

## Research Article

# Multiconstrained QoS Routing Using a Differentially Guided Krill Herd Algorithm in Mobile Ad Hoc Networks

D. Kalaiselvi<sup>1</sup> and R. Radhakrishnan<sup>2</sup>

<sup>1</sup>Computer Science & Engineering, Jay Shriram Group of Institutions, Tiruppur 638 660, India

<sup>2</sup>Sri Shakthi Institute of Engineering & Technology, Coimbatore 641 062, India

Correspondence should be addressed to D. Kalaiselvi; [kalai.tabu@yahoo.com](mailto:kalai.tabu@yahoo.com)

Received 6 February 2014; Revised 29 May 2014; Accepted 28 July 2014

Academic Editor: Albert Victoire

Copyright © 2015 D. Kalaiselvi and R. Radhakrishnan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a multiconstrained quality-of-service routing (QoSR) based on a differentially guided krill herd (DGKH) algorithm in mobile ad hoc networks (MANETs). QoSR is a NP-complete and significant problem with immense challenges in determining an optimum path that simultaneously satisfies various constraints in MANETs, where the topology varies constantly. Various heuristic algorithms are widely used to solve this problem with compromise on excessive computational complexities and/or low performance. This paper proposes a krill herd based algorithm called DGKH, where the krill individuals do not update the position in comparison with one-to-one (as usual), but instead it uses the information from various krill individuals and then searches to determine a feasible path. Experiment results on MANETs with different number nodes (routes) are considered with three constraints which are maximum allowed delay, maximum allowed jitter, and minimum requested bandwidth. It is demonstrated that the proposed DGKH algorithm is an effective approximation algorithm exhibiting satisfactory performance than the KHA and existing algorithms in the literature by determining an optimum path that satisfies more than one QoS constraint in MANETs.

## 1. Introduction

Mobile ad hoc networks (MANETs) are class of wireless communication networks without a fixed infrastructure. The MANET nodes do not provide reliable services and QoS (quality of service) guarantees as compared to other wireless networks such as WiFi, WiMAX, GSM, and CDMA [1]. The QoS parameters to be guaranteed for multimedia group communication are bandwidth, delay, packet loss rate, jitters, and bandwidth-delay product. QoS is one of the significant components to evaluate MANET performance since QoS restricts the bounds on bandwidth, delay, bandwidth delay product, jitter, and packet loss [2, 3]. The violation of these parameters degrades the overall performance of an application.

Its primary goal is to allocate network resources efficiently while the different QoS requirements are satisfied simultaneously. The QoS requirements can be classified into link constraints (e.g., band width), path constraints (e.g., end to end delay), and tree constraints (e.g., delay-jitter).

The interdependency and confliction among multiple QoS parameters, however, make the problem very difficult and NP-complete [4, 5]. For the reason that the QoS multicast routing problem is NP-complete, several heuristic algorithms have been deployed to address this problem. For example, simulated annealing (SA), a powerful global optimization procedure, was utilized to solve the QoSR problem [6]. Unlike other heuristic techniques SA involves single candidate to search for the solution that make the method inefficient while solving large dimension problems.

Subsequently another powerful population based technique called genetic algorithms (GAs) has been engaged to solve the QoSR problem [7–9]. GA assures a larger possibility of locating a global optimum by starting with multiple random search points and processing several candidate solutions simultaneously. Perhaps, it is always criticized that genetic algorithms encounters some faults such as lack of local search ability, premature convergence, and slow convergence speed. Similarly, due to the advent of swarm intelligence theory, a category of stochastic search methods

for solving optimization (GO) problems of NP nature, ant colony optimization (ACO), and particle swarm optimization (PSO) is introduced.

Similar to GA, these two methods were also shown to be efficient tools for solving the multicast routing problem [10–13]. In [10, 11] the QoS-multiple constrained optimization problem has been solved using ant colony optimization (ACO) algorithm. Here the authors examined the degree and delay-constrained broadcasting problem with minimum cost using an ant colony-based algorithm. After simulations and comparisons with previous ant algorithm, and to two GA-based algorithms, the obtained results exhibited the best cost but again with too long computation time.

Consequently PSO [12, 13] was also used to address the QoS problem. PSO can be easily implemented and is computationally inexpensive, since its memory and CPU speed requirements are low. PSO has been proved to be an efficient approach for many optimization problems with its fast convergence and simple steps to solve a problem. In spite of these merits, a known fact that PSO has a severe drawback in the update of its global best (gbest) particle, which has a crucial role of guiding the rest of the particles, this leads to overcoming better solutions in the course of search.

In row of these heuristics, another new metaheuristic called Bees Life Algorithm (BLA) is proposed [14]. As one of the specie colony optimization, BLA is considered as a swarm intelligence algorithm and an approximate optimization method which performs according to collaborative individual behaviors in the population. But still like other probabilistic optimization algorithms, BLA has inherent drawback of premature convergence or stagnation that leads to loss of exploration and exploitation capability. In line to this, other heuristics like tabu search [15] and harmony search algorithm [16] were proposed for QoS with minimum number of constraints.

In fact there is a growing body of literature in the application of emerging heuristics to solve the NP-hard QoS problem. This paper proposes a krill herd based algorithm for QoS in MANETS. Krill herd algorithm (KHA) is a recently developed powerful evolutionary algorithm proposed by Gandomi and Alavi [17]. The KHA is based on the herding behavior of krill individuals. Each krill individual modifies its position using three processes, namely, (1) movement induced by other individuals, (2) foraging motion, and (3) random physical diffusion. The motion induced by other individuals and foraging motion contain global and local strategies, respectively, which make KHA a powerful technique. However, sometimes, KHA is unable to generate global optimal solutions on some high-dimensional nonlinear optimization problems.

Thus this paper proposes a differentially guided krill herd (DGKH) algorithm where the krill individuals do not update the position in comparison with one-to-one (as usual) instead it uses the information from various krill individuals and then searches to determine a feasible path. Thus the induction stage uses the position information from various krill individuals at random, making the search exhaustive, and leads to quick search of better solutions.

The rest of the paper is organized as follows. In Section 2, the problem statement of QoS routing problem in MANETS is introduced. The overview of KHA is briefly explained in Section 3 with description of the proposed differential guiding mechanism. Section 4 presents the numerical experiments with a standard QoS multicast routing problem. A detailed discussion about the proposed method in theoretical perspective is detailed in Section 5. The paper is concluded in Section 6.

## 2. Problem Statement

Let  $G = (V, E)$  be an undirected graph representing the network topology, where  $V$  is the set of nodes and  $E$  is the set of links. The source node and the multicast destination node set containing  $r$  destinations are the routes for multicast [18]. The end-to-end QoS requirements are expressed as constraints from the source to the destinations and are given by the following.

The delay (DL) is the maximum delay along any branch of the tree. A penalty is added if no link is found between two nodes. Consider

$$D(p(s, t)) = \sum_{e \in p(s, t)} \text{delay}(e) + \sum_{n \in p(s, t)} \text{delay}(n). \quad (1)$$

The bandwidth (BW) of the multicast tree is the average bandwidth requirement of all the tree branches. Consider

$$B(p(s, t)) = \min(\text{bandwidth}(e)), \quad e \in p(s, t). \quad (2)$$

The delay-jitter (DJ) is the maximum delay variation along any branch of the tree. Consider

$$DJ(p(s, t)) = \sum_{e \in p(s, t)} \text{delay\_jitter}(e) + \sum_{n \in p(s, t)} \text{delay\_jitter}(n). \quad (3)$$

The packet loss rate is the average packet loss along the tree branches. The cutoff for PLR is set at 10%. If a branch is having more than 10% packet loss, then a penalty is assigned, such that the model tries to search a path less than 10% PLR:

$$PL(p(s, t)) = 1 - \prod_{n \in p(s, t)} (1 - \text{packet\_loss}(n)). \quad (4)$$

The above QoS criteria have to be fulfilled with an objective to minimize cost function of the multicast tree as given by

$$C(T(s, M)) = \sum_{e \in T(s, M)} \text{cost}(e) + \sum_{n \in T(s, M)} \text{cost}(n). \quad (5)$$

Thus the QoS routing problem is formulated as an optimization problem whose goal is to find a multicast tree  $T(s, M)$

such that the cost function  $C(T(s, M))$  is minimized subject to the constraints as follows:

$$\begin{aligned} & \text{Minimize } C(T(s, M)) \\ & \text{s.t. } D(p(s, T)) \leq \text{QD} \\ & \quad B(p(s, T)) \geq \text{QB} \\ & \quad \text{DJ}(p(s, T)) \leq \text{QDJ} \\ & \quad \text{PL}(p(s, T)) \leq \text{QPL}. \end{aligned} \quad (6)$$

Subsequently the fitness function is defined as the sum of the cost function and penalized constraints as follows:

$$\begin{aligned} & \text{Minimize } f(T(s, M)) \\ & \quad = f_c + \omega_1 f_b + \omega_2 f_d + \omega_3 f_{dj} + \omega_4 f_{pl}, \end{aligned} \quad (7)$$

where

$$\begin{aligned} f_c &= \cos t(T(s, M)), \\ f_b &= \sum_{t \in M} \max\{\text{QB} - B(p(s, t)), 0\}, \\ f_d &= \sum_{t \in M} \max\{D(p(s, t)) - \text{QD}, 0\}, \\ f_{dj} &= \sum_{t \in M} \max\{\text{DJ}(p(s, t)) - \text{QJ}, 0\}, \\ f_{pl} &= \sum_{t \in M} \max\{\text{PL}(p(s, t)) - \text{QPL}, 0\}, \end{aligned} \quad (8)$$

where QD is delay constraint, QB is bandwidth constraint, QDJ is delay-jitter constraint, and QPL is packet loss constraint. Also  $\omega_1, \omega_2, \omega_3, \omega_4$ , are the weights of bandwidth, delay, delay-jitter, and packet loss, respectively. In the next section, the solution methodology to solve the multicast QoS routing problem is explained in detail.

### 3. Proposed Methodology

**3.1. Krill Herd Algorithm: An Overview.** Krill herd algorithm (KHA) is a recently developed heuristic algorithm based on the herding behavior of krill individuals. It has been first proposed by Gandomi and Alavi [17]. It is a population based method consisting of a large number of krill in which each krill moves through a multidimensional search space to look for food. In this optimization algorithm, the positions of krill individuals are considered as different design variables and the distance of the food from the krill individual is analogous to the fitness value of the objective function. In KHA, the individual krill alters its position and moves to the better positions. The movement of each individual is influenced by the three processes, namely, (i) induction process, (ii) foraging activity, and (iii) random diffusion. These operators are briefly explained and mathematically expressed as follows.

(i) *Induction.* In this process, the velocity of each krill is influenced by the movement of other krill individuals of the

multidimensional search space and its velocity is dynamically adjusted by the local, target, and repulsive vector. The velocity of the  $i$ th krill at the  $m$ th movements may be formulated as follows [17]:

$$v_i^m = \alpha_i v_i^{\max} + \omega_n v_i^{m-1}, \quad (9)$$

$$\begin{aligned} \alpha_i &= \sum_{j=1}^{N_s} \left[ \frac{f_i - f_j}{f_w - f_b} \times \frac{Z_i - Z_j}{|Z_i - Z_j| + \text{rand}(0, 1)} \right] \\ &+ 2 \left[ \text{rand}(0, 1) + \frac{i}{i_{\max}} \right] f_i^{\text{best}} Z_i^{\text{best}}, \end{aligned} \quad (10)$$

where  $V_i^{\max}$  is the maximum induced motion;  $V_i^m$  and  $V_i^{m-1}$  are the induced motion of the  $i$ th krill at the  $m$ th and  $(m-1)$ th movement;  $\omega_n$  is the inertia weight of the motion induced;  $f_w$  and  $f_b$  are the worst and the best position, respectively, among all krill individuals, of the population;  $f_i, f_j$  are the fitness value of the  $i$ th and  $j$ th individuals, respectively;  $N_s$  is the number of krill individuals surrounding the particular krill;  $i$  and  $i_{\max}$  are the current iteration and the maximum iteration number.

A sensing distance ( $SD_i$ ) parameter is used to identify the neighboring members of each krill individual. If the distance between the two krill individuals is less than the sensing distance, that particular krill is considered as neighbor of the other krill. The sensing distance may be represented by [17]

$$SD_i = \frac{1}{5n_p} \sum_{k=1}^{n_p} |Z_i - Z_k|, \quad (11)$$

where  $n_p$  is the population size;  $Z_i$  and  $Z_k$  are the position of the  $i$ th and  $k$ th krill, respectively.

(ii) *Foraging Action.* Each individual krill updates its foraging velocity according to its own current and previous food location. The foraging velocity of the  $i$ th krill at the  $m$ th movement may be expressed by [17]

$$\begin{aligned} V_{f_i}^m &= 0.02 \left[ 2 \left( 1 - \frac{i}{i_{\max}} \right) f_i \frac{\sum_{k=1}^{N_s} (Z_k / f_k)}{\sum_{k=1}^{N_s} (1 / f_k)} + f_i^{\text{best}} X_i^{\text{best}} \right] \\ &+ \omega_x V_{f_i}^{m-1}, \end{aligned} \quad (12)$$

where  $\omega_x$  is the inertia weight of the foraging motion;  $V_{f_i}^{m-1}$  and  $V_{f_i}^m$  are the foraging motion of the  $i$ th krill at the  $(m-1)$ th and  $m$  movement.

(iii) *Random Diffusion.* In KHA algorithm, in order to enhance the population diversity, random diffusion process is incorporated in krill individuals. This process maintains or increases the diversity of the individuals during the whole optimization process. The diffusion speed of krill individuals may be expressed as follows [17]:

$$v_{D_i}^m = \mu v_{D_i}^{\max}, \quad (13)$$

where  $V_D^{\max}$  is the maximum diffusion speed;  $\mu$  is a directional vector uniformly distributed between  $(-1, 1)$ .

(iv) *Position Update.* In KHA, the krill individuals fly around in the multidimensional space and each krill adjusts its position based on induction motion, foraging motion, and diffusion motion. In this way, KHA combines local search with global search for balancing the exploration and exploitation. The updated position of the  $i$ th krill may be expressed as [17]

$$Z_i^{m+1} = Z_i^m + (V_i^m + V_{f_i}^m + V_{D_i}^m) P_t \sum_{j=1}^{N_d} (u_j - l_j), \quad (14)$$

where  $N_d$  is the number of control variables;  $u_j$  and  $l_j$  are the maximum and minimum limits of the  $j$ th control variable;  $P_t$  is the position constant factor. The above procedure will be used to optimize (7) for MANET routing optimization.

**3.2. Differentially Guided Procedure.** This procedure is incorporated in the movement induced by other krill individuals. In the original KHA, the velocity of the  $i$ th krill at the  $m$ th run is given in (10). The vectors show the induced directions by different neighbors and each value presents the effect of a neighbor. The neighbors' vector can be attractive or repulsive since the normalized value can be negative or positive. For choosing the neighbor, in original KHA a sensing distance ( $SD_i$ ) is adopted. Instead of such choice, where the krill individuals update the position in comparison with one-to-one (as in original KHA), this research proposes a new mechanism whence the induction stage uses the position information from various krill individuals at random.

**Differential Guiding Mechanism.** All krill individuals can generate new positions in the search space using the information derived from different krill individuals' using best information. To ensure that a krill individual learns from good exemplars and to minimize the time wasted on poor directions, we allow the krill individual to learn from the exemplars until the krill individual ceases improving for a certain number of generations called the refreshing gap. We observe three main differences between the DGKH algorithm and the original KHA [17].

- (1) Once the sensing distance is used to identify the neighboring members of each krill individual, as exemplars to update the position, this mechanism utilizes the potentials of all krill individuals as exemplars to guide a krill individual's new position.
- (2) Instead of learning from the same exemplar krill individual's for all dimensions, each dimension of a krill individual in general can learn from different krill individuals for different dimensions to update its position. In other words, each dimension of a krill individual may learn from the corresponding dimension of different krill individual based on the proposed equation (15).

- (3) Finding the neighbor for different dimensions to update a krill individual position is done randomly (with a vigil that repetitions are avoided). This improves the thorough exploration capability of the original KHA with large possibility to avoid premature convergence in complex optimization problems.

Compared to the original KHA, DGKH algorithm searches more promising regions to find the global optimum. The difference between KHA and DGKH is that the differential operator applied to only accept the basic KHA generating new better solution for each krill instead of accepting all the krill updating adopted in KH. This is rather greedy. The original KHA is very efficient and powerful but is highly prone to premature convergence. Therefore, to evade premature convergence and further improve the exploration ability of the original KHA, a differential guidance is used to tap useful information in all the krill individuals to update the position of a particular krill individual. The following expresses the differential mechanism:

$$Z_i - Z_j = (z_{i1} \ z_{i2} \ z_{i3} \ \cdots \ z_{in}) - (z_{\rho 1} \ z_{\rho 2} \ z_{\rho 3} \ \cdots \ z_{\rho n}), \quad (15)$$

where  $z_{i1}$  is the first element in the  $n$  dimension vector  $Z_i$ ,  $z_{in}$  is the  $n$ th element in the  $n$  dimension vector  $Z_i$ ,  $z_{\rho 1}$  is the first element in the  $n$  dimension vector  $Z_\rho$ , and  $\rho$  is the random integer generated separately for each  $z$ , between 1 to  $n$ , but  $\rho \neq i$ .

Figure 1 shows the differential mechanism for choosing the neighbor krill individual for (10). This assumes that the dimension of the problem as 5 and assumes krill size as 6 are in the search. Once the neighbor krill individuals are identified using the sensing distance, the  $i$ th individual position will be updated (with all neighbor krill individuals) as shown in Figure 1. This is in an effort to avoid premature convergence and explore a large promising region in the prior run phase to search the whole space extensively.

## 4. Experimental Analysis

Numerical Simulations were carried out to validate the performance of the proposed DGKH based QoS algorithm and experimental analysis is performed on a test network [12] as shown in Figure 2, on three different cases. For Case (1) constraint Set1, delay constraint QD = 20, delay-jitter constraint QDJ = 30, and bandwidth constraint QB = 40. In Case (2) constraint Set2, delay constraint QD = 25, delay-jitter constraint QDJ = 35, and bandwidth constraint QB = 40. In Case (3) in addition to the constraints Set 2, packet loss rate constraints are also taken into account. For the packet loss rate constraint, we set QPL = 30 as used in [4], since ad hoc networks are high mobility networks with limited resources. Also considering all these parameters can indicate timeliness and precision which is used to measure the output performances of a routing algorithm.

First two cases are studied to establish the superiority of the proposed DGKH based QoS algorithm in comparison to the existing QPSO method. The proposed DGKH algorithm was

	ith individual						jth individual							$Z_i - Z_j$
Dimension	$Z_{11}$					$Z_{11}$	$Z_{12}$	$Z_{13}$	$Z_{14}$	$Z_{15}$	$Z_{16}$	$Z_{11} - Z_{13}$		
	$Z_{21}$					$Z_{21}$	$Z_{22}$	$Z_{23}$	$Z_{24}$	$Z_{25}$	$Z_{26}$	$Z_{21} - Z_{22}$		
	$Z_{31}$					$Z_{31}$	$Z_{32}$	$Z_{33}$	$Z_{34}$	$Z_{35}$	$Z_{36}$	$Z_{31} - Z_{36}$		
	$Z_{41}$					$Z_{41}$	$Z_{42}$	$Z_{43}$	$Z_{44}$	$Z_{45}$	$Z_{46}$	$Z_{41} - Z_{45}$		
	$Z_{51}$					$Z_{51}$	$Z_{52}$	$Z_{53}$	$Z_{54}$	$Z_{55}$	$Z_{56}$	$Z_{51} - Z_{54}$		

FIGURE 1: Differential mechanism illustrated.

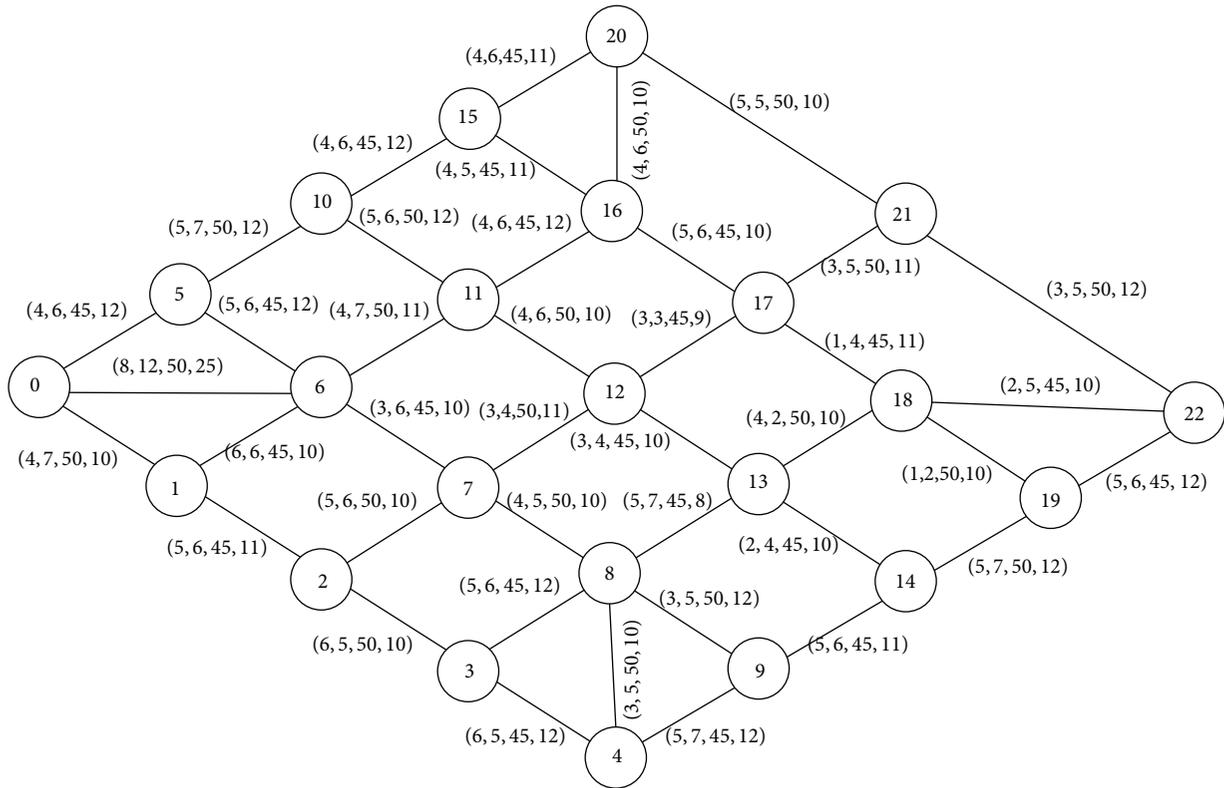


FIGURE 2: Network used as test bench for Cases 1 and 2.

executed for 30 trials on each case. The performance of the proposed algorithm from the perspectives of the best multicast tree obtained, best fitness value, average fitness value, convergence speed, and so forth is demonstrated.

The multicast tree with the best fitness value of Cases 1 and 2 that satisfies the constraints estimated out of 30 runs of each algorithm is shown in Figure 3. The least cost, delay, and delay-jitter of each identified multicast tree are summarized in Table 1.

In addition the mean and standard deviation of the best fitness values in 30 trial runs of all algorithms under consideration are also listed. The averaged values are obtained for delay and delay-jitter over all the paths of the multicast tree from the source node to the end nodes. It is proved that for both cases of the constraint sets, the best fitness value of the multicast trees generated by both QPSO [12] and proposed DGKH have the same least costs with the

considered constraints, informing that DGKH algorithm could guarantee better multicast tree than QPSO, PSO and GA [12].

To demonstrate the convergence properties of the algorithms under considerations, the convergence graph of both KHA and DGKH algorithm on the problem is experimented and plotted. Since this research did not implement other algorithms which are used for comparison, the KHA and DGKH algorithms are considered. Figure 3 traces the dynamic changes of cost, average delay, and average delay-jitter with the development of iteration for two algorithms, that is, QPSO [12] and the proposed DGKH algorithm. As evident from Figure 3, the convergence speed of DGKH is better. It conveys that proposed DGKH algorithm has stronger search ability than QPSO on the tested problem.

To validate the proposed algorithm with additional constraints as mentioned in Case (3), the network under

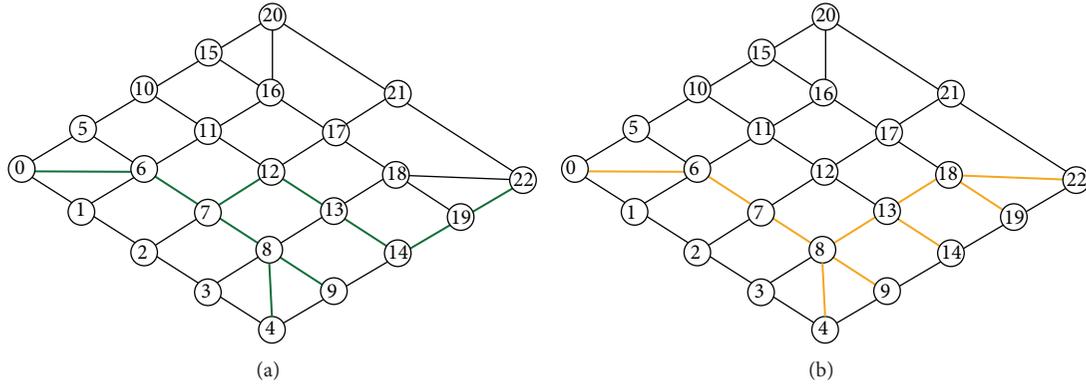


FIGURE 3: Best multicast tree generated by DGKH algorithm constraint, (a) Set 1 and (b) Set 2.

TABLE 1: Summary of performance metrics by various methods for multiconstrained QoS for various cases.

Index	Methods	Best cost	Average delay	Average delay-jitter	Average cost	Standard deviation	Average packet loss rate
Case 1 [12]	GA	138	18.5	30.2	145.29	3.73	NA
	PSO	140	19.6	30.8	143.83	3.59	NA
	QPSO	122	21.6	33.0	135.20	3.25	NA
	DGKH	122	22.1	33.3	125.37	3.20	NA
Case 2 [12]	GA	134	21.8	31.2	142.93	3.26	NA
	PSO	131	20.4	31.6	138.67	2.80	NA
	QPSO	127	21.8	32.2	131.89	2.19	NA
	DGKH	127	21.3	32.4	129.21	2.16	NA
Case 3	KHA	135	21.8	32.2	131.89	2.19	0.9442
	DGKH	127	21.3	32.4	129.21	2.16	0.9370

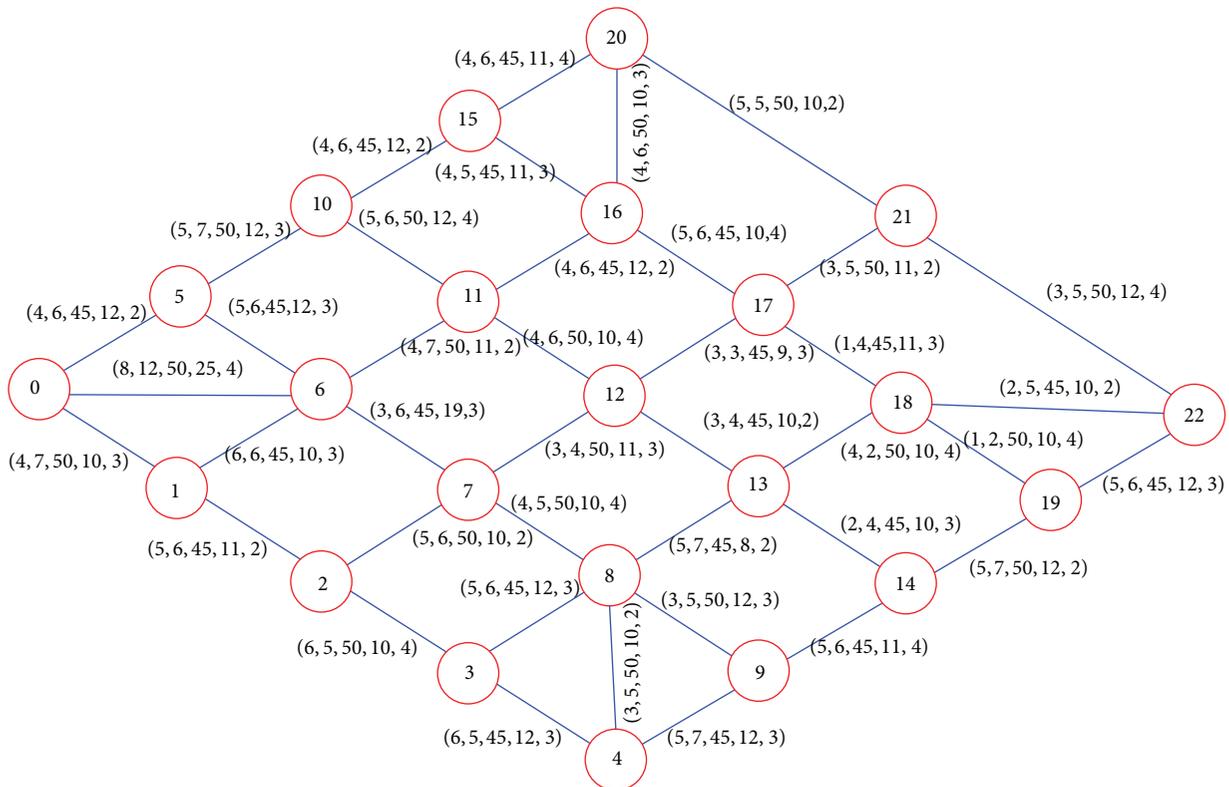


FIGURE 4: Network used as test bench for Case 3.

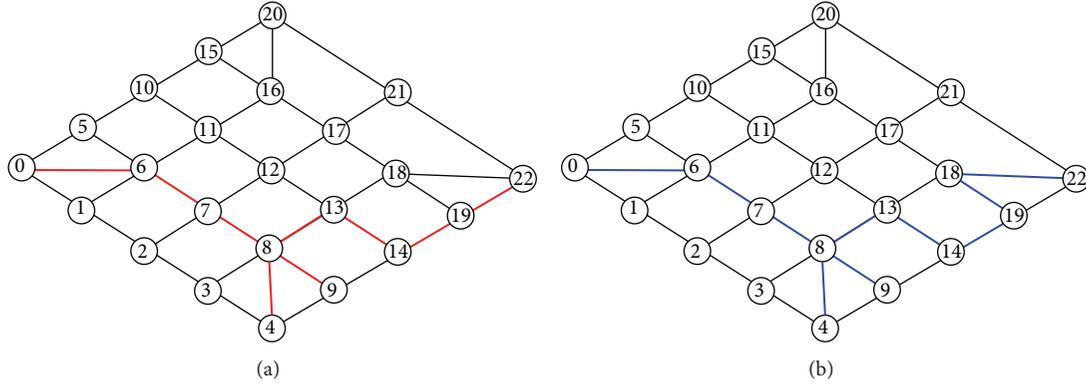


FIGURE 5: Best multicast tree generated by DGKH algorithm constraint, (a) Set 1 and (b) Set 2 for Case 3.

TABLE 2: Network data.

Parameter	Range
Cost of link	2–10
Bandwidth	50–200 kb/s
Delay-jitter	5–15 ms
Bandwidth constraint	100–170
Delay constraint	60–180
Delay-jitter constraint	20–50

consideration is amended by including the packet loss rate constraint in the last part of the representation. Thus the features of the edges in the proposed network, as shown in Figure 4, are arranged by a pentuple (D, DJ, B, C, PL) with the components representing delay, delay-jitter, band width, cost, and packet loss, respectively. The multicast tree as shown in Figure 5, with the best fitness value of Case 3, that satisfies the constraints estimated out of 30 runs of each algorithm, is shown in Figure 5. The least cost, delay, delay-jitter, and packet loss of each identified multicast tree for Case 3 are also summarized in Table 1.

### 5. Experiments on Scalability

In this section, numerical simulations were conducted in order to investigate the scalability of the proposed DGKH algorithm. To achieve this objective, we used the network topologies created by Salama stochastic network topology generation [4, 12]. The data is adopted from [12] for comparison purpose and is shown in Table 2. Here the packet loss rate constraints are relaxed as the purpose of this section is to demonstrate the scalability of the proposed DGKH algorithm for QoSR. The number of nodes of each network was set randomly from 10 to 100. Table 3 summarizes the results of the proposed algorithm. Due to the space limitation, we only show in Figure 6 the convergence properties of the algorithms averaged over 30 runs on the problem under constraint Set 2 for Case 3.

The other parameters of each algorithm were configured as in the first group of experiments. The algorithms

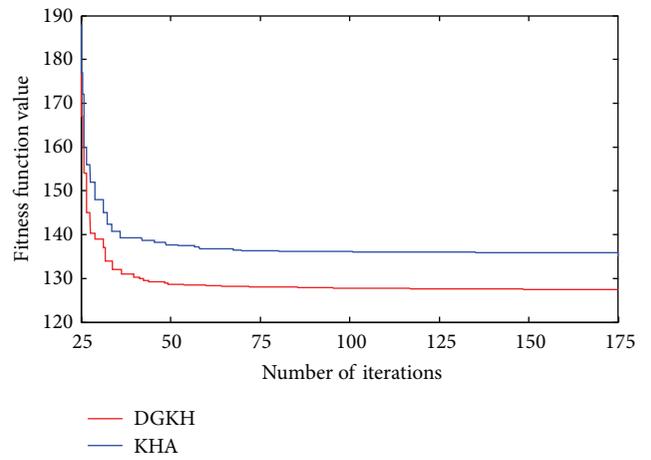


FIGURE 6: Convergence plot for the KHA and DGKH algorithm while optimizing the fitness function.

were evaluated from angles of least cost of obtained multicast tree, execution time, and routing request success ratio.

By simulation, scalability of the algorithm with the problem size was evaluated according to execution time and least cost of the obtained multicast tree. For comparison purpose the plots of the proposed DGKH algorithm is merged with the plots of the QPSO method. We did not implement the QPSO at any part of this research. The values of QPSO [12] are taken from the simulation software. Figure 7 shows the least cost generated by each algorithm that is averaged over different constraints for the problems with different numbers of network nodes. It can be observed that the multicast tree obtained by DGKH on each problem has best fitness value compared to the QPSO method; establishing it can generate the higher-quality solutions on the multicast routing problems in MANETS. Figure 8 shows the execution time of the two algorithms in comparison to the problems with different numbers of network nodes. The results convey the superiority of the proposed method, as the QPSO is more time consuming than DGKH on the problems with the same network scale. In summary both the DGKH and the QPSO,

TABLE 3: Performance metrics on the random networks of 10–100 nodes by the proposed DGKH algorithm.

Nodes	Best cost	Average delay	Average delay-jitter	Average cost	Standard deviation
10	148.25	21.3	34.2	148.89	3.52
20	301.41	21.2	34.6	302.63	4.21
30	480.36	21.2	34.2	481.52	3.96
40	562.47	21.5	34.1	563.31	4.52
50	775.23	21.4	33.9	776.24	5.01
60	925.62	21.3	33.9	926.22	3.67
70	1052.31	21.3	34.5	1053.54	4.23
80	1218.32	21.1	34.2	1218.89	3.87
90	1384.95	21.1	34.1	1385.62	5.31
100	1572.34	21.2	34.3	1573.05	6.54

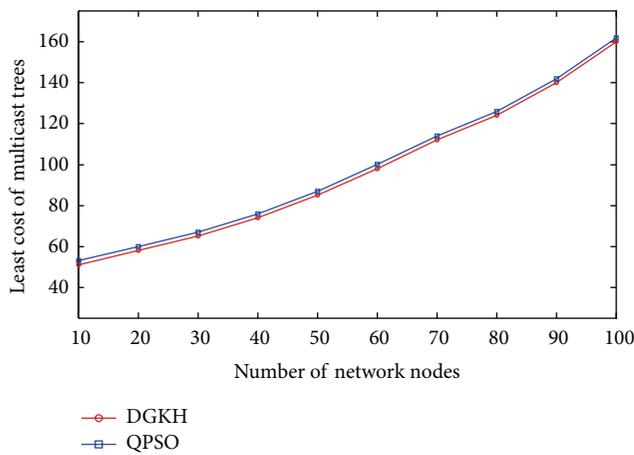


FIGURE 7: The average cost in 30 runs of each algorithm on the problems with different number of network nodes.

algorithms for QoS in MANETS, seem to consume almost the same amount of simulation time on the problem with the same scale. In conclusion, when comparing two algorithms for OoS in multicast routing problem of MANETS, DGKH method could generate best multicast trees (with best fitness (cost) values) compared to QPSO. Additionally to find out the solutions with same quality, the DGKH algorithm has consumed (comparatively) least computational time among its contestant algorithm QPSO.

The performance of the algorithm can also be evaluated by the routing request success ratio  $\xi_{req}$  given by

$$\xi_{req} = \frac{N_{ack}}{N_{req}}, \quad (16)$$

where  $N_{ack}$  is the number of successful routing requests and  $N_{req}$  is the number of all routing requests.

Successful routing request means that the multicast tree obtained by the algorithm satisfies the bandwidth, delay, and delay-jitter constraints [18]. It is, however, too complicated to compare all of the QoS parameters synchronously. Therefore, the performance is verified by each of QoS parameters. The comparison of routing request success ratios among DGKH

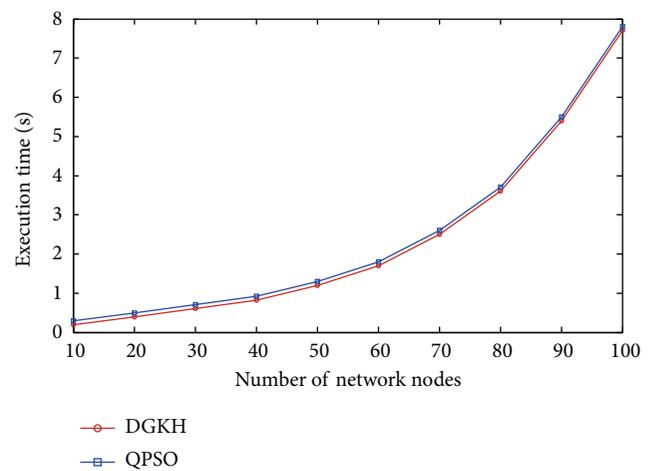


FIGURE 8: The execution time of each algorithm on the problems with different number of network nodes.

and QPSO is curved and shown in Figure 9. It is proved that DGKH has the highest success ratio according to each QoS parameter and thus has the best convergence performance.

## 6. Conclusion

A QoS multicast routing method based on a modified krill herd algorithm is proposed in this paper. Searching for the feasible/optimal route after simultaneously satisfying more than one QoS constraint in wireless mobile ad hoc networks is classified as an NP-complete problem. This research considers four different constraints while optimizing the total cost of the multicast tree. Hence the original krill herd algorithm is refined to perform the thorough exploration of the search space by differentially guided mechanism to update their krill individual positions. Experimental result reveals the proposed that DGKH algorithm can search the solution space in a very effective and efficient manner. Simulation results also shows that the proposed algorithm can search the better multicast trees with rapid convergence speed effectively compared with other algorithms.

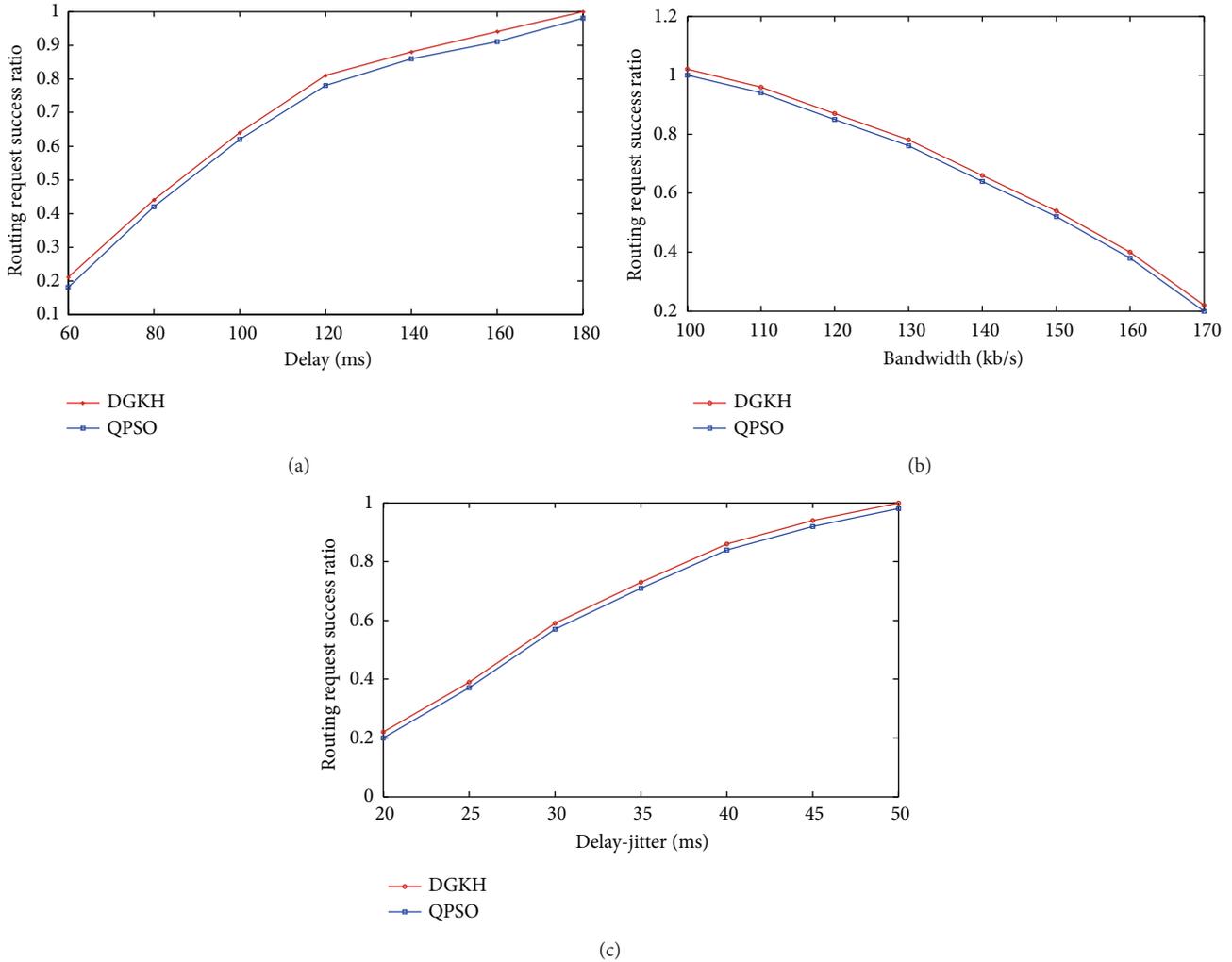


FIGURE 9: Comparisons of routing request success ratio. (a) Delay, (b) bandwidth, and (c) delay-jitter.

As a future work, we plan to consider other heuristic techniques for estimation of multicast routing considering other constraints like queuing delays, propagation error, maximum link utilization, bandwidth consumption parameters, blocking probability, and so forth as multiobjective optimization problem.

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**References**

- [1] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicast routing for multimedia communication," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 286–292, 1993.
- [2] W.-L. Yang, "Optimal and heuristic algorithms for quality-of-service routing with multiple constraints," *Performance Evaluation*, vol. 57, no. 3, pp. 261–278, 2004.
- [3] L. Junhai, X. Liu, and Y. Danxia, "Research on multicast routing protocols for mobile ad-hoc networks," *Computer Networks*, vol. 52, no. 5, pp. 988–997, 2008.
- [4] H. F. Salama, Y. Director-Viniotis, and D. S. Director-Reeves, *Multicast Routing for Real-Time Communication of High-Speed Networks*, North Carolina State University, Raleigh, NC, USA, 1996.
- [5] K. Kunavut and T. Sanguankotchakorn, "Multi-Constrained Path (MCP) QoS routing in OLSR based on multiple additive QoS metrics," in *Proceedings of the International Symposium on Communications and Information Technologies (ISCIT '10)*, pp. 226–231, IEEE, October 2010.
- [6] L. Liu and G. Feng, "Simulated annealing based multi-constrained QoS routing in mobile ad hoc networks," *Wireless Personal Communications*, vol. 41, no. 3, pp. 393–405, 2007.
- [7] C. Lin and C. Chuang, "A rough penalty genetic algorithm for multicast routing in mobile ad hoc networks," *Journal of Applied Mathematics*, vol. 2013, Article ID 986985, 11 pages, 2013.
- [8] F. Xiang, L. Junzhou, W. Jiyei, and G. Guanqun, "QoS routing based on genetic algorithm," *Computer Communications*, vol. 22, no. 15, pp. 1392–1399, 1999.
- [9] C. Li, C. Cao, Y. Li, and Y. Yu, "Hybrid of genetic algorithm and particle swarm optimization for multicast QoS routing," in *Proceedings of the IEEE International Conference on Control and Automation (ICCA '07)*, pp. 2355–2359, IEEE, June 2007.

- [10] D. Rupérez Cañas, A. L. Sandoval Orozco, L. J. García Villalba, and P.-S. Hong, "Hybrid ACO routing protocol for mobile Ad Hoc networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 265485, 7 pages, 2013.
- [11] H. Wang, H. Xu, S. Yi, and Z. Shi, "A tree-growth based ant colony algorithm for QoS multicast routing problem," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11787–11795, 2011.
- [12] J. Sun, W. Fang, X. Wu, Z. Xie, and W. Xu, "QoS multicast routing using a quantum-behaved particle swarm optimization algorithm," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 123–131, 2011.
- [13] H. Wang, X. Meng, S. Li, and H. Xu, "A tree-based particle swarm optimization for multicast routing," *Computer Networks*, vol. 54, no. 15, pp. 2775–2786, 2010.
- [14] S. Bitam and A. Mellouk, "Bee life-based multi constraints multicast routing optimization for vehicular ad hoc networks," *Journal of Network and Computer Applications*, vol. 36, no. 3, pp. 981–991, 2013.
- [15] N. Ghaboosi and A. T. Haghghat, "Tabu search based algorithms for bandwidth-delay-constrained least-cost multicast routing," *Telecommunication Systems*, vol. 34, no. 3-4, pp. 147–166, 2007.
- [16] R. Forsati, A. T. Haghghat, and M. Mahdavi, "Harmony search based algorithms for bandwidth-delay-constrained least-cost multicast routing," *Computer Communications*, vol. 31, no. 10, pp. 2505–2519, 2008.
- [17] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.
- [18] L. Zhang, L.-B. Cai, M. Li, and F.-H. Wang, "A method for least-cost QoS multicast routing based on genetic simulated annealing algorithm," *Computer Communications*, vol. 32, no. 1, pp. 105–110, 2009.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

