Ad Hoc Networks 58 (2017) 213–221

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

A micro-artificial bee colony based multicast routing in vehicular ad hoc networks

Xiu Zhang[a,b], Xin Zhang[a,b,*], Cheng Gu[a,b]

[a] College of Electronic and Communication Engineering, Tianjin Normal University, Tianjin, China
[b] Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University, Tianjin, China

ARTICLE INFO

Article history:
Received 12 April 2016
Revised 16 June 2016
Accepted 18 June 2016
Available online 20 June 2016

Keywords:
Vehicular ad hoc network
Artificial bee colony
Multicast routing
Quality of service
Steiner tree

ABSTRACT

Vehicular ad hoc networks (VANETs) have drawn great attention in wireless communications. Prompt and reliable vehicular communication is a must to provide a good service. Routing is the key problem in information transmission of VANETs. This paper studies quality of service (QoS) constrained multicast routing problem. This problem has been proved to be NP-complete problem, and swarm intelligence algorithms are more suitable than classical algorithms. A micro artificial bee colony (MABC) algorithm is proposed to deal with the problem. The QoS constraints include maximize network lifetime and minimizing delay cost. Multicast routing is abstracted to a continuous optimization problem. Then, it is linked with MABC. Numerical simulation is implemented on a traffic scenario with three instances. Results show that the MABC algorithm successfully attains the optimal routes. Moreover, the routing framework can be applied in real time given the network structure does not change too frequently.

1. Introduction

Hybrid wireless network is formed by wireless nodes and base stations [1]. Wireless networks without support from the fixed infrastructure are known as ad hoc networks. Due to the lack of infrastructure, the data is forwarded to the destination via a multi-hop fashion. Quite often, the ad hoc network has been studied in optimization [2], target detection [3,4], etc. In some scenarios, a set of base stations are connected by wired links and placed within the ad hoc networks to form a wired infrastructure, aiming to enhance the whole network performance. This resulting network is referred to as a hybrid wireless network. Due to the dynamic nature of such network, quite often computational intelligence approaches such as fuzzy logic systems [5–7] and evolutionary computing could be applied to optimization in hybrid wireless networks. In this paper, we are interested in applying evolutionary computing (artificial bee colony) to Vehicular ad Hoc Network (VANET).

Vehicular ad hoc networks (VANETs) have drawn great attention in wireless communications. Typical application scenarios of VANETs include military communications, where base stations could not be built ahead of war area; traffic status reports, where car traffic happens in rush hour; emergency services, where a temporary network is established to assure the communication of relief workers and medical staff; sensor networks, which connect sensors and control center. In VANETs, vehicles are able to arrange themselves to fulfill the application requirements based on the current situations. To support information transmission among vehicles, wireless communications is clearly the primary method. In VANETs, each vehicle is assumed to contain necessary equipments to communicate with nearby vehicles in a short distance. Long distance communication in VANETs is usually stuck in some signal propagation effects, which could be overcome by multi-hop communication. Due to the mobility feature of vehicles and the lack of fundamental architectures, the real time status of networks is hardly to acquire. On the other hand, the connectivity in the networks is an essential basis for information exchange and application requirements. Thus, routing is the key problem in ad hoc network fields. A good routing protocol presents reliable performance in receiving and sending messages from a source node to a destination node [8]. Because multicast routing could effectively organize network resources, reduce network congestion, and node work load, researches concerning multicast routing is meaningful and valuable in VANETs.

VANETs are often expressed as a graph $G = (N, E, W)$, where N is a set of vertex, $W = \sum_{e\in E} w_e$ is weight function of defined on edge set E. Based on graphic theory, the design of multi-hop routing has recently received great attention. In [9], the routing

* Corresponding author.
E-mail addresses: zhang210@126.com (X. Zhang), tjnumark@126.com (X. Zhang), 15620270595@163.com (C. Gu).

http://dx.doi.org/10.1016/j.adhoc.2016.06.009
1570-8705/

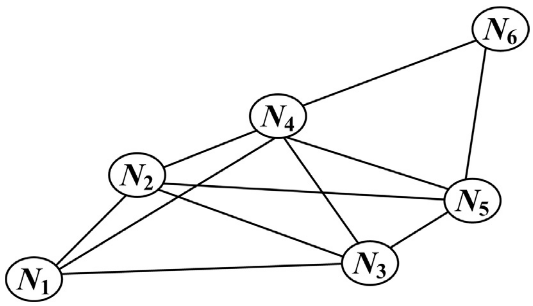


Fig. 1. Example of initializing a binary string.

algorithms and protocols are classified into three types: connected dominating set, disjoint sets, and Steiner minimum spanning tree. Most of these problems are NP-hard or NP-complete.

Assume a VANET has been transformed to a graph G with n nodes and $C > 0$ is a constant number, the Steiner minimum tree (SMT) problem refers to add m new nodes to G such that the resulting minimum spanning tree has the minimum number of additional nodes (i.e., m) and the edge length in the tree is equal to or less than C. The additional nodes are named Steiner points. This problem is also described as "Steiner tree problem with minimum number of Steiner points and bounded edge-length (STP-MSPBEL) [10]". When the manhattan distance is used in the network, the SMT problem is known to be NP-hard.

Due to the failure of sensor nodes, fast moving vehicles, or vehicle breakdown, VANETs require a routing path between source node and destiny node being frequently reconstructed. Transmission based on tree structure is the most popular one in multicast routing. Once the multicast tree is built, the information generated by source node can be sent to end nodes. Source-based tree and core-based tree (i.e., shared tree) are two common types in multicast tree. Routing algorithm is a crucial part of routing protocol, which is in charge of constructing a tree linking source nodes and destiny nodes.Quality of service (QoS) is a must when routing algorithm builds pathes. QoS mainly contains path length, bandwidth, delay, delay jitter and packet loss ratio.

As quality of service (QoS) constrained multicast routing has been proved to be NP-complete problem [11,12], classical routing algorithms become unable to handle this problem. Recent researches focus on heuristic algorithms and computational intelligence algorithms. Typical computational intelligence paradigms include particle swarm optimization [13], differential evolution [14], neighborhood field optimization [15]. Artificial bee colony (ABC) is a popular computational intelligence algorithm. This paper concentrates on modifying ABC to tackle the SMT problem in multicast routing.

To efficiently deal with the SMT problem, this paper designs a micro-ABC method with binary representation. The method contains a micro bee colony, which saves computational time in each cycle compared with the use of a regular colony size. Moreover, two novel search equations are proposed to improve the convergence speed of the algorithm. The performance of the algorithm is studied on a VANET with 16 cars.

The paper is organized as follows. Section 2 reports the SMT problem and related works. Section 3 gives the proposed algorithm and its analysis. Section 4 shows the numerical simulation setting and results. Section 5 gives the conclusion.

2. Problem overview and related works

VANET is featured with exchanging information amongst vehicles in real time. It requires data packets have to travel through the vehicular network from source nodes to destiny nodes. Routing protocol is crucial in the operation of VANET. Information transmission between a source node and an end node is easily resolved by shortest path algorithm given the network topology. However, the transmission task in VANET are mainly a source node broadcasting to multiple end nodes, which is well known as multicast routing.

Based on the implementation of multicast routing, the algorithms can be classified to centralized algorithm and distributed algorithm. In the former, source nodes is in charge of find a proper route based on the status information that it acquires; while the latter requires each node has local status information instead of mastering the whole network status, and the computation of route is accomplished by inter-sites on the route. Neither could completely outperform the other. Users have to decide which one is more proper depending on the practical necessity.

The information transmission in multicast communication is realized by building a multicast tree. Among all possible multicast trees, the most economic one evaluated by QoS indices is called Steiner minimum tree. The SMT problem is described as follows. Let $G = (N, E, W)$ denote an undirected graph, where $W = \sum_{e \in E} w_e$ is weight function defined on edge set E. Under the condition that auxiliary nodes are allowed to add to G, SMT is equivalent to find the minimum spanning tree of graph $\bar{G}$, where $\bar{G}$ is G with auxiliary node set $\bar{N}$ and updated edge set $\bar{N}$. Since the distance in VANETs is generally measured by Euclidean metric, the SMT problem is also called ESMT [11].

Kompella et al. proposed a heuristic multicasting routing for multimedia communications [16]. Sun and Langendorfer proposed a constrained Dijkstra heuristic algorithm for delay-constrained multicast routing algorithm [17]. Parsa et al. proposed a bounded shortest delay-constrained multicasting algorithm [18]. Gutierrez-Reina et al. studied a railway scenario, which is a kind of MANETs, and applied GA to optimize the network topology [19]. Natarajan and Rajendran proposed a modified Dijkstra algorithm to deal with an advanced optimized link state routing protocol [20].

Recently, researchers attempted to handle multicast routing by computational intelligence approaches [21]. Hwang et al. used genetic algorithm (GA) to deal with multicast routing [22]. Yen et al. considered multicast routing with multiple QoS constraints in mobile ad hoc networks (MANETs) and proposed an energy-efficient GA for this kind of problems [23]. Based on Tabu search, Forsati et al. studied several methods to tackle the bandwidth delay-constrained least-cost multicast routing [24]. Toutouh et al. studied the optimal parameter setting in the optimized link state routing protocol of MANETs, where GA, differential evolution (DE), particle swarm optimization, and simulated annealing are applied to do the optimization [25].

This paper attempts to tackle QoS multicast routing protocol in VANETs. Specifically, minimum cost and maximum network life time are taken as measurements to access the QoS of network. SMT is constructed linking source nodes and destiny nodes. In multicast routing, it is usually assumed that one source node send messages to multiple end nodes. Unlike building a network architecture, Steiner points are not permitted to be placed anywhere under the coverage area of a vehicle. At a time slot, the positions of vehicles in VANETs are relatively stable, though they may change in the next time slot after driving. The candidate Steiner points are those in network graph G excluding source and destiny nodes. Fig. 1 presents a VANET example including 6 nodes and 11 edges. All nodes are numbered in order. Given node N_1 is the source and node N_5 is the end. The SMT problem becomes to select middle nodes from set $\{N_2, N_3, N_4, N_6\}$ such that the communication cost is minimized. Each node in the network can dynamically very its emitted energy. In case a node becomes a inter-site in a routing tree, we suppose that it could adapt its radiation energy to different transmission paths. Moreover, the topology of a VANET is

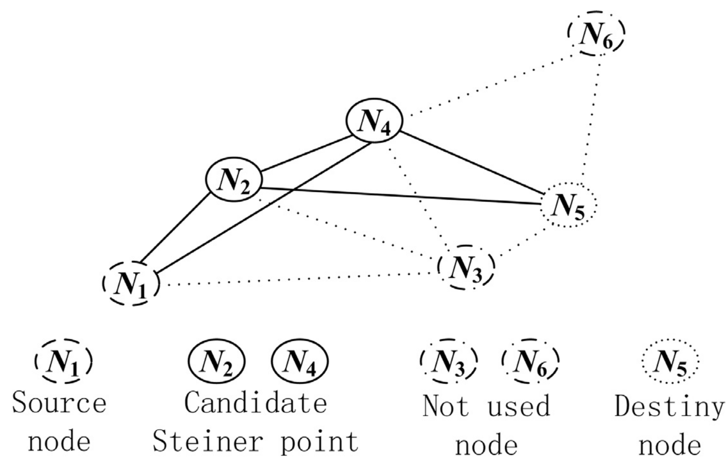


Fig. 2. Subgraph associated with bit string 1010 in the order of N_2, N_3, N_4, and N_6.

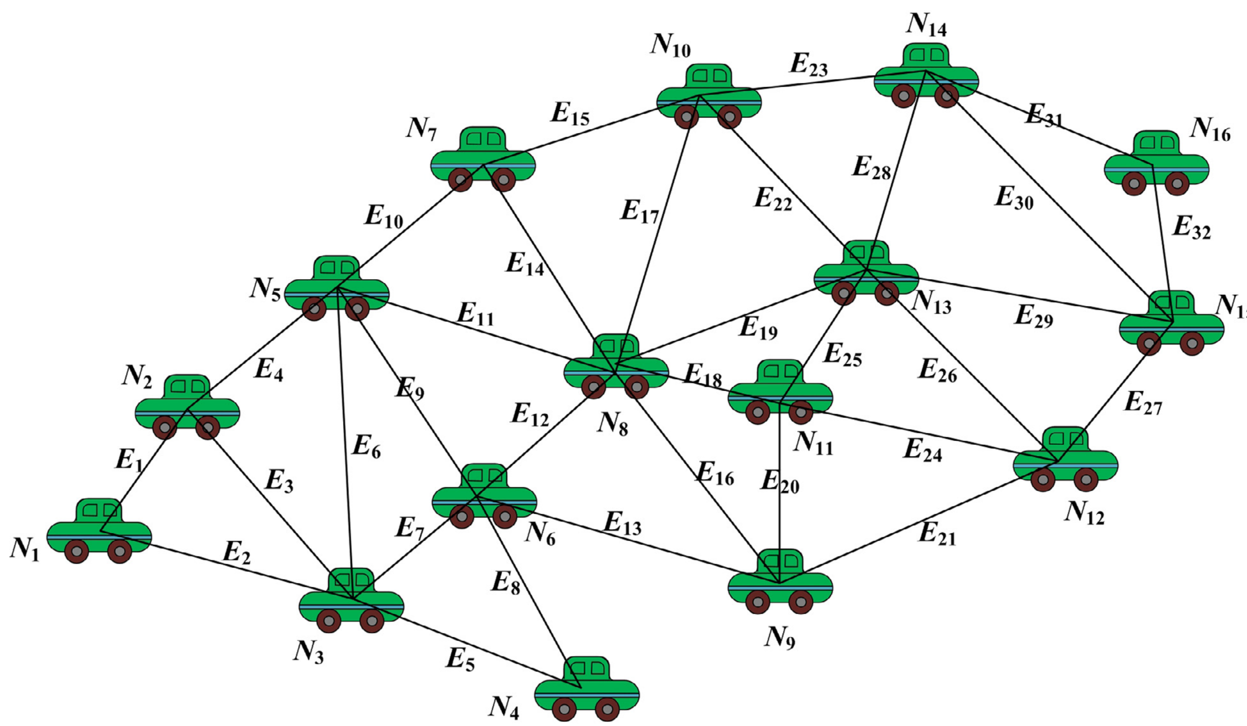


Fig. 3. A multicast routing example with 16 nodes and 32 edges locating in 400 × 500 m^2.

assumed not to change too frequently to computing and updating routing paths. In other words, there is a short stable period after the network topology has been altered.

3. Optimization algorithm

The original ABC algorithm is proposed to solve continuous optimization problems. It is recently extended to handle combinational optimization problems [26,27]. The colony of ABC is comprised by employed bees, onlooker bees, and scout bees. Accordingly, the algorithm is divided into three stages. The three bee groups are sent out one after another to search potential food sources. A loop of the three stages constitute a cycle of the algorithm as shown in Algorithm 1. The target of the search of bees is mapped to minimize an objective function or maximize a fitness function. In Algorithm 1, n_s is the number of solutions (i.e., food sources), *limit* is the number of consecutive evaluations that a solution fails to be updated. Generally, both the number of employed bees and the number of onlooker bees are equal to n_s. The number of scout bees relies on *limit* and the algorithm's evolutionary status.

To deal with the SMT problem, binary representation is utilized in this paper. As mentioned above, all nodes in graph G excluding source and destiny nodes are candidates for Setiner points. Thus, the problem dimension is $|N| - |N^s| - |N^d|$, where $|N^s|$ and $|N^d|$ are the number of source nodes and destiny nodes, respectively. The candidate nodes are coded to a binary string, where all nodes are numbered in order. Each element in binary string corresponds to a node of G and takes value 0 or 1, where value 0 means the associated node is not included in SMT, and value 1 means the node is a member of SMT. Thus, a subgraph can be built based on an instance of such kind of binary string. Take the graph in Fig. 1 for example. Suppose N_1 and N_5 are respectively source node and end node, the other nodes are encoded to a binary string in the order of N_2, N_3, N_4, and N_6. Hence 1010 can be decoded to subgraph shown in Fig. 2.

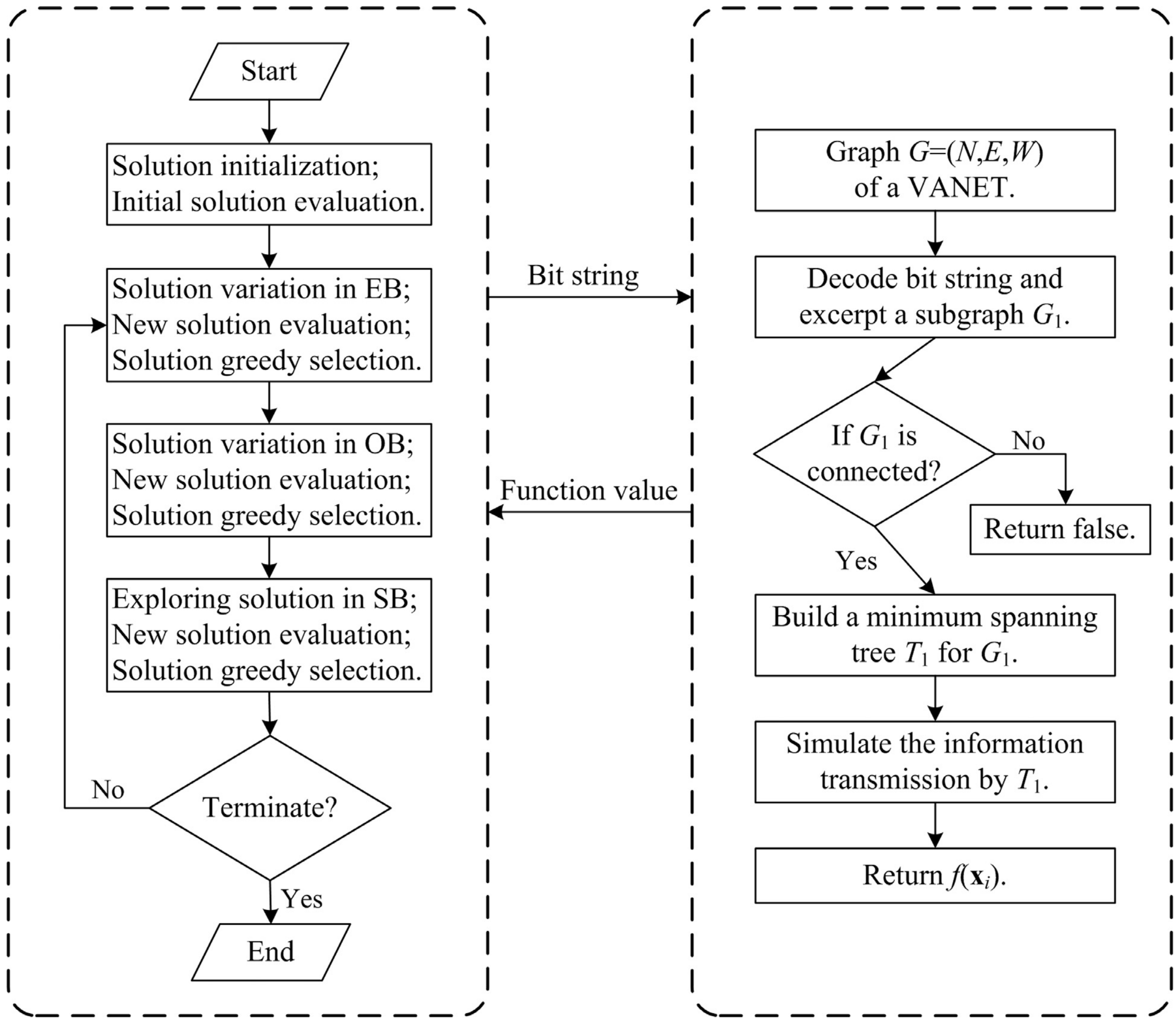


Fig. 4. Flow chart of applying MABC to the network example. EB: employed bee stage, OB: onlooker bee stage, SB: scout bee stage.

In employed bee stage, the following formula (1) is designed to generate a variation $\mathbf{v}_i$ based on solution $\mathbf{x}_i$:

$$v_{i,j} = \begin{cases} 1 - x_{i,j} & \text{if } j = j1 \\ x_{r1,j} & \text{if } j = j2 \\ x_{i,j} & \text{otherwise} \end{cases} \tag{1}$$

where subscript i denotes the solution index and subscript j is the bit string index; $j1 \le j2$ are randomly chosen positions; $\mathbf{x}_{r1}$ is also randomly selected solutions with $r1 \neq i$ in the population. This formula is modified based on the one-position inheritance scheme, which is proposed to deal with continuous variables [28].

In onlooker bee stage, a honey bee chooses a solution to exploit depending on the fitness of solutions in population. Because a micro population is used here, the selection pressure should not be too heavy; otherwise the population may be easily trapped in local optima. Each solution has at least selection probability $0.7/n_s$. In the following, the formula (2) is designed to generate $\mathbf{v}_t$ based on chosen solution $\mathbf{x}_t$:

$$v_{t,j} = \begin{cases} x_{t1,j} & \text{if } j = j3 \text{ and } fit(\mathbf{x}_{t1}) \ge fit(\mathbf{x}_t) \\ 1 - x_{t2,j} & \text{if } j = j4 \text{ and } fit(\mathbf{x}_{t2}) \le fit(\mathbf{x}_t) \\ x_{t,j} & \text{otherwise} \end{cases} \tag{2}$$

where $j3 \le j4$ are randomly chosen positions; $\mathbf{x}_{t1}$ and $\mathbf{x}_{t2}$ are randomly selected solutions with $t1 \neq t2 \neq t$ in the population. The physical meaning of this formula is a honey bee would fly toward the same direction given a solution is better than its current choice $\mathbf{x}_t$, and it would fly toward the opposite direction given a solution is worse than $\mathbf{x}_t$.

If the *limit* flag of a solution becomes true, the associated solution would be abandoned, and are substituted for a new one, which is randomly explored by a scout bee in search space. The procedure is alike to that in initialization. To assure convergence, elitism of size 1 is used in scout bee stage. That is the best solution would never be abandoned by honey bees. Theoretical study has proved that keeping the best so far solution is necessary for a SI algorithm converging to global optimum of a problem with probability one [29,30]. Thus, this usage of scout bee is helpful to solve the SMT problem.

The pseudo code of the micro ABC (MABC) algorithm is given in Algorithm 2 with $D = |N| - |N^s| - |N^d|$.

Observed from the variation formulas, candidate solution $\mathbf{v}_i$ is always a feasible solution. Hence, boundary repair method is not used in MABC. This avoids the notorious problem of choosing proper repair method [31]. Unlike the proposed algorithm in [27,32–34], The MABC algorithm does not introduce any new parameter, and does not increase the burden of algorithmic parameter control.

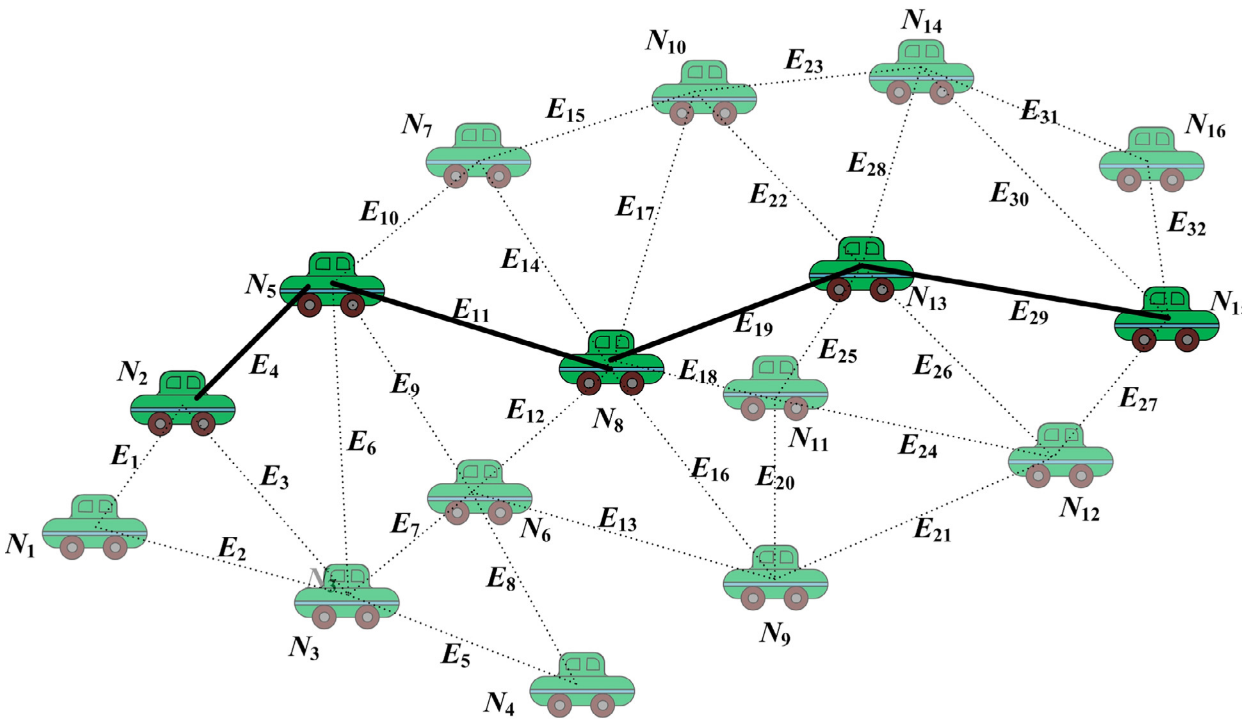


Fig. 5. The Steiner minimum tree with source node N_2 and destiny node N_{15}.

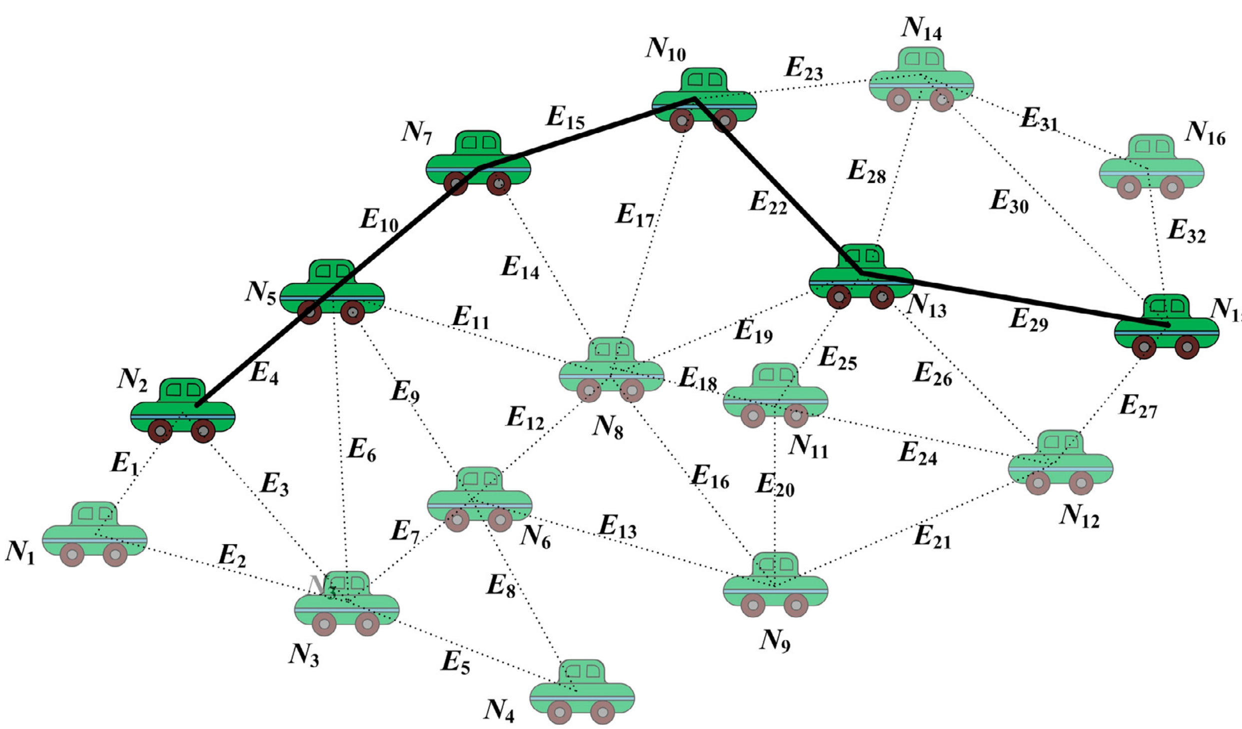


Fig. 6. The Steiner minimum tree with source node N_2 and destiny nodes N_{10}, N_{15}.

4. Simulation

In this section, the proposed MABC algorithm will be tested on a QoS constrained multicast routing example of inter-vehicle communication (IVC) network, which is a typical VANET.

4.1. Test example and simulation setting

An IVC network of 16 cars are taken as an example (see Fig. 3). The network illustrates a traffic scenario that cars are moving from bottom left to upper right among the main road. Its corresponding graph contains sixteen nodes ($|N| = 16$) and thirty two edges $|E| = 32$. The distance between two nodes is computed using Euclidean distance measure based on the locations of nodes. The node locations are obtained by global positioning system sensors embedded in cars. Moreover, suppose all vehicles utilize omnidirectional antennas. When a node N_i sends out data packages, all nodes that locates in the coverage area of N_i with transmission power p_i could receive the packages. An edge in graph G means a

Algorithm 1: Flow chart of the standard ABC algorithm.

Input: $\boldsymbol{f}(\cdot)$, D, Ω, n_s, $limit$, $feval = 0$
Output: a set of optimal solutions found by the ABC algorithm
randomly generate n_s solutions, $feval = feval + n_s$;
evaluate the function values of the initialized solutions, and compute their fitness values;
while *termination criteria are not satisfied* **do** `// Main cycle`
  for $i \leftarrow 1$ **to** n_s **do** `// Employed bee stage`
    employed bee i flies around solution i based on some rules and locates position $\mathbf{v}_i$;
    evaluate $\mathbf{v}_i$, $feval = feval + 1$;
    greedy selection between $\mathbf{v}_i$ and $\mathbf{x}_i$;
  end
  for $t \leftarrow 1$ **to** n_s **do** `// Onlooker bee stage`
    onlooker bee t chooses a solution based on the fitness of solutions;
    onlooker bee t flies around the chosen solution k based on some rules and produces $\mathbf{v}_t$;
    evaluate $\mathbf{v}_t$, $feval = feval + 1$;
    greedy selection between $\mathbf{v}_t$ and $\mathbf{x}_k$;
  end
  for $i \leftarrow 1$ **to** n_s **do** `// Scout bee stage`
    if *solution i has not been improved in the last limit evaluations* **then**
      a scout bee flies out and randomly explores in search space Ω to produce $\mathbf{x}_i^{'}$;
      replace solution $\mathbf{x}_i$ by $\mathbf{x}_i^{'}$
    end
  end
end

Algorithm 2: Flow chart of the MABC algorithm with binary representation.

Input: $\boldsymbol{f}(\cdot)$, D, $\Omega = \{0, 1\}$, n_s, $limit$, $feval = 0$
Output: a set of optimal solutions found by the MABC algorithm
randomly generate n_s solutions, $feval = feval + n_s$;
evaluate the function values of the initialized solutions, and compute their fitness values;
while *termination criteria are not satisfied* **do** `// Main cycle`
  for $i \leftarrow 1$ **to** n_s **do** `// Employed bee stage`
    employed bee i flies around solution i based on (1) and locates position $\mathbf{v}_i$;
    evaluate $\mathbf{v}_i$, $feval = feval + 1$;
    greedy selection between $\mathbf{v}_i$ and $\mathbf{x}_i$;
  end
  for $t \leftarrow 1$ **to** n_s **do** `// Onlooker bee stage`
    onlooker bee t chooses a solution $\mathbf{x}_t$ based on the fitness of solutions;
    onlooker bee t flies around the chosen solution based on (2) and produces $\mathbf{v}_t$;
    evaluate $\mathbf{v}_t$, $feval = feval + 1$;
    greedy selection between $\mathbf{v}_t$ and $\mathbf{x}_t$;
  end
  for $i \leftarrow 1$ **to** n_s **do** `// Scout bee stage`
    if $\boldsymbol{x}_i$ *has not been improved in the last limit evaluations and is not the best one in population* **then**
      a scout bee flies out and randomly explores in search space Ω to produce $\mathbf{x}_i^{'}$;
      replace solution $\mathbf{x}_i$ by $\mathbf{x}_i^{'}$
    end
  end
end

node lies in the coverage area of the other node. The weight function $w_{i,j}$ between two nodes N_i, N_j is related with the queue delay and propagation delay $d_{i,j}$ as well as their distance $dist_{i,j}$. The energy consumption c_i of data transmission from N_i to N_j is a function of $dist_{i,j}$ based on the energy attenuation model of wireless communication:

$$c_{i,j} = k\left(dist_{i,j}\right)^{\beta} + c_0, \tag{3}$$

where k is a constant related with antenna property, c_0 is the energy amount of receiving a data package, $\beta \in [2, 4]$ is path propagation attenuation factor. For simplicity, these parameters are $k = 1$, $\beta = 2$, and $c_0 = 1$ in the simulation. Thus, the amount of energy consumption from source node to destiny nodes is:

$$c(T)^e = \sum_{N_i \in T} c_{i,j}, \tag{4}$$

where T is a Steiner minimum tree linking source nodes and end nodes. The transmission delay cost of tree T is:

$$c(T)^d = \sum_{N_i, N_j \in T} d_{i,j}. \tag{5}$$

Therefore, the target of the example is to minimize both energy consumption cost and transmission delay cost. In this way, the maximum lifetime and minimum cost multicast routing can be accomplished. The SMT T is attained by the proposed MABC algorithm. The function value of each solution is defined as follows:

$$f(\mathbf{x}_i) = c(T)^e + c(T)^d. \tag{6}$$

where T is the corresponding Steiner minimum tree of solution $\mathbf{x}_i$.

The flow chart of applying MABC to deal with the network example is shown in Fig. 4. The MABC algorithm is responsible for searching the Steiner points, while the function value is computed by (6) from the network. In case the subgraph decoded from a solution is disconnected, which indicating the solution does not fulfill the constraints, the function value is set to infinity. Minimum spanning tree of a subgraph is obtained by Kruskal algorithm.

The parameter setting of the MABC algorithm is:

(1) $n_e = 6$;
(2) $limit = \min(0.5n_eD, 0.3 * MFE)$.

where MFE is the maximum number of function evaluations. The algorithm is independently executed 25 times to obtain its average performance. The proper of this parameter setting has already been demonstrated as in [35,36]. It terminates when either the following conditions is met:

(1) MFE is reached, where $MFE = 2000$;
(2) $|f(\mathbf{x}) - f(\mathbf{x}^*)| \le 10^{-6}$, where $f(\mathbf{x})$ stands for the best value found by MABC and $f(\mathbf{x})^*$ is the optimal value found by hand with the assistance of shortest path algorithm.

Expected running time (ERT) means the average running time of MABC reaching optimal value. The following metric is used [37]:

$$E(RT(\Delta f)) = \hat{E}(FE^s) + \frac{1 - \hat{p}^s}{\hat{p}^s}\hat{E}(FE^u)^{\tilde{}}, \tag{7}$$

where $RT(\Delta f)$ denotes the runtime of the algorithm finding $\Delta f = 10E^{-6}$; $\hat{E}(FE^s)$ means the average number of functions evaluations that those runs successfully reach Δf; $\hat{E}(FE^u)$ is the average number of function evaluations in the unsuccessful runs; $\hat{p}^s$ is the ratio of runs that reaching Δf over 25 runs. In this paper, $\hat{E}(FE^u) = 2000$.

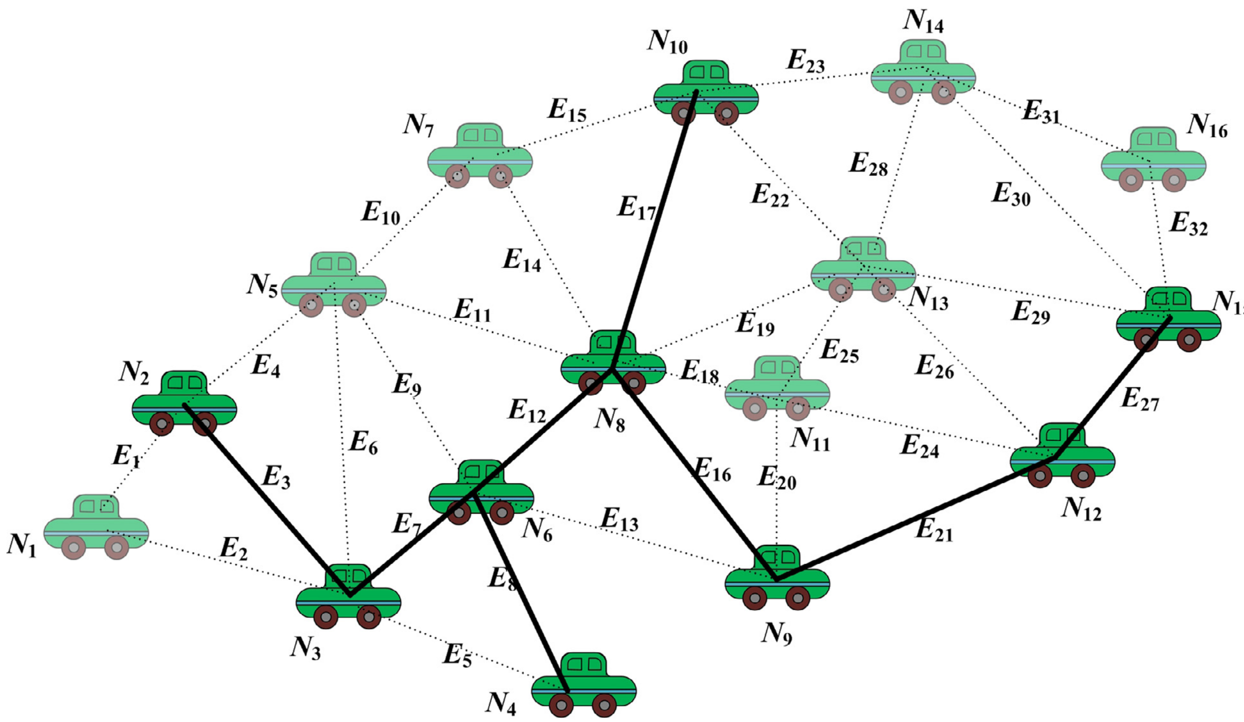


Fig. 7. The Steiner minimum tree with source node N_2 and destiny nodes N_4, N_9, N_{10}, N_{15}.

Table 1
Summary of the results found by the MABC algorithm in dealing with the three test cases.

case	Weight	Node sequence	$\hat{p}^s$	ERT	ECT
1	843.98	N_2 N_5 N_8 N_{13} N_{15}	**100%**	308.60	**0.071**
2	887.52	N_2 N_5 N_7 E_{10} N_{13} N_{15}	**100%**	315.96	**0.079**
3	1273.53	N_2 N_3 N_6 N_4 N_8 E_{10} N_9 N_{12} N_{15}	**100%**	307.72	**0.106**

Note: In case 1, N_2 is source node, N_{15} is destiny node;
In case 2, N_2 is source node, N_{10} and N_{15} are destiny nodes;
In case 3, N_2 is source node, N_4, N_9, N_{10}, and N_{15} are destiny nodes.

The simulation of the network and MABC algorithm is implemented in Matlab, and executed on a personal computer with 4-core 2.50GHz CPU and 4GB of memory. Thus, a fair comparison can be conducted under the same running environment.

4.2. Simulation results

Case 1: suppose source node is N_2 and destiny node is N_{15} in the network of Fig. 3. Dijkstra shortest path algorithm is used in this case to find the optimal path, which is shown in Fig. 5. The optimal path is $N_2 - N_5 - N_8 - N_{13} - N_{15}$, and the optimal function value is 843.98. The MABC algorithm is independently executed 25 trials. It reaches the predefined threshold Δf in all trials.

Case 2: suppose source node is N_2 and destiny nodes are N_{10} and N_{15} in the example. The SMT route in this case is $N_2 - N_5 - N_7 - N_{10} - N_{13} - N_{15}$ as shown in Fig. 6, and the optimal function value is 887.52. The optimal solution is obtained by hand with the help of Dijkstra shortest path algorithm. MABC can find the optimal solution in all trials.

Case 3: suppose source node is N_2 and destiny nodes are N_4, N_9, N_{10} and N_{15} in the example. The SMT route in this case is given in Fig. 7, and the optimal function value is 1273.53. The MABC algorithm can find the optimal solution in all trials in the case of four destiny nodes.

Table 1 presents a summary of the results attained by the MABC algorithm in dealing with the above three cases. In the table, MABC successfully finds the SMT solution for all test cases with a success rate $\hat{p}^s = 100\%$. This means that the algorithm is reliable to handle multicast routing problem in VANETs. The SMT routes of case 2 and case 3 are much different, while the algorithm is able to attain the optimal solution. Moreover, in all test cases, their ERT does not change much. Note that ERT assesses the performance of an algorithm in the view of function evaluations, hence the algorithm is effective when the number of destiny nodes increases.

The expected computational time (ECT) of the MABC algorithm in searching the optimal route is shown in the last column of Table 1. ECT is counted in seconds (s) and approximated by the average of computer time in 25 trials. Clearly, the ECT value of case 3 is the greatest, the value of case 2 is the second, and following is case 1. For case 3, the computer time is 0.106 s. In a carriageway scenario where the network structure does not change frequently, the multicast routing algorithm can be applied in real time.

5. Conclusion

Efficient and reliable communication in VANETs heavily depends on the construction of strong routes among vehicles. Multicast routing plays an important part in information transmission in VANETs. Hence, the research of optimal route in multicast routing is meaningful and valuable. Relating with graphic theory, an undirected acyclic graph G is apt to characterize a VANET, then the routing problem is transformed to Steiner minimum tree (SMT) searching problem. Existing studies prove that SMT is NP-hard.

This paper considers QoS constrained multicast routing problem, where the quality measures include energy consumption cost and transmission delay cost. The main contributions of this paper are as follows:

(1) Multicast routing is modeled as a continuous optimization problem. The objective is to maximize network lifetime and minimize communication cost.

(2) MABC algorithm is proposed in this paper, which works on a micro population for reducing the computational time, whereas it should not be too small to being effective.
(3) New variation formula for employed bee and onlooker bee stages are created to producing good solutions by using solutions in current population.

The micro ABC (MABC) algorithm is tested on three cases with increasing number of destiny nodes. In all cases, the algorithm successfully finds the optimal solutions. From the viewpoint of function evaluations, which is measured by expected running time (ERT), MABC needs about the same ERT to solve all test cases. In the view of computational time, the computer time is 0.106 s. Therefore, the simulation results show that MABC is effective and reliable in handling QoS constrained multicast routing problem.

The ERT of MABC could be reduced by inventing more powerful variation formula or by identifying search patterns in solutions, which should be investigated further in future.

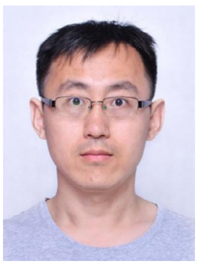
Acknowledgements

This research was supported in part by the Applied Basic Research Program of Tianjin (15JCYBJC51500, 15JCYBJC52300), and the Tianjin Thousand Youth Talents Plan Project of Tianjin Normal University (ZX110023).

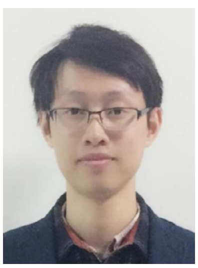
References

[1] X. Wang, Q. Liang, On the throughput capacity and performance analysis of hybrid wireless networks over fading channels, IEEE Trans. Wireless Commun. 12 (6) (2013) 2930–2940.
[2] Q. Liang, X. Cheng, S. Huang, D. Chen, Opportunistic sensing in wireless sensor networks: theory and applications, IEEE Trans. Comput. 63 (8) (2014) 2002–2010.
[3] Q. Liang, Radar sensor wireless channel modeling in foliage environment: UWB versus narrowband, IEEE Sensors J. 11 (6) (2011) 1448–1457.
[4] Q. Liang, X. Cheng, S. Samn, NEW: Network-enabled electronic warfare for target recognition, IEEE Trans. Aerosp. Electron. Syst. 46 (2) (2010) 558–568.
[5] G. Singal, V. Laxmi, M.S. Gaur, V. Rao, Moralism: mobility prediction with link stability based multicast routing protocol in MANETs, Wireless Netw. (2016) 1–17.
[6] Q. Liang, Situation understanding based on heterogeneous sensor networks and human-inspired favor weak fuzzy logic system, IEEE Syst. J. 5 (2) (2011) 156–163.
[7] Q. Liang, X. Cheng, KUPS: Knowledge-based ubiquitous and persistent sensor networks for threat assessment, IEEE Trans. Aerosp. Electron. Syst. 44 (3) (2008) 1060–1069.
[8] R. Lachowski, M.E. Pellenz, M.C. Penna, E. Jamhour, R.D. Souza, An efficient distributed algorithm for constructing spanning trees in wireless sensor networks, Sensors 15 (1) (2015) 1518–1536.
[9] H. Zhou (Ed.), Wireless Ad-Hoc Networks, InTech, Vienna, Austria, 2012.
[10] G.-H. Lin, G. Xue, Steiner tree problem with minimum number of Steiner points and bounded edge-length, Inf. Process. Lett. 69 (1999). 53–47.
[11] M.R. Garey, R.L. Graham, D.S. Johnson, The complexity of computing steiner minimal trees, SIAM J. Appl. Math. 32 (4) (1977).
[12] G.L. Xue, Minimum-cost QoS multicast and unicast routing in communication networks, IEEE Trans. Commun. 51 (5) (2003) 817–824.
[13] R.-L. Tang, Z. Wu, Y.-J. Fang, Adaptive multi-context cooperatively coevolving particle swarm optimization for large-scale problems, Soft Comput. (2016) 1–20.
[14] Y. Lou, J. Li, Y. Shi, L. Jin, Gravitational co-evolution and opposition-based optimization algorithm, Int. J. Comput. Intell. Syst. 6 (5) (2013) 849–861.
[15] Z. Wu, X. Xia, B. Wang, Improving building energy efficiency by multiobjective neighborhood field optimization, Ener. Buildings 87 (2015) 45–56.
[16] V.P. Kompella, J.C. Pasquale, G.C. Polyzos, Multicasting routing for multimedia communication, IEEE/ACM Trans. Netw. 1 (3) (1993) 286–292.
[17] Q. Sun, H. Langendorfer, An efficient delay-constrained multicast routing algorithms, J. High-Speed Netw. 7 (1) (1998) 43–55.
[18] M. Parsa, Q. Zhu, J.J. Garcia-Luna-Aceves, An iterative algorithm for delay–constrained minimum-cost multicasting, IEEE/ACM Trans. Netw. 6 (4) (1998) 461–474.
[19] D. Gutierrez-Reina, S.L.T. Marin, P. Johnson, F. Barrero, An evolutionary computation approach for designing mobile ad hoc networks, Expert Syst. Appl. 39 (8) (2012) 6838–6845.
[20] D. Natarajan, A.P. Rajendran, AOLSR: hybrid ad hoc routing protocol based on a modified Dijkstra's algorithm, EURASIP J. Wireless Commun. Netw. 2014 (1) (2014) 1–10.
[21] W. Li, K. Li, Y. Huang, S. Yang, L. Yang, A EA- and ACA-based qos multicast routing algorithm with multiple constraints for ad hoc networks, Soft Comput. (2016) 1–11.
[22] R.-H. Hwang, W.-Y. Do, S.-C. Yang, Multicast routing based on genetic algorithms, J. Inf. Sci. Eng. 16 (1) (2000) 885–901.
[23] Y.S. Yen, Y.K. Chan, H.C. Chao, J.H. Park, A genetic algorithm for energy-efficient based multicast routing on MANETs, Comput. Commun. 31 (4) (2008) 2632–2641.
[24] R. Forsati, A.T. Haghighat, M. Mahdavi, Harmony search based algorithms for bandwidth-delay-constrained least-cost multicast routing, Telecommun. Syst. 34 (3–4) (2007) 147–166.
[25] J. Toutouh, J. Garcia-Nieto, E. Alba, Intelligent OLSR routing protocol optimization for VANETs, IEEE Trans. Vehicular Technol. 61 (4) (2012) 1884–1894.
[26] G. Pampara, A.P. Engelbrecht, Binary artificial bee colony optimization, in: 2011 IEEE Symposium on Swarm Intelligence (SIS), IEEE, Paris, France, 2011, pp. 1–8.
[27] C. Ozturk, E. Hancer, D. Karaboga, A novel binary artificial bee colony algorithm based on genetic operators, Inf. Sci. 297 (2015) 154–170.
[28] X. Zhang, S.Y. Yuen, Improving artificial bee colony with one-position inheritance mechanism, Memetic Comput. 5 (3) (2013) 187–211.
[29] G. Rudolph, Convergence analysis of canonical genetic algorithm, IEEE Trans. Neural Netw. 5 (1) (1994) 96–101.
[30] M. Jiang, Y.P. Luo, S.Y. Yang, Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm, Inf. Process. Lett. 102 (1) (2007) 8–16.
[31] J. Arabas, A. Szczepankiewicz, T. Wroniak, Experimental comparison of methods to handle boundary constraints in differential evolution, in: Parallel Problem Solving from Nature, PPSN XI, Krakow, Poland, 2010, pp. 411–420.
[32] P.K. Singhal, R. Naresh, V. Sharma, A modified binary artificial bee colony algorithm for ramp rate constrained unit commitment problem, Int. Trans. Electr. Ener. Syst. 25 (12) (2015) 3472–3491.
[33] M.H. Kashan, N. Nahavandi, A.H. Kashan, DisABC: A new artificial bee colony algorithm for binary optimization, Appl. Soft Comput. 12 (1) (2012) 342–352.
[34] Q.-K. Pan, L. Wang, J.-Q. Li, J.-H. Duan, A novel discrete artificial bee colony algorithm for the hybrid flowshop scheduling problem with makespan minimisation, Omega 45 (2014) 42–56.
[35] S.Y. Yuen, X. Zhang, On composing an algorithm portfolio, Memetic Comput. 7 (3) (2015) 203–214.
[36] X. Zhang, X. Zhang, S.L. Ho, W.N. Fu, A modification of artificial bee colony algorithm applied to loudspeaker design problem, IEEE Trans. Magnet. 50 (2) (2014) 737–740.
[37] N. Hansen, A. Auger, R. Ros, S. Finck, P. Pošík, Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009, in: Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO '10, ACM, New York, NY, USA, 2010, pp. 1689–1696.

Xiu Zhang received the B. Eng. and M. Eng. degrees in biomedical engineering from the Hebei University of Technology, Tianjin, China, in 2006 and 2009, respectively. Her master degree research concerns the analysis of the electroencephalograph signal when the magnetic signal is used to stimulate the acupoints in human subjects. She received the Ph.D. degree and completed the postdoctoral research work in electrical engineering from The Hong Kong Polytechnic University in 2012 and 2015, respectively. She is now a lecturer in Tianjin Normal University. She has published about 20 papers in the IEEE TRANSACTIONS. Her research interests mainly focus on numerical methods of electromagnetic field computation, novel wireless energy transfer systems, and wireless network optimization.



Xin Zhang received the B. Sc., M. Sc., and Ph.D. degrees from Ludong University, Shandong University of Science and Technology, City University of Hong Kong, in 2006, 2009, and 2013, respectively. He is currently a Lecturer in the College of Electronic and Communication Engineering, Tianjin Normal University. His main research interests are swarm intelligence, communication network optimization, evolutionary computation, and machine learning. He has published more than 15 technical papers on these subjects, including more than eight international journal papers.



Cheng Gu received the B. Sc. degree in applied physics from Nanjing University of Information Science and Technology, Nanjing, China, in 2014. He is pursuing the M. Eng. degree at Tianjin Normal University since 2015. His research interests mainly focus on metamaterials and its applications in wireless power transfer system.