BA          | MBO         |
|------------|------------|-------------|-------------|-------------|
| 1          | 35,717.64  | 46,290.62   | 46,309.36   | 49,163.7    |
| 2          | 27,134.35  | 30,032.38   | 34,786.42   | 36,696.46   |
| 3          | 21,692.76  | 30,015.78   | 30,457.64   | 36,667.44   |
| 4          | 20,749.46  | 26,591.38   | 30,143.22   | 30,875.04   |
| 5          | 20,390.94  | 22,862.6    | 30,143.22   | 30,875.04   |
| 6          | 20,075.92  | 22,606.02   | 30,093.62   | 26,969.8    |
| 7          | 19,874.16  | 22,342.82   | 20,266.64   | 26,969.8    |
| 8          | 19,340.83  | 19,683.8    | 20,266.64   | 24,757.2    |
| 9          | 18,822.5   | 19,683.8    | 20,266.64   | 24,757.2    |
| 10         | 18,822.5   | 19,297      | 20,266.64   | 24,197      |
| 11         | 18,822.5   | 19,297      | 20,266.64   | 22,717      |
| 12         | 18,822.5   | 19,297      | 20,266.64   | 22,717      |
| 13         | 18,822.5   | **18,905.29** | 20,266.64   | 22,237      |
| 14         | 18,822.5   | 18,905.29   | 20,266.64   | 22,237      |
| 15         | 18,822.5   | 18,905.29   | 20,266.64   | 22,237      |
| 16         | 18,822.5   | 18,905.29   | 20,266.64   | **18,987.34** |
| 17         | 18,822.5   | 18,905.29   | 20,266.64   | 18,987.34   |

https://doi.org/10.1371/journal.pone.0193142.t006
algorithm attained the minimum overall fitness value. Even though the performance of proposed MA and PM-ISGSA is very close, the variance of all parameters is much lower for MA, which leads to a more robust and stable performance. In addition, due to the fact that the proposed MA parameters were optimized based on statistical analysis, the proposed MA is more efficient compared to other algorithms in terms of convergence speed and computational complexity.

**Discussion and conclusion**

This paper proposes a new adaptive mutation based MA to solve the quality of service multicast routing problem for *ad hoc* networks. The proposed MA uses an adaptive mutation parameter. Four metrics were used to test the reliability and the efficiency of the new algorithm: transmission cost, delay, jitter and bandwidth.

The results of the performed statistical analysis revealed that 50 rotations, 40 chromosomes population, adaptive mutation rate, and 0.85 crossover rate provided the optimal fit and were sufficient for *ad-hoc* network applications. These findings are in accordance with the recommendations of previous studies [34–36, 39, 22].

Through simulation, the proposed MA was tested using the dataset published in the study conducted by Liang et al. [9]. The proposed MA demonstrated promising results in terms of its fitness values when compared to the MBO [47], BA [46], BLA [19], and the PM-modified Physarum network model [9]. Remarkably, MA outperformed other algorithms by achieving minimum fitness value = 18,822.5, lowest delay = 493.14ms, and lowest jitter = 7ms. The proposed MA provided the minimum value for most of the tested runs (i.e., supported by a success rate of 72%). The reasons behind this success rate are; the local search included in the MA and an adaptive mutation component, in which mutation probability decreased during convergence stage.

To sum up, the numerical results obtained from previous studies were utilized to test the proposed adaptive mutation parameter that was built in the MA, using a range of commonly used changeable functions. The results of these examinations demonstrated that the form of adaptive MA is capable of creating more robust search performances than its authoritative evolutionary counterparts.

**Supporting information**

S1 File. Data set.

(DOCX)

**Author Contributions**

**Formal analysis:** Rahab M. Ramadan, Mohamed S. El-Mahallawy.

**Methodology:** Rahab M. Ramadan, Safa M. Gasser.

**Software:** Karim Hammad, Ahmed M. El Bakly.

**Supervision:** Safa M. Gasser, Mohamed S. El-Mahallawy, Ahmed M. El Bakly.

**Writing – original draft:** Rahab M. Ramadan.

**Writing – review & editing:** Safa M. Gasser, Mohamed S. El-Mahallawy, Karim Hammad.
References

1. Zhengying W, Bingxin S, and Erdun Z. Bandwidth-Delay-Constrained Least-Cost Multicast Routing Based on Heuristic Genetic Algorithm. Computer Communications, 2001, 24(7): 685–692.

2. Abdelhaq M, Aisaqaur R, Abdelhaq S. Securing Mobile Ad-Hoc Networks Using Danger Theory-Based Artificial Immune Algorithm. PLoS ONE, 2015, 10(5): e0120715. https://doi.org/10.1371/journal.pone.0120715 PMID: 25946001

3. Onwuka E, Folaponmile A, and Ahmed M. Manet: A Reliable Network in Disaster Areas. J. Res. Natl. Dev. Transcampus, 2011.

4. Al-Nahari A, Mohamad MM. Receiver-Based Ad Hoc On Demand Multipath Routing Protocol for Mobile Ad Hoc Networks. PLoS ONE, 2016, 11(6): e0156670. https://doi.org/10.1371/journal.pone.0156670 PMID: 27258013

5. Cheng H and Yang S. Genetic Algorithms with Immigrants Schemes for Dynamic Multicast Problems in Mobile Ad-Hoc Networks. Engineering Applications of Artificial Intelligence, 2010, 23(5): 806–819.

6. Chen L and Heinzelman W. QoS-Aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks. IEEE Journal on Selected Areas in Communications, 2005, 23(3):561–572.

7. Abdulkader TAMM. QoS Routing in Wireless Mesh Networks. M.Sc. Thesis, University of Waterloo, Ontario, Canada, 2008.

8. Vadivel R, Bhaskaran VM. Energy Efficient with Secured Reliable Routing Protocol (EESRRP) for Mobile Ad-Hoc Networks. Procedia Technology, 2012, 4: 703–707.

9. Liang M, Gao C, and Zhang Z. A New Genetic Algorithm Based on Modified Physarum Network Model for Bandwidth-Delay Constrained Least-Cost Multicast Routing. Natural Computing, 2017, 16(1): 85–98.

10. Santhi G. Agent Assisted Multi Constrained Quality of Service Multicast Routing in Mobile Ad Hoc Networks. Ph.D. Dissertation, ANNA University, Chennai, India. 2014.

11. Chitra C. Performance Comparisons of Multi-Objective Evolutionary Algorithms for QoS Routing Problems in Computer Networks. Ph.D. Dissertation, ANNA University, Chennai, India. 2011.

12. Sabar NR, Song A, Tari Z, Yi X, and Zomaya A. A Memetic Algorithm for Dynamic Shortest Path Routing on Mobile Ad-Hoc Networks. 21st International Conference on Parallel and Distributed Systems, 2015 December 14–17, Melbourne, VIC, Australia. IEEE. pp. 60–67.

13. Sabar NR, Turky A, Song A. A Multi-Memory Multi-Population Memetic Algorithm for Dynamic Shortest Path Routing in Mobile Ad-hoc Networks. Proceedings of the 14th Pacific Rim International Conference on Trends in Artificial Intelligence, 2016 August 22–26, Phuket, Thailand. ACM. pp. 406–418.

14. Haghhighat AT, Faez K, Dehghan M, Mowlaei A and Ghahremani Y. GA-Based Heuristic Algorithms for QoS Based Multicast Routing. Knowledge-Based Systems, 2003, 16(5–6): 305–312.

15. Baumann R, Heimlicher S, Strasser M and Weibel A. A Survey on Routing Metrics. TIK Report 262, Computer Engineering and Networks Laboratory, 2007, ETH-Zentrum, Switzerland.

16. Koyama A, Nishie T, Arai J and Baroli L. A GA-Based QoS Multicast Routing Algorithm for Large-Scale Networks. International Journal of High Performance Computing and Networking, 2008, 5(5–6): 381–387.

17. Tseng S, Lin C, Huang Y. Ant Colony-Based Algorithm for Constructing Broadcasting Tree with Degree and Delay Constraints. Expert Systems with Applications, 2008, 35: 1473–1481.

18. Huang C, Chuang Y, Hu K. Using Particle Swarm Optimization for QoS in Ad-Hoc Multicast. Engineering Applications of Artificial Intelligence, 2009, 22: 1188–1193.

19. Salmi B and Abdelhamid M. Bee Life-Based Multi Constraints Multicast Routing Optimization for Vehicular Ad Hoc Networks. Journal of Network and Computer Applications, 2013, 36(3): 981–991.

20. Hwang RH, Do WY and Yang SC. Multicast Routing Based on Genetic Algorithms. Journal of Information Science and Engineering, 2000, 16(6): 885–901.

21. Zhang L, Cai L, Li M and Wang F. A Method for Least-Cost QoS Multicast Routing Based on Genetic Simulated Annealing Algorithm. Computer Communications, 2009, 32(1): 105–110.

22. Bäck T and Schütz M. Intelligent Mutation Rate Control in Canonical Genetic Algorithms. In: Raś ZW, Michalewicz Z, editors. Foundations of Intelligent Systems. Springer-Verlag; 1996. pp.158–167.

23. Elbeltagi E, Hegazy T and Grierson D. Comparison among Five Evolutionary-Based Optimization Algorithms. Advanced Engineering Informatics, 2005, 19(1): 43–53.

24. De Oliveira HCB, Alexandrino JL and De Souza MM. Memetic and Genetic Algorithms: A Comparison among Different Approaches to Solve Vehicle Routing Problem with Time Windows. Sixth International Conference on Hybrid Intelligent Systems, 2006 December 13–15; Rio de Janeiro, Brazil. IEEE. pp. 1–4.
25. Skiena SS. The Algorithm Design Manual. 2nd ed., Springer-Verlag London, 2008.
26. McCall J. Genetic Algorithms for Modeling and Optimization. Journal of Computational and Applied Mathematics, 2005, 184(1): 205–222.
27. Alajmi A and Wright J. Selecting the Most Efficient Genetic Algorithm Sets in Solving Unconstrained Building Optimization Problem. International Journal of Sustainable Built Environment, 2014, 3(1): 18–26.
28. Yuan B and Gallagher M. Combining Meta-EAs and Racing for Difficult EA Parameter Tuning Tasks. In: Lobo FG, Lima CF, and Ichalewicz Z, editors. Parameter Setting in Evolutionary Algorithms. Springer Berlin/Heidelberg; 2007, pp. 121–142.
29. Rudolph G. Self-Adaptive Mutations May Lead to Premature Convergence. IEEE Transactions on Evolutionary Computation, 2001, 5(4): 410–414.
30. Dk Dey. Mathematical Study of Adaptive Genetic Algorithm (AGA) with Mutation and Crossover probabilities. International Journal of Advanced Computer Technology, 2014, 3(5): 765–768.
31. Haupt RL and Haupt SE. Optimum Population Size and Mutation Rate for a Simple Real Genetic Algorithm that Optimizes Array Factors. ACES Journal, 2000, 15(2): 94–102.
32. DeJong KA. An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph.D. Dissertation, University of Michigan, MA, USA. 1975.
33. Fogarty TC. Varying the Probability of Mutation in the Genetic Algorithm. Proceedings of the Third International Conference on Genetic Algorithms, 1989 June 4–7; George Mason University, USA. ACM. pp. 104–109.
34. Grefenstette J. Optimization of Control Parameters for Genetic Algorithms. IEEE Transactions on Systems, Man and Cybernetics, 1986, 16(1): 122–128.
35. Schaffer JD, Caruana RA, Eshelman LJ and Das R. A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization. Proceedings of the Third International Conference on Genetic Algorithms, 1989 June 4–7; George Mason University, USA. ACM. pp. 51–60.
36. Alander JT. On Optimal Population Size of Genetic Algorithms. Proceedings of the CompEuro ’92 Conference on Computer Systems and Software Engineering, 1992 May 4–8; The Hague, Netherlands. IEEE. pp. 65–70.
37. Hesser J and Manner R. Towards an Optimal Mutation Probability for Genetic Algorithms. Proceedings of the 1st Workshop on Parallel Problem Solving from Nature, 1990 October 1–3. ACM. pp. 23–32.
38. Srinivas M and Patnaik LM. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. IEEE Transactions on Systems, Man and Cybernetics, 1994, 24(4): 656–667.
39. Mühlenbein H. How Genetic Algorithms Really Work: Mutation and Hill-climbing. Proceedings of the Second Conference on Parallel Problem Solving from Nature, 1992 September 28–30; Amsterdarn, North-Holland, pp. 15–25.
40. Bäck T and Khuri S. An Evolutionary Heuristic for the Maximum Independent Set Problem. Proceedings of the First Conference on Evolutionary Computation, 1994 June 27–29; Orlando, Florida, USA. IEEE. pp. 531–535.
41. Khuri S, Back T and Heitkotter J. The Zero/One Multiple Knapsack Problem and Genetic Algorithms. Proceedings of the 1994 ACM Symposium on Applied Computing, 1994 March 6–8; Phoenix, Arizona, USA. ACM. pp. 188–193.
42. Khuri S and Bäck T. An Evolutionary Heuristic for the Minimum Vertex Cover Problem. Genetic Algorithms within the Framework of Evolutionary Computation–Proc. of the KI-94 Workshop, 1994; Germany. Saarbrücken. pp. 86–90.
43. ManKF, Tang KS and Kwong S. Genetic Algorithms for Control and Signal Processing. 1st ed., Springer-Verlag London, 1997.
44. Keppel G. Design and Analysis: A Research’s Handbook. 2nd ed., New Jersey: Inc., Englewood Cliffs, 1982.
45. Shapiro SS and Wilk MB. An Analysis of Variance Test for Normality. Biometrika, 1965, 52(3–4): 591–611.
46. Pham DT, Kog E, Ghanbarzadeh A, Otri S, Rahim S and Zaidi M. The Bees Algorithm- A Novel Tool for Complex Optimization Problems. Proceedings of the Second International Virtual Conference on Intelligent Production Machines and Systems, 2006 July 13–17; Oxford. Elsevier. pp. 454–459.
47. Abbass HA. A Single Queen Single Worker Honey-Bees Approach to 3-SAT. The Genetic and Evolutionary Computation Conference, 2001 July 7–11; San Francisco, CA, USA. pp. 807–814.
48. Lu T and Zhu J. Genetic Algorithm for Energy-Efficient QoS Multicast Routing. IEEE Communications Letters, 2013, 17(1): 31–34.