Cybernetics and Systems
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ucbs20

A New Routing Protocol for Energy Optimization in Mobile Ad Hoc Networks Using the Cuckoo Optimization and the TOPSIS Multi-Criteria Algorithm

Shayesteh Tabatabaei

CYBERNETICS AND SYSTEMS
https://doi.org/10.1080/01969722.2021.1899597

A New Routing Protocol for Energy Optimization in Mobile Ad Hoc Networks Using the Cuckoo Optimization and the TOPSIS Multi-Criteria Algorithm

Shayesteh Tabatabaei

Department of Computer Engineering, Higher Education Complex of Saravan, Saravan, Iran

ABSTRACT

The Mobile Ad Hoc Network (MANET) is a set of ad hoc nodes without any infrastructure that form a temporary network without the presence of central control. Nodes present in such networks are usually in movement and result in frequent topological changes. The design of proper routing protocols to control dynamic topological changes in an ad hoc network can enhance the performance of a network. In this regard, the current study attempts to present a new routing protocol to improve energy consumption using the Cuckoo Optimization Algorithm and the TOPSIS multi-criteria algorithm. The proposed approach can be applied in dynamic environments, and four important variables including the accessible bandwidth, the remaining energy, the speed of movement, and the number of steps necessary for routing have been considered for it. To select proper groups for the purpose of relaying data packages, the TOPSIS multi-criteria algorithm was applied. In addition, the Cuckoo Optimization Algorithm was used to select the shortest route between the relay groups. The simulation results showed that the process of selecting a stable route using the Cuckoo Optimization Algorithm and the TOPSIS algorithm has a significant impact on the performance of networks, and the proposed algorithm shows better performance compared to the methods introduced in Tabatabaei and Behravesh with regard to the throughput and the end-to-end delay.

KEYWORDS

Cuckoo Optimization Algorithm; energy consumption; mobile ad hoc network; routing algorithm; TOPSIS multi-criteria algorithm

1. Introduction

The Mobile Ad Hoc Network (MANET) refers to self-configuring wireless mobile devices that make up a temporary and dynamic wireless system without the help of any network infrastructure or a concentrated base network. Constraints of a MANET include such items as dynamic topology, limited resources of the network (e.. delay, bandwidth, and energy), high

CONTACT Shayesteh Tabatabaei shtabatabaey@yahoo.com Department of Computer Engineering, Higher Education Complex of Saravan, Saravan, Iran.

mobility of the node, and the low bandwidth of the channel (Junhai Liu, and Danxia 2008).

The ad hoc networks are applied in sections of a network – like military battlefields, emergency search operations, rescue sites, classrooms, and conventions where the participants are able to share their information in a dynamic manner with the mobile devices – where rapid mapping and the dynamic reconfiguration are required. Nodes in such networks are mobile. Due to the high mobility of such nodes, the routes might be destroyed in a way that communications break down until a new route is discovered. In addition, due to the limited communication range of mobile nodes in ad hoc networks, some nodes are not able to communicate directly with each other. The mobile nodes not only act as hosts but also can be applied as routers for maintaining the route toward the destination and sending data packages to the other nodes of the network. The performance of ad hoc networks depends on the efficiency of the routing protocols, and their efficiency is related to a number of factors like the time of convergence after topological changes, the bandwidth overload for proper routing, and energy consumption (Raich and Vidhate 2013). Based on different structures of a network, mobility scenarios, and different programs for ad hoc networks, numerous routing protocols have been proposed. The mobility of nodes is a challenging feature that has to be taken into consideration by the designers of routing protocols. That is because high mobility can lead to dynamic topological changes in the network, and complicate the operation of routing algorithms (Bilgin and Khan 2010). Routing protocols in ad hoc networks have to be completely adapted to topological changes, and the nodes have to exchange information related to the topology of the network for the purpose of creating routes. In ad hoc networks, routing is considered a challenging issue. The number of steps is the typical parameter used in selecting a route from the source to the destination. However, this parameter may show poor performance in networks. That is because a path with the minimum number of steps may involve slow or missed steps, which can reduce the throughput. The common routing protocols have been designed for static bonding, and cannot converge to a stable state in ad hoc networks with frequently changing bonding. Based on the above points, routing is a major challenge in MANETs. Thus, the current study is an attempt to propose an intelligent routing protocol based on the Cuckoo Optimization Algorithm and the TOPSIS multi-criteria algorithm to find routes that are highly stable and reduce energy consumption. The current article has been arranged in the following manner: Section 2 reviews the related literature. The proposed method has been presented in Section 3, while Section 4 presents the simulation results. Finally, the conclusion of the study has been presented in Section 5.

2. The Related Literature

A MANET can be described as a self-configuring network with independent nodes that have no infrastructure. The nodes participating in MANETs are in permanent movement and result in constant topological changes. Designing appropriate routing protocols to deal with topological changes in MANETs can enhance the performance of networks. In this regard, four algorithms have been proposed in Tabatabaei and Behravesh (2017) for the problem of routing in MANETs. The first method is a new method called the "Classic Logic Routing Algorithm" (CLRA). The second is called the "Fuzzy Logic Routing Algorithm" (FLRA), while the third method is the "Reinforcement Learning Routing Algorithm" (RLRA). Finally, the fourth algorithm is the "Reinforcement Learning and Fuzzy Logic-based Algorithm" (RLFLRA).

The goal of the above methods is to design a routing protocol for MANETs and the training of network nodes with reinforcement learning and fuzzy learning for the purpose of updating their routing tables and estimating the stability of the routes. Such proposed approaches can be implemented in dynamic environments, and four important fuzzy variables – the accessible bandwidth, the remaining energy, the speed of movement, and the number of steps – have to be taken into consideration. The simulation results showed that the process of training had a major impact on the performance of a network, and the RLFLRA outperformed the other proposed methods in terms of the throughput, time is taken for discovering a route, the rate of package delivery, delays in access to the network, and the number of steps.

In He, Yang, and Zhou (2018), the Globally-aware Limited Bandwidth Allocation Optimization (GLBAO) in a wireless network using multi-path routing was investigated. The researchers considered the current cost a function of end-to-end delay, the consumed energy, and the distance so as to select the best path among several alternatives. Using the current costs makes it possible to take into account not only the current local transmission but also any other global transmission from other parts of the networks. Thus, an effective scheme for bandwidth allocation was proposed for multi-path routing. In addition, a Half-Circle Zone (HCZ) algorithm together with the arrangement of separate to reduce the size of the routing table (RPT) to improve the efficiency of routing. The efficiency factor was also applied in the optimal objective function to enhance the performance of the line. The researchers' main goal was to introduce the concept of current costs as a general/all-encompassing method in bandwidth allocation and path selection (in comparison to many traditional routing algorithms in which only the shortest distance or the lowest level of energy consumption are taken into consideration), dealing with the concept of globally-aware multi-path routing and bandwidth allocation by defining a crowded node that is used for transmission by other nodes of a network, using the operation actor to evaluate the performance of a line in the model of bandwidth optimization while making decisions on the allocation of proper bandwidth and proposing a new HCZ algorithm to enhance the efficiency of routing and reducing the transmission time, proposing the GLBAO algorithm, and approving the enhancement of performance through simulating their proposed method. The simulation results using OPNET 14.5 indicated the high reliability of package transmission and the reduction of congestion in a network.

In Yadav, Das, and Tripathi (2017), it was attempted to control the unreliability issues by the fuzzy logic in order to maintain the resources of a network. In this mechanism, all network criteria within a path are converted into a unitary criterion called “Fuzzy Cost” or “Communication Cost”. The paths with the minimum fuzzy cost are selected as the favorable path, and the data from the source node are transmitted from this path to the set of receivers. In this article, an Efficient Fuzzy-based Multi-Constraint Multicast Routing Protocol (EFMMRP) was proposed for a MANET. In this protocol, the constraints of multiple QoS in a particular time in terms of end-to-end delay, channel bandwidth, and the consumed energy are taken into consideration, and are regarded as a unitary criterion called “Fuzzy Cost”. Their proposed protocol was simulated using the NS-2 simulator and the MATLAB instrument, and the results indicated that this method outperformed the present routing protocols like ODMRP or Multipath Ad hoc On-demand Distance Vector (MAODV) in terms of package delivery rate, package delivery delay, and control overload.

In Clausen, Yi, and Herberg (2017), the Lightweight On-Demand Ad hoc Distance Vector Routing Protocol-Next Generation (LOADng) was proposed to perform an efficiency, scalable, and safe routing in low-energy and weak networks. In the routing table, the reaction protocol does not maintain paths for all destinations, but starts discovering the paths considering a destination where the data will be sent to. In this manner, the routing costs and the memory needed for path discovery can be reduced. Using a modular approach, loading can be expanded using extra components to adapt the protocol to different topologies, traffics, and features of the bonding layer of the data. In this study, several extra components were considered to expand the LOADng, which included supporting the request for intelligent paths and expanding the search sphere, making it possible to maintain the topology of trees from the source to the destination, and a quick format of changing paths. All these forms of expansion were investigated in terms of characteristics, the potential of cooperation with other mechanisms, security weaknesses, performance, and operation. In addition, a general framework was proposed for guaranteeing safety in the routing protocol. The goals of this protocol include reducing delays and reducing them and controlling the overload of consumed memory in each router to maintain the complete topology of the network. Typically, in the reaction protocols, the required memory to routing is limited to the active paths, which increases the costs of delay for routing as required and the overload dependent on the flow of data.

In Tabatabaei, Teshnehlab, and Mirabedini (2015), the routing protocols and their performance were investigated by simulating them in NS-3 in order to identify their major weaknesses. The early results showed that the previous routing protocols are not able to take the dynamic nature of ad hoc networks into consideration, and consequently perform weakly with regard to the rate of transmitted packages and the transmission overload. Thus, they proposed a reliable and practical opportunistic routing protocol called "ORGMA", which is able to provide reliable delivery of packages in ad hoc networks. ORGMA is a gradient-based approach where the sender simply releases a package and the receivers take decisions regarding the process of routing. Using the signal to noise (SNR) ratio to determine the costs of routing and developing the management of low lightweight costs bases on flooding enabled ORGMA to actualize the high rate of package delivery in the dynamic environments of ad hoc networks. The use of the gradient-based sending method in these dynamic environments is the key to the success of this protocol. First, the routing protocol uses the SNR ratio – which can be obtained easily without any need for long-term measurement – as the routing criterion. When there are unavoidable measurement errors for the SNR ratio, the opportunistic routing variations can be effectively overcome. Second, the routing protocol provided the management of lightweight and effective costs using the flooding technique. Through designing the routing of messages and tables differently, the routing protocol provides for a reliable end-to-end package delivery rate with acceptable control overload. Using the NS-3 simulator, the researchers evaluated their proposed routing protocol and compared it with other methods. The results showed that the proposed protocol not only was able to outperform the previous routing protocols – particularly in terms of package delivery – but also its performance was very close to the ideal routing protocol.

In Bhattacharya and Sinha (2017), a new routing protocol called the Least Common Multiple-based Routing was proposed to distribute loads in an ad hoc multi-path network. In this method, the protocol first finds several paths between the source and destination. Then, it estimates the time required to send packages in each one of these paths. The data packages are transmitted from the source to the destination, and then they are distributed in these multiple paths in a way that the number of data packages sent along each one of these paths is proportionate to the inverted time of routing in that particular path. This strategy balances load distribution across all paths and minimizes the overall taken to send data packages. The paths between the sources and the destination are discovered similar to the Ad hoc On-demand Distance Vector (AODV) routing protocol. However, the difference is that instead of the number of steps, the time taken to reach the destination in each step is measured, and the multiple paths – if they exist – are determined through the process of path discovery. Their proposed method worked well in terms of delivering packages in different paths and showed better performance in terms of theoretical analysis and simulation results in comparison to such methods as the Fibonacci Multipath Load Balancing (FMLB) and the MAODV protocols in terms of theoretical analysis and the simulation results. That is because in their proposed method the load distribution technology considered the real-time routing time in various paths, while this criterion had not been taken into consideration in the FMLB protocol. The real-time routing ratio along different paths might be quite different from the corresponding Fibonacci ratio (which has a bit natural relationship with the routing duration in different paths). In addition, since LCMR sends a larger number of packages (assuming they have equal sizes) along paths needing lesser routing time – instead of routing with an equal number of packages along all paths – it outperforms the MAODV method. In Upadhyay et al. (2012), three routing protocols called the AODV, Dynamic Source Routing (DRS), and the Destination-sequenced Distance Vector (DSDV) were compared with the concentration of nodes in the node mobility scenario. It was found that when the source nodes are fixed and the destination node is mobile, the DSR protocol outperforms the AODV and the DSDV protocol. Thus, the performance of the DSR routing protocol was found to be better than the AODV and DSDV protocols, and the DSR protocol performed better when faced with higher traffic.

3. The Proposed Method

In this section, first, the TOPSIS multi-criteria decision-making algorithm will be explained, and then the stages of the proposed algorithm to solve the problem of routing by the selection of proper nodes for relaying and the optimization of energy consumption in an ad hoc network will be discussed. The TOPSIS algorithm was proposed by Huang and Yun in 1981 and is one of the best multi-criterion decision-making models. In this method, the selected option has to have the least distance with the positive

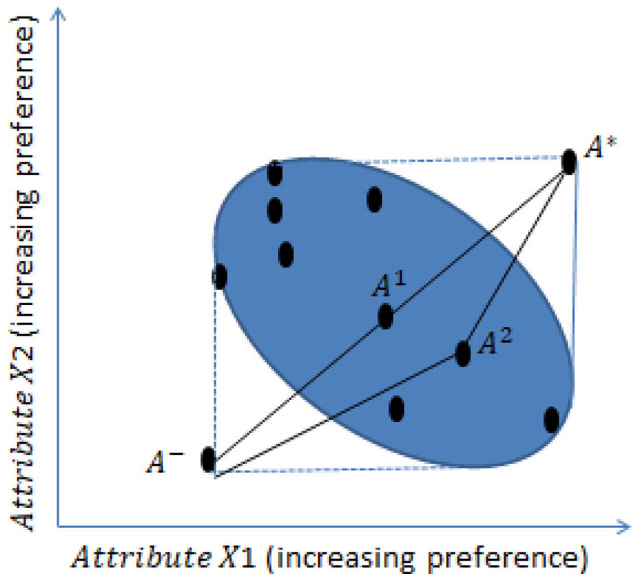


Figure 1. Basic concept of TOPSIS method (A^*: ideal point, A^-: negative-ideal point).

ideal solution (the best case possible) and the highest distance with the negative ideal solution (the worst case possible).

The proposed algorithm consists of three phases: In the first phase in which the relaying nodes are selected, the network nodes are selected using the TOPSIS algorithm according to the distance to the destination, the volume of performed work, the number of neighboring nodes, and the remaining energy in the nodes. In the second phase, which is the routing phase, the shortest path is selected from the set of the paths formed during the previous phase using the Cuckoo Optimization Algorithm. In the third phase called "Path Maintenance", the mobile nodes send a greeting package in an alternate fashion to their neighbors. In case that those neighbors are not within the range due to an error or battery failure, they will not reply to the greeting. Then, the node is informed of this failure and uses an alternative path that had been found within the routing phase by the Cuckoo Optimization Algorithm. The details of the proposed method have been presented below.

Stage one – The selection of relaying nodes

In the proposed method, the TOPSIS algorithm was used for finding the best nodes as cluster heads in clustering the MANET. The basic concept of TOPSIS is that the selected option has to have the least distance from the positive ideal solution and the highest distance from the negative ideal solution (Figure 1). In this method, m options are evaluated using n criteria, and each problem can be considered as a geometric system consisting of m points in an n-dimensional space. This algorithm consists of six stages that will be discussed below.

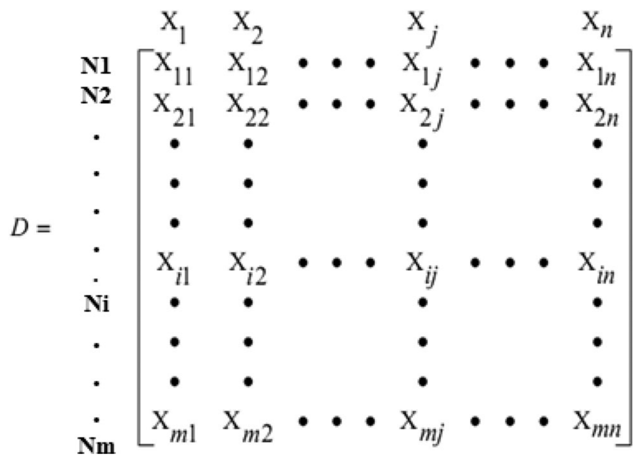


Figure 2. The decision matrix in the proposed method, where $\mathbf{A_i}$ is the *i*th node, and $\mathbf{X_{ij}}$ is the numerical value obtained from the *i*th option using the *j*th index.

Stage zero – Obtaining the decision matrix

In this method, the decision matrix is formed from m options and n indices, and the index with positive desirability is called "index profit", while the index with negative desirability is the "cost index". In the proposed method, $m = 25$ indicates the number of mobile nodes and $n = 4$ which indicates the number of neighbors, distance from the node to the destination, the number of jobs, and the remaining energy, respectively. Here, the indices for the number of neighbors and the remaining energy are profit indices, while the indices for the distance from the node to the destination and the number of jobs are cost indices. Figure 2 has illustrated the decision matric of the proposed method.

Stage one – Normalizing the decision matrix

In this stage, each value of the decision matrix obtained in stage zero is divided by the value of the vector related to the same index. Consequently, the value of r_{ij} is obtained from Eq. (1):

$$r_{ij} = \frac{X_{ij}}{\sqrt{\sum_{i=1}^{m} X_{ij}{}^2}} \tag{1}$$

where r_{ij} is the normalized value of the decision matrix $\mathrm{X_{ij}}$, and M indicates the number of mobile nodes.

Stage two – weighing the normalized matrix

In this stage, a weight is assigned to each index, and the total weight (W) is multiplied by the normalized matrix (R). In the proposed method, the weights assigned for the indices job, the number of neighbors, the remaining energy, and distance from the source to the destination was 0.1, 0.2, 0.4, and 0.3, respectively

$$W = (w_1,\ w_2,\ w_3,\ \ldots,\ w_j,\ \ldots, w_n)$$

$$\sum_{j=1}^{n} w_j = 1 \tag{2}$$

where w_j indicates the weight of indices.

Since the matrix W_{n*1} cannot be multiplied by the normalized decision matrix $(n * n)$, the weight matrix has to be converted into a diagonal matrix W_{n*n} (while the weights are on the main diameter).

Stage three – Determining the positive and negative ideal solutions

Two virtual options of positive and negative ideals A^* and A^- are defined as Eqs. (3) and (4), respectively. The two options $(A^-,\ A^*)$, in fact, indicate the best and worst solutions.

The positive ideal option

$$\begin{aligned} A^* &= \left\{ \left(\max v_{ij} | j \in J\right), \left(\min v_{ij} | j \in J'\right) | i = 1, 2, \ldots, m \right\} \\ &= \left\{ v_1^*, v_2^*, \ldots, v_j^*, \ldots, v_n^* \right\} \end{aligned} \tag{3}$$

The negative ideal option

$$\begin{aligned} A^- &= \left\{ \left(\min v_{ij} | j \in J\right), \left(\max v_{ij} | j \in J'\right) | i = 1, 2, \ldots, m \right\} \\ &= \left\{ v_1^-, v_2^-, \ldots, v_j^-, \ldots, v_n^- \right\} \end{aligned} \tag{4}$$

$j \to J = \{j = 1, 2, 3, \ldots, n\}$ js related to the profit index; $j \to J' = \{j = 1, 2, 3, \ldots, n\}$ js related to the cost index; where v_{ij} is a value of the normalized and weighted decision matrix.

Stage four – Obtaining distances

The distance to the next option n is calculated using the Euclidean method. In this stage, the distance to option i from the positive ideal options is obtained using Eq. (5), while the distance to option i from the negative ideal options is obtained using Eq. (6)

$$S_{i*} = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^*\right)^2}\ \ i = 1, 2, 3, \ldots, m \tag{5}$$

where S_{i*} is a Euclidean distance from the positive ideal options

$$S_{i-} = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^-)^2}\ \ i = 1, 2, 3, \ldots, m \tag{6}$$

where S_{i-} is the Euclidean distance from the negative ideal options.

Stage five – Calculating proximity relative to the ideal solution

D=

	Weight: 0.3 distance to the destination	Weight: 0.4 remaining energy	Weight: 0.1 number of Job	Weight: 0.2 number of neighbors
Node 1	1	200	1000	2
Node 2	2.5	100	1800	4
Node 3	2	150	1500	5
Node 4	3	100	2000	3
Node 5	4	400	2300	7

Figure 3. The decision matrix for five mobile nodes.

This index can be obtained using Eq. (7):

$$C_{i*} = \frac{s_{i-}}{s_{i*} + s_{i-}} \quad 0 < C_{i*} < 1 \;\; m \tag{7}$$

where C_{i*} indicates the relative proximity to the ideal solution.

It can be observed that if $A_i = A^*$ and if $A_i = A^-$, then C_{i*} =0. It is obvious that the shorter the distance A_i is from the ideal solution, the relative proximity will be closer to 1.

Stage six – Ranking the options

Finally, the options are ranked in a descending manner, and from among 25 nodes, five nodes that have the highest desirability are selected as the relaying nodes. The nodes selected for this purpose send an advertisement message within their range, and other nodes that are not relaying ones send communication messages to the nearest relaying node. In this manner, the routing tree is formed.

In the following, a numerical example of the TOPSIS algorithm will be performed by selecting five nodes from among 25 nodes, and the algorithm will be conducted on them.

Stage zero – Obtaining the decision matrix

It is assumed that the indices of the number of neighbors and the remaining energy from the profit index node and the indices of the distance from the mobile node to the destination and the number of jobs form the cost index node (see Figure 3).

Normalizing the decision matrix (see Figure 4).

Weighing the decision matrix – Assume that the weights assigned for the job, the number of neighbors, the remaining energy, and the distance to the destination indices are 0.1, 0.2, 0.4, and 0.3, respectively (see Figure 5)

$$\mathrm{W} = (\mathrm{w}_1,\ \mathrm{w}_2,\ \mathrm{w}_3,\ \mathrm{w}_4) = (0.3, 0.4, 0.1, 0.2)$$

Three – Determining the positive and negative ideal solutions

$$V^{*+} = \left(\min_i v_{i1},\ \max_i v_{i2},\ \min_i v_{i3},\ \max_i v_{i4} \right)$$

R=

	Weight: 0.3 distance to the destination	Weight: 0.4 remaining energy	Weight: 0.1 number of Job	Weight: 0.2 number of neighbors
Node 1	0.16609096	0.406138466	0.25173668	0.197065856
Node 2	0.415227399	0.203069233	0.453126024	0.394131711
Node 3	0.332181919	0.30460385	0.37760502	0.492664639
Node 4	0.498272879	0.203069233	0.50347336	0.295598783
Node 5	0.664363839	0.812276932	0.578994364	0.689730495

Figure 4. The normalized matrix for five mobile nodes.

V*+	0.049827288	0.324910773	0.02517	0.137946099

V=

	Cost distance to the destination	Benefit remaining energy	Cost number of Job	Benefit number of neighbors
Node 1	0.049827288	0.162455386	0.025173668	0.039413171
Node 2	0.12456822	0.081227693	0.045312602	0.078826342
Node 3	0.099654576	0.12184154	0.037760502	0.098532928
Node 4	0.149481864	0.081227693	0.050347336	0.059119757
Node 5	0.199309152	0.324910773	0.057899436	0.137946099

Figure 5. The normalized and weighted matrix for five mobile nodes.

$$V^{*-} = \left(\underset{i}{\max} v_{i1}, \ \underset{i}{\min} v_{i2}, \ \underset{i}{\max} v_{i3}, \ \underset{i}{\min} v_{i4} \right)$$

V*-	0.199309152	0.081227693	0.0579	0.039413171

Four – Calculating the distance

The distance of the *i*th option from the positive ideal

$$S_{i*+} = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^*\right)^2} , \quad i = 1, 2, 3, 4, 5$$

Si*+
0.190001
0.262428
0.213147
0.275971
0.153022

The distance of the *i*th options from the negative ideal

$$S_{i-} = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^-)^2}, \quad i = 1, 2, 3, 4, 5$$

Si*-
0.173245
0.085429
0.124424
0.054112
0.26285

Five – Calculating the relative proximity to the ideal solution

$$C_{i*} = \frac{s_{i*-}}{s_{i*+} + s_{i*-}}$$

Ci*
0.476935
0.245586
0.368585
0.163936
0.632045

Six – Ranking the options

The ranking of options is performed in a descending manner C_{i*}.

Ranking	
4	Node 1
2	Node 2
3	Node 3
1	Node 4
5	Node 5

In the above figure, nodes number 5, 1, and 3 have more desirable conditions compared to the other mobile nodes, and they will be selected as the relaying nodes.

Stage two – The routing phase

In this phase, a protocol for routing in MANETs is presented that can use the Cuckoo Optimization Algorithm to select the shortest path between the relaying nodes and the replacement path. As it was mentioned before, this phase uses the Cuckoo Optimization Algorithm to select the shortest path within the routing tree. The Cuckoo Optimization Algorithm starts with a population of cuckoos, which lay a number of eggs in the nest of host birds. Some of these eggs are more similar to the eggs of the host birds, and consequently, they have higher chances of growing and becoming grow-up birds. On the other hand, the eggs with less similarity are identified by the host birds and are destroyed. The number of eggs grown within an area indicates the fitness of nests in that place. In other words, the number of eggs that have survived is the profit index of a particular area. Thus, a situation where more eggs survive is a situation that will optimize the Cuckoo Optimization Algorithm (Rajabioun 2011). The cuckoos try to find the most appropriate place for laying their eggs in order to increase their survival rate. After the remaining eggs grow and become grown-up birds, they form communities. Each group has a particular domain to live in, and the best domain among all groups will become the next destination of the cuckoos in other groups. For this purpose, they will migrate toward the best habitat and will reside close to that place. Based on the number of eggs that each cuckoo has and the distance of a cuckoo to the target (the best habitat), several egg-laying radii. Then, the bird starts to randomly lay eggs inside nests that are within the radii. This process continues until the best position with the maximum profit value is obtained, and the majority of the cuckoos gather around that point. In the following, this phase of the following method will be explained.

To solve the problem of optimization, it is necessary that the values of the variables in the problem are formed into an array. In the Cuckoo Optimization Algorithm, this array is called "Habitat". In an optimization problem, the next position or Nvar of a habitat is indicated with 1 * Nvar, which shows the current position of the cuckoos. This array has been defined in Eq. (8)

$$\text{Nvarhabitat} = [x1, x2, x3, \ldots x\text{Nvar}] \tag{8}$$

Each value of the variable ($x1$, $x2$, $x3, \ldots x$ Nvar) indicates the number of floating points.

The fitness of the current habitat is obtained through evaluating the profit function $\boldsymbol{f_p}$ of that habitat (Eq. (9))

$$\text{profit} = \boldsymbol{f_p}(\textbf{habitat}) = \boldsymbol{f_p}(\boldsymbol{x}1, \boldsymbol{x}2, \boldsymbol{x}3, \ldots \boldsymbol{x}\textbf{Nvar}) \tag{9}$$

As it was mentioned before, the maximum profit function is used in the Cuckoo Optimization Algorithm. To apply this algorithm in the minimization algorithm (e.g. finding the shortest path), a minus sign has to be multiplied by the cost function (Eq. (10))

$$\text{Profit} = -\text{Cost}(\text{habitat}) = -\text{fc}\ (x1,\ x2,\ \ldots,\ x\text{Nvar}) \tag{10}$$

To start the optimization algorithm, first, a Habitat matrix with the size of Nvar $\times$ Npop is produced. Then, a random number of eggs is assigned to each habitat. In nature, each cuckoo lays between 1 and 22 eggs. These

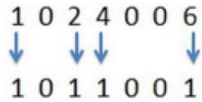


Figure 6. Converting the position of the cuckoos into binary strings.

numbers are used as the lower and upper limits for the assignment of eggs to each cuckoo in different iterations. In the routing problem, the number of these variables Nvar is the same as the number of relaying nodes in the routing tree. Thus, the relaying points on the routing tree have to be coded. To each relaying node selected in the previous phase, a number ranging from 1 to *n* is assigned. To produce the initial population, a number ranging from 1 to *n* has to be assigned to each cuckoo. These number has to form a string so that a path can be shown. However, two conditions have to be observed while selecting these numbers: first, the first and the last variable or relay are constant and are the source and destination relaying nodes. Second, the selection of a node has to be in a way that each relay is connected to the previous relay.

The fundamental point in the production of the initial population is that to solve the problem of the shortest path, the length is considered as a string that is called the position of the cuckoos in the Cuckoo Optimization Algorithm. In the problem of the shortest path, since moving from the source toward the designation is not performed on a similar number of relaying nodes, the main challenge is determining the length of the position of the cuckoos. To produce the initial population, first, a matrix of network connections is made (i.e. anywhere that a relaying node is connected to another relay is indicated with 1, otherwise it is indicated with 0). Then, a string with the length of the variables of the problem – the number of relays in the network – is formed. To select the next relaying node, first, it is determined which relays are connected with other relaying nodes, and then one of the relays connected to a node under consideration is randomly selected. These steps continue until the destination node is reached. However, if along the path some relays are observed that are not connected to other relaying nodes, they are atomically documented as the destination relays. In such a situation, a string with the length of the variables, with two points in the beginning and the end, is formed. However, this string is then eliminated from the population set as a useless string. After the strings related to the position of the cuckoos are formed, they have to be converted into binary strings. For this purpose, along the created path, 1 is used whenever there is a relay, and otherwise, 0 is used. For instance, for a string of the position of cuckoos with the length of 6, its binary string is formed in the shape of 6 (see Figure 6).

In an optimization problem with the upper limit of var_{hi} and the lower limit of var_{low}, each cuckoo will have an Egg-Laying Radius (ELR), which corresponds to the total number of eggs, the number of current eggs, and the upper and lower limits of the variables in the problem. Thus, the ELR can be defined as Eq. (11):

$$\text{ELR} = \alpha \times (\text{Number of cuckoo's eggs/Total number of eggs}) \times (\text{varlow} - \text{varhi}) \tag{11}$$

where α is a variable that regulates the maximum value of ELR.

After the initial population was produced, each cuckoo created in the range of the minimum and maximum eggs is assigned with a random number of eggs. Then, the set of cuckoo eggs is calculated. At this stage, the ELR of the cuckoos has to be determined using Eq. (11). The cuckoos start to lay eggs randomly within the limits of their ELR. Of course, this process is performed in a way that each cuckoo searches a wider sphere within its ELR. In the end, the new positions of the eggs are converted into 0 and 1. The repeated eggs are eliminated since only one egg can be in a particular position. In addition, when the number of the produced eggs exceeds the limit, those eggs that do not have proper values are eliminated. Then, the cuckoo that is at the best location is selected for the production of the next generation. In the proposed method, the sigmoid function is used for the mapping of the new location into the 0 and 1 range (Eq. (12)) so that the new location can be suitable for the binary space. According to Eq. (12), the value of the location is changed into a binary one

$$S = 1/(1 + e^{-x} \text{ NextHabitat}) \tag{12}$$

If $S \leq$ rand Then X NextHabitat $= 0$
If $S >$ rand Then X NextHabitat $= 1$
where X NextHabitat indicates the new location of the cuckoos after migration, which is performed according to the sigmoid function. Here, first, the current location of the cuckoos is converted into 0 or 1 using this function. Then, the obtained results are shown as 0 or 1 by applying the coefficient of mobility and the distance between the current and the target locations. The coefficient of mobility is used so that the cuckoos do not migrate directly to the target location, and have some deviance to search the domain in a wider manner. The best cuckoo obtained at this stage is in fact the best path that will be used to produce the next generation. After their migration, the cuckoos lay eggs once again, and the previous stages are repeated. This process continues until the target location is equal or close to the optimal value. Then, the termination condition, which is 100

iterations, is met. Since the obtained result is a binary one, it must be converted to actual figures so that the obtained path can be identified in the network.

At the end of the algorithm, once again it is investigated whether the obtained result is an actual path and if there is a path between the two points. After finding the paths, the two paths that are shorter than the others are selected; the first path is used for sending data packages, while the second one is saved as a replacement path so that it can be used in times of the failure of the first path.

Stage three – Path maintenance

Since nodes in these networks can be displaced or turned off altogether, the overall structure of a network changes. For instance, if the relaying node G is turned off, node A will not be informed that the path used for reaching the destination is no longer accessible. This protocol has to solve this problem in some way. Each node of the network alternately sends the Hello message, which contains identification and information regarding its location, to its neighbor nodes. The neighbors are expected to respond. However, if no response was made, the sender gets informed that its neighbors are not in the range, and no communication can be made with it any longer. Similarly, if it sends an ordinary package to its neighbor and receives no response, it gets informed that the neighbor is not accessible.

These pieces of information can be used to eliminate paths that no longer function. Each cluster head node N keeps a list for each one of its neighbors that have sent packages within ΔT to the destination nodes. These packages are typically called the "active cluster heads of the cluster head N". Thus, node N has a table whose key is the destination of the network nodes; also, the next node to reach the destination, the number of steps to reach it, the list of active neighbors, and the lifetime are included in this list. Whenever one of the neighbors of the relaying nodes gets inaccessible, the cluster head node checks its routing table to see which one of the network nodes has used the eliminated node in their paths. The fact that all active paths growing through node G are invalid gets spread among them. In addition, the active neighbors inform their active neighbors, and this process goes on until all paths dependent on the recently disappeared path are eliminated from the routing table. Thus, the replacement path, which had been saved in the second phase as the replacement path, is used instead.

4. The Simulation of the Proposed Method

4.1. Simulation Environment

In the current study, the Opnet Modeler 10.5 (http://www.opnet.com) was used to simulate the proposed method and compare it with the four

Table 1. Simulation parameters.

Parameter	Value
Number of nodes	25
Size of simulation environment	1000 × 1000 × 1000 m^3
Radio transmission range	250 m
Packet size	1024 bytes
Transmission type	Constant Bit Rate
Simulation time	200 s
Initial energy value	200–400 Joules
Mobility speed	10 M/s
Rotation duration	200 s

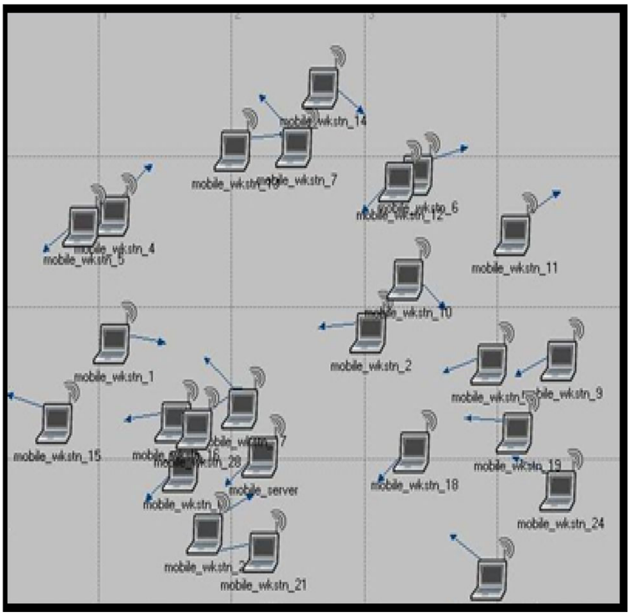
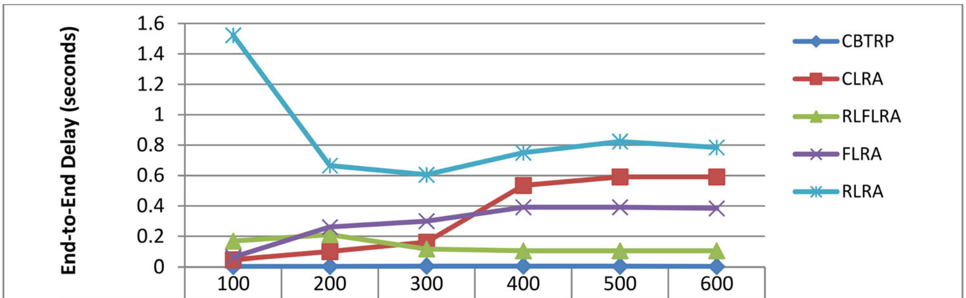


Figure 7. A schematic illustration of the editor of the simulated network model.

methods proposed in Tabatabaei and Behravesh (2017). The simulation parameters are presented in Table 1.

As it can be observed in Figure 7, the integration of a network consisting of 25 has been considered where five scenarios – in the first scenario that is called CBTRP1, the mobile nodes are distributed randomly within the space – are routed using the proposed method (the Cuckoo Optimization Algorithm, the TOPSIS multi-criteria decision making, and fuzzy logic). According to the scenarios proposed in Tabatabaei and Behravesh (2017), scenarios 2–5 perform routing in the MANET in a random fashion based on the CLRA1, RLFLRA2, FLRA3, and RLRA4 algorithms, respectively. For each one of the five scenarios, a similar integration was assumed at the beginning of the simulation. In the following, the results obtained from the simulation of the proposed results will be discussed according to the scenarios.



	100	200	300	400	500	600
CBTRP	0.004247	0.004481	0.006045	0.006065	0.005643	0.004254
CLRA	0.04672	0.10016	0.16259	0.53496	0.59121	0.59121
RLFLRA	0.16906	0.2111	0.11694	0.10499	0.10499	0.10499
FLRA	0.06349	0.26169	0.30021	0.39168	0.39168	0.38428
RLRA	1.5229	0.66544	0.60535	0.74981	0.82372	0.78475

Pause Time

Figure 8. The end-to-end delay.

4.2. Criteria to Evaluate the Efficiency of the Proposed Method

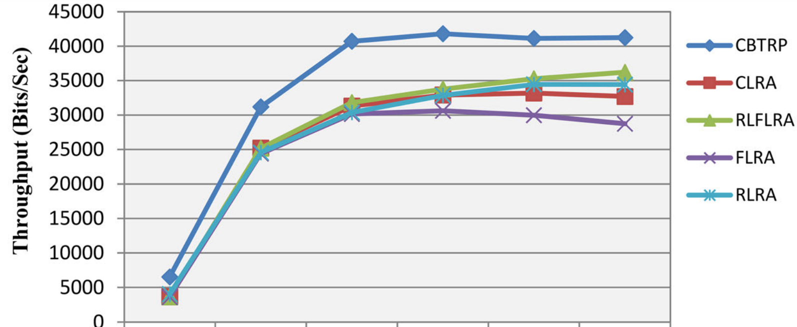
The following criteria were applied to evaluate the efficiency of the following method.

End-to-end delay: It refers to the time taken for a data package to be transmitted from the sender to the receiver. To calculate the average end-to-end delay, the end-to-end delay of all packages received by the senders is measured, and then these values are averaged.

The throughput: It refers to the total number of packages received by the receivers divided by the time difference between receiving the first and last packages (i.e. the size of the file divided by time in terms of Mbps).

4.3. Simulation Results

Figure 8 illustrates the comparison between the end-to-end delay for the proposed algorithm and the scenarios proposed in Tabatabaei and Behravesh (2017). The vertical axis indicates the end-to-end delay, while the horizontal one shows the simulation time. As it can be observed since the proposed protocol is capable of determining proper links for data transfer according to such stability parameters as the remaining energy and distance to the destination, the created routes deliver the packages to the destination with very high rates of success, and there is no need for the repeated discovery of the path. Consequently, the overall delay of the network decreases significantly compared to the other four methods.



	100	200	300	400	500	600
CBTRP	6512	31183.62	40704.65	41804.48	41127.55	41243.63
CLRA	3668.59	25198.9	31308.2	32914.1	33201.9	32693.4
RLFLRA	3654.67	25192.3	31816.7	33767.1	35271.2	36210
FLRA	3624	24413.6	30163.6	30622.9	29979.6	28740.7
RLRA	4036.15	24569.8	30371.3	32853.7	34450.1	34428.8

Pause Time

Figure 9. The rate of throughput.

Figure 9 shows the rate of throughput. The horizontal axis is the simulation time, while the vertical one indicates the number of packages delivered in a particular time (or the throughput). According to Figure 9, the proposed protocol was able to determine proper link with high stability through selecting proper relaying nodes by the application of the TOPSIS algorithm and the Cuckoo Optimization Algorithm as well as such criteria as the number of jobs and neighbors, distance to the destination, and the battery-level energy, and transmit the data from that link if it was found favorable. Thus, the throughput found for this scenario was higher than the ones found for the scenarios proposed in Tabatabaei and Behravesh (2017). In the proposed protocol, the stable path is not changed until the end of the data transmission phase, so the number of packages delivered to the destination node will be higher. Also, the rate of throughput in the RLFLRA was found to be higher than the values obtained for the CLRA, FLRA, and RLRA. Most routing protocols proposed for the MANETs assume mutual links between the nodes. However, in case that an ad hoc network has been formed from heterogeneous nodes with varying energy levels and transmission ranges, some nodes may be able to receive the data transmitted from other nodes without being capable of sending them to other nodes. According to Figure 9, the RLFLRA can be used to determine proper links after the training, and send the data through those links if they are found to be proper. Thus, the rate of throughput in these scenarios was found to be higher than the other scenarios proposed in Tabatabaei and Behravesh (2017).

5. Conclusion

In the current study, energy consumption was investigated as one of the most challenging issues faced by ad hoc networks in their various applications. Then, a new routing method was proposed using the Cuckoo Optimization Algorithm and the TOPSIS multi-criteria algorithm to select the next proper node for data routing. The proposed method was simulated using the OPNET on the algorithms proposed in Tabatabaei and Behravesh (2017), and the results were obtained in terms of the end-to-end delay and the rate of throughput to investigate the performance of the proposed method. In general, it was found that the proposed method performs better than the RLRA, FLRA, CLRA, and RLFLRA. In other words, the method was able to enhance the overall efficiency of the network and improve the reliability of data package delivery through the selection of more stable paths where nodes had higher rates of energy.

ORCID

Shayesteh Tabatabaei http://orcid.org/0000-0001-5513-6165

References

Bhattacharya, A., and K. Sinha. 2017. An efficient protocol for load-balanced multipath routing in mobile ad hoc networks. *Ad Hoc Networks* 63:104–14. doi:10.1016/j.adhoc.2017.05.008.

Bilgin, Z., and B. Khan. 2010. A dynamic route optimization mechanism for AODV in MANETs. In 2010 IEEE International Conference on Communications (ICC), May, 1–5. IEEE.

Clausen, T., J. Yi, and U. Herberg. 2017. Lightweight On-demand Ad hoc Distance-vector Routing-Next Generation (LOADng): Protocol, extension, and applicability. *Computer Networks* 126:125–40. doi:10.1016/j.comnet.2017.06.025.

He, J., O. Yang, and Y. Zhou. 2018. Using flow cost to globally allocate and optimize limited bandwidth in multipath routing. *Computer Networks* 132:15–25. doi:10.1016/j.comnet.2017.12.009.

Junhai, L., X. Liu, and Y. Danxia. 2008. Research on multicast routing protocols for mobile ad-hoc networks. *Computer Networks* 52 (5):988–97. doi:10.1016/j.comnet.2007.11.016.

Raich, A., and A. Vidhate. 2013. Best path finding using location aware AODV for MANET. *International Journal of Advanced Computer Research* 3 (3):336.

Rajabioun, R. 2011. Cuckoo Optimization Algorithm. *Applied Soft Computing* 11 (8):5508–18. doi:10.1016/j.asoc.2011.05.008.

Tabatabaei, S., and R. Behravesh. 2017. New approaches to routing in mobile ad hoc networks. *Wireless Personal Communications* 97 (2):2167–90. doi:10.1007/s11277-017-4602-8.

Tabatabaei, S., M. Teshnehlab, and S. J. Mirabedini. 2015. Fuzzy-based routing protocol to increase throughput in mobile ad hoc networks. *Wireless Personal Communications* 84 (4):2307–25. doi:10.1007/s11277-015-2706-6.

Upadhyay, S., P. Joshi, N. Gandotra, and A. Kumari. 2012. Comparison and performance analysis of reactive type DSR, AODV and proactive type DSDV routing protocol for wireless mobile ad-hoc network, using NS-2 simulator. *Journal of Engineering and Computer Innovations* 3 (2):36–47.

Yadav, A. K., S. K. Das, and S. Tripathi. 2017. EFMMRP: Design of efficient fuzzy based multi-constraint multicast routing protocol for wireless ad-hoc network. *Computer Networks* 118:15–23. doi:10.1016/j.comnet.2017.03.001.