Hybrid Optimization Algorithm Based Cluster Head Selection and Trusted Node Identification for Mobile Adhoc Network

N.Angel[1] and Dr.Krishnaveni Sakkarapani[2]

[1]*Ph.D Research Scholar, PG & Research Department of Computer Science, Pioneer College of Arts & Science, Coimbatore, Tamil Nadu, India*
E-mail: angel.johnson2211@gmail.com

[2]*Assistant Professor, Department of Computer Science, PSGR Krishnammal College for Women, Coimbatore, Tamil Nadu, India*
sss.veni@gmail.com

Abstract: The dynamic nodes topology makes the Mobile Adhoc network (MANET), an attractive solution for data transfer in the wireless domain. The clustering and routing protocols are utilized in MANET and this research work utilizes an optimization algorithm for optimum data transfer. An energy-efficient metaheuristic cluster-based routing protocol with trusted node identification (EEMCR-TNI) was put forward in this work, efficient energy was achieved with increased network lifetime. The quantum glowworm swarm optimization (QGSO) algorithm was employed for the formulation of clusters and cluster head identification. The trusted node identification was done by the ant-lion optimizer. The adhoc on-demand distance vector (AODV) protocol was employed finally to estimate the optimal route between two mobile nodes. The simulation results were found to be superior in the data transfer in wireless topology networks.

Keywords: MANET, Clustering, Routing, Energy Efficiency, Metaheuristics

1. INTRODUCTION

An energy-efficient routing protocol was proposed in [1] for the mobile adhoc network topology, cat slap single-player algorithm (C-SSA) was used in this work. The adaptive weighted clustering protocol (AWCP) was utilized in [2] for the cluster node grouping in vehicular adhoc networks and improved whale optimization was employed for the cluster head selection. A hybrid optimization comprising of grey wolf and crow optimization was employed for the optimum cluster head selection in the wireless network [3]. The performance was found to be proficient when compared with the classical optimization techniques; firefly, artificial bee colony, grey wolf, and firefly cyclic grey wolf optimization algorithms. A hybrid optimization algorithm was employed in [4] for the detection and prevention of spoofing in MANET. In [5], clustering was done using an energy-efficient routing protocol and particle swarm optimization was employed for the selection of cluster head. The general MANET configuration is represented in figure 1.

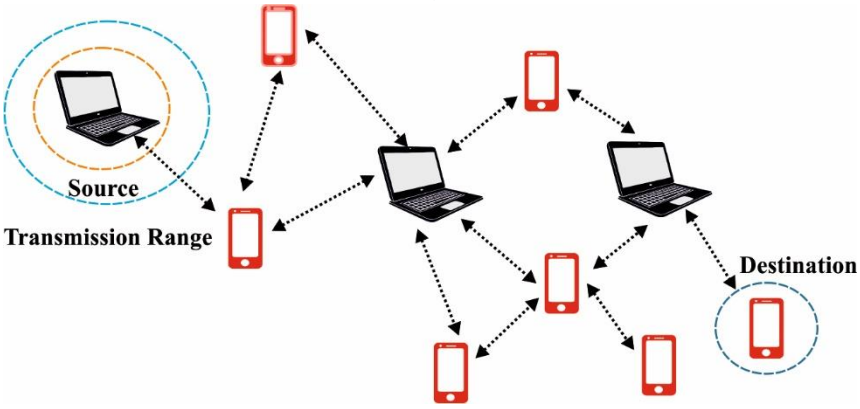

Fig. 1. General MANET configuration

Power optimization plays a vital role in MANET and in [6], a hybrid optimization algorithm was employed for the cluster formulation, power, and energy management using topology management. A hybrid

optimization technique was employed in [7] for spoof detection in mobile adhoc networks. The modified rider optimization algorithm was incorporated in [8] for the cluster head selection in IoT-based wireless sensor networks that find its application in smart cities. Neighborhood failure node detection was done by Environs Aware Neighbor-knowledge based Broadcasting (EANKBB) technique and for cluster head selection, a genetic algorithm was employed [9]. The proficient energy head selection was done by maximum weight independent set (MWIS) and it was solved by the quantum approximate optimization [10]. Section 2 focuses on the proposed algorithm and the results are highlighted in section 3, finally, the conclusion is drawn in section 4.

2. MANET ARCHITECTURE WITH CLUSTER HEAD SELECTION AND TRUSTED NODE IDENTIFICATION

The proposed architecture employed in the MANET is depicted in figure 2. For node clustering, the QGSO algorithm was employed and for the trusted node identification, a hybrid combination of optimization algorithms was employed. The Ant Lion Optimizer was utilized here in this work for trusted node identification.

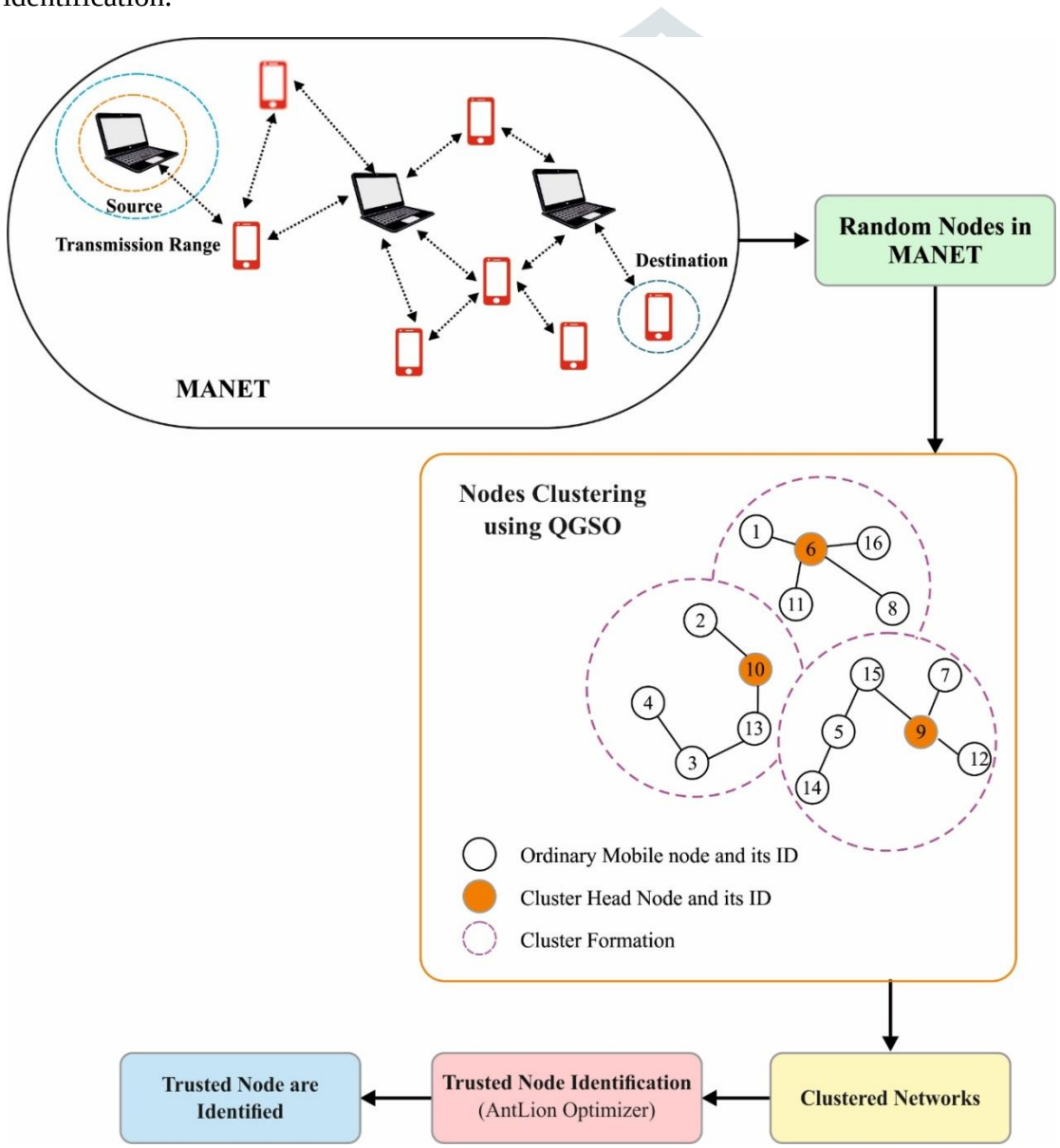

Fig. 2. Proposed architecture for the MANET

2.1 Glowworm swarm algorithm

The GSO algorithm is a swarm-based model in which a group of glow worms is randomly distributed in a search area. The glow worms in consideration emit a luminous substance known as 'luciferin'. Furthermore, the glow worms use a decision-making application to make their own decisions$GM_e^g (0 < GM_e^g \leq GM_u)$. Glowworm communication is based on the fact that glowworm light intensity is precisely proportional to luciferin, and communication is carried out in an adaptive environment. In addition, luciferin intensity is strongly linked to recent position fitness. As a result, the best glow-worm location is discovered when the luciferin intensity is highest. Let g be a glowworm, and w be the glow-worm closest to it.The glow worms are chosen in a precise set-up in which local glow worms with a high luciferin value are compared to recent glow worms and sent towards it. Initialization, Luciferin-update, Movement, and Neighbourhood range Update are the four phases of the GSO model.Initially, the glow worms in this module are spread at random. The glowworms are then made up of the same luciferin intensity as the sane decision domain GM_0.

In general, the intensity of glow-worm luciferin is related to fitness location. The optimal position and goal value are reached when the intensity value is maximum. Alternatively, a target in a minimal condition has been assumed. As a result, the position of the glow-worm differs from the previous iteration while increasing the value of luciferin.The objective function value at the g^{th}glowworm place at t is $(X_g(t))$, and the glowworm position at r time is $\chi_g(\mathrm{f})$.Furthermore, $J(X_g(t))$to $LU_g(\mathrm{t})$ has been established, in which the luciferin level is related with g glowworm at t, and it is described in Eq. (1), where v represents the luciferin decay constant $(0 < v < 1)$, and η represents the luciferin enhancement constant.

$$LU_g(\mathrm{f}) = (1 - v))LU_g(\mathrm{t} - 1) + \eta\big(J(X_g(t))\big) \tag{1}$$

The glowworms choose their neighbor based on a unique chance. The glowworm's neighbor must meet two requirements: the first is that the glowworm in the g glowworm's decision application must be validated with a higher value. Furthermore, g glow worm shifts towards w neighbor, which is inferred from $N_g(\mathrm{f})$and modeled using $T_{g/}(t)$and Eq (2).

$$PT_{gJ}(t) = \frac{LU_w(\mathrm{t}) - LU_g(t)}{\sum_{l \in W_g(ti)} LU_1(\mathrm{t}) - LU_g(\mathrm{t})} \tag{2}$$

Once the g glowworm's action is known, the location is tuned, and the estimation of position update is shown in Eq. (3), where size represents the step size.

$$X_g(\mathrm{t}+1) = X_i(\mathrm{f}) + size * \left(\frac{X_w(\mathrm{t}) - X_g(\mathrm{t})}{\left|\left|X_w(t) - X_g(\mathrm{t})\right|\right|} \right) \tag{3}$$

When the glow worm's position is enhanced, the neighborhood range is updated. The range of the neighborhood then encapsulates the minimal glowworm density, and the range of the neighborhood is increased. Eq. (4) contains the update function, whereλ refers to the constant attribute. The nu_t is used as a variable to regulate the number of nearest neighbors in this procedure.

$$CM_e^g(\mathrm{t}+1) = \min\left\{CM_{\mathrm{u}}, \max\left\{0, GM_e^g(\mathrm{t}) + \lambda|N_g(\mathrm{t})|\right\}\right\} \tag{4}$$

2.2 Quantum Computing in GSO Algorithm

The QGSO algorithm is presented to increase the performance of the classical GSO algorithm by using quantum theory features. Quantum processing is based on the use of data in the form of quantum bits (Q-bits) and emphasizes the advantages of superposition states. The Q-bit is a fundamental data unit in a two-state quantum computer.Quantum states$|\Psi\rangle = \mathrm{a}|0\rangle + b|1\rangle$ are a linear superposition of individual states based on the individual condition $|0\rangle$ and $|1\rangle$. The complex weights of a quantum particle in positions $|0\rangle$ and $|1\rangle$are represented by the probability amplitudes, a and b, respectively. $|\mathrm{a}|^2$ and $|b|^2$ allow the potential of a Q-bit being in

$$|a|^2 + |b|^2 = 1 \quad (5)$$

When a representation is made up of d states, it is assumed to be made up of 2^d states at the same time, and the probability total is 1. As a result, the d-length of Q-bit is merged into the d-length binary vector as follows:

$$if|a_i|^2 < thresholdtheny_i \leftarrow 1elsey_i \leftarrow 0 \quad (6)$$

where y_i is the observed Q-bit's corresponding binary bit. The threshold value is created in a completely random manner. As a result, collections of binary vectors are analyzed to determine fitness.

Best, Random, Roulette, and Tournament are selection operators that are used to pick individuals, resulting in the generation of successive populations. Following that, the mutation operator is used to change the probability of huge Q-bits of an individual as defined below. P_{muf}at a mutation point, for example, is shown as follows for quantum individual P, a mutated individual:

$$P = \begin{pmatrix} a_1 a_2 a_3 a_4 a_5 \dots a_d \\ b_1 b_2 b_3 b_4 b_5 \dots b_d \end{pmatrix} P_{muf=} \begin{pmatrix} a_1 a_2 a_3 a_4 a_5 \dots a_d \\ b_1 b_2 b_3 b_4 b_5 \dots b_d \end{pmatrix} \quad (7)$$

Based on 1-point, n-point crossover, the crossover is performed on two quantum individuals, resulting in the generation of two individuals. As an example, consider a 1-point crossing in the third place between parents P^1 and $P^2$to produce children C^1 and C^2.

$$P^1 = \begin{pmatrix} a_1^1 a_2^1 a_3^1 a_4^1 a_5^1 \dots a_d^1 \\ b_1^1 b_2^1 b_3^1 b_4^1 b_5^1 \dots b_d^1 \end{pmatrix} \quad P^2 = \begin{pmatrix} a_1^2 a_2^2 a_3^2 a_4^2 a_5^2 \dots a_d^2 \\ b_1^1 b_2^2 b_3^2 b_4^2 b_5^2 \dots b_d^2 \end{pmatrix} \quad (8)$$

$$C^1 = \begin{pmatrix} a_1^1 a_2^1 a_3^2 a_4^2 a_5^2 \dots a_d^2 \\ b_1^1 b_2^1 b_3^2 b_4^2 b_5^2 \dots b_d^2 \end{pmatrix} \quad C^2 = \begin{pmatrix} a_1^2 a_2^2 a_3^1 a_4^1 a_5^1 \dots a_d^1 \\ b_1^1 b_2^2 b_3^1 b_4^1 b_5^1 \dots b_d^1 \end{pmatrix} \quad (9)$$

2.3 Ant Lion Optimizer (ALO)

ALO is a new metaheuristic algorithm that mimics antlion hunting behavior. It considers and implements the random movement of ants, building traps, confinement of ants in traps, collecting prey, and re-building traps to mathematically mimic the interaction of ants and antlions in nature.Clearly, ALO refers to the hunting process of ant-lions and ants in a trap, in which ants wander stochastically in search of food and ant-lions catch the ants using traps that are represented in matrix form.

$$M_{Ant} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1d} \\ A_{21} & A_{22} & \cdots & A_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ A_{n1} & A_{n2} & \cdots & A_{nd} \end{bmatrix} \quad (10)$$

$$M_{Antlion} = \begin{bmatrix} AL_{11} & AL_{12} & \cdots & AL_{1d} \\ AL_{21} & AL_{22} & \cdots & AL_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ AL_{n1} & AL_{n2} & \cdots & AL_{nd} \end{bmatrix} \quad (11)$$

The random movement of ants

In all optimization iterations, ants use random moments to enhance their placements. A random moment is determined by the function provided:

$$X(t) = [0, cumsum(2r(t_1) - 1) \quad (12)$$

$$cumsum(2r(t_2) - 1), \dots, cumsum(2r(t_n) - 1),] \quad (13)$$

where r(t) is a stochastic function defined as follows:

$$r(t) = \begin{cases} 1\ if rand > 0.5 \\ 0\ if rand \leq 0.5 \end{cases} \quad (14)$$

The cumulative total is represented as cumsum, n signifies the larger number of iterations, t denotes the current iteration, and rand specifies a random variable generated with an equal distribution from [0,1]. Because the search space is constrained by a boundary, random moments are kept inside it, and ants are generalized by applying the specified function (min-max normalisation) before upgrading their location:

$$X_i^t = \frac{(X_i^t - a_i) \times (d_i - c_i^t)}{(d_i^t - a_i)} + c_i \tag{15}$$

where $a_i$means the lower random walk of the i-th variable, d_i denotes the higher random walk of the i-th variable, c_i^trepresents the low i-th variable at the t-th iteration, and d_i^trepresents the greater i-th variable at the t-th iteration.

Building Trap

An antlion's hunting abilities are modeled using a roulette wheel. Ants are thought to be confined in a single antlion. During optimization, the ALO algorithm must use a roulette wheel operator to select antlions based on their fitness. The technique gives the fitter antlions a better opportunity of trapping ants.

Confinement of ants in traps

Antlion traps have an impact on random moments of ants in this area. The ant's path is now bounded by the position of the chosen antlion, which may be modeled by adjusting the range of the ant's random moment to the antlion's position, as shown in the equations below.

$$c_i^t = Antlion_j^t + c^t \tag{16}$$

$$d_i^t = Antlion_j^t + d^t \tag{17}$$

According to Equations (16) and (17), ants walk randomly in a hypersphere represented by vectors c and d over a defined antlion, where c^t represents the vector with the least number of parameters at the t-th iteration, d^t represents the vector with the most variables at the t-th iteration, c_j^t represents the vector with the least number of variables for the i-th ant, d_j^t represents the maximal variables for the i-th ant, and $Antliont_j$ represents the position of the chosen j-th antlion at the t-th iteration.

Sliding Ants towards Antlion

Antlions build traps and ants migrate in an erratic pattern based on their fitness ratings. Antlions induce sands externally from the middle of a pit when the ants are imprisoned. This nature is inclined, and confined ants want to escape, resulting in a large reduction in the radius of ant's random travels in the hypersphere.

$$c^t = \frac{c^t}{I} \tag{18}$$

$$d^t = \frac{d^t}{I} \tag{19}$$

The above functions reduce the radius of extending ants' location and mimic the sliding operation of ants within pits, where I denote a ratio, c^t denotes the vector with the fewest variables at the t-th iteration, and d^t denotes the vector with the greatest number of variables at the t-th iteration. I=10w (t/T) in Eqs. (18) and (19), where T signifies larger iterations and w specifies a constant from the current iteration t. The constant w changes the level of exploitation.

Catching Prey and Rebuilding traps

When an ant reaches the bottom of a pit, it becomes prey for an antlion. Then, to optimize the choice of hunting new prey, antlion improves the location to the advanced place of hunted ant.

$$Antlion_j^t = Ant_i^t iff Ant_i^t > f\left(Antlion_j^t\right) \tag{20}$$

where $Antlion_j^t$ denotes the location of the j-th antlion at the t-th iteration and Ant_i^t denotes the location of the i-th antlion at the t-th iteration.

Elitism

It's one of the key elements of Evolutionary Algorithms (EA), which are used to keep optimization algorithms simple, elegant, and efficient.Each ant takes a random walk and chooses an antlion using a roulette wheel and an elite simultaneously.

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \quad (21)$$

where R_A^t denotes a random walk over the antlion determined by a roulette wheel at the t-th iteration, R_E^t denotes a random walk from the elite at the t-th iteration, and Ant it denotes the position of the i-th ant at the t-th iteration, and Ant Et denotes the random walk from elite at the t-th iteration. Assume A is a function that creates random initial solutions, B is a function that extends function A's primary population, and C is a function that returns a positive value when the termination condition is met. The ALO technique is represented as a 3-tuple in the following way using the above-mentioned operators:

$$ALO(A, B, C) \quad (22)$$

where the functions A,B, and C are as follows:

$$\emptyset \rightarrow^A \{M_{\text{Ant}}, M_{OA}, M_{Antlion}, M_{OAL}\} \quad (23)$$

$$\{M_{Ant}, M_{Antlion}\} \rightarrow^B \{M_{Ant}, M_{Antlion}\} \quad (24)$$

$$\{M_{Ant}, M_{Antion}\} \rightarrow^C \{true, false\} \quad (25)$$

where M_{Ant} is an ant's matrix position, $M_{Antlion}$ is an antlion's matrix position, $M_{?A}$ is an ant's fitness, and M_{OAL} is an antlion's fitness.

3. RESULTS AND DISCUSSION

The performance of the QGSO-ALO algorithm was validated by the following metrics; packet delivery ratio (PDR), network lifetime (NLT), quality of service (QoS), and energy consumption (EC). The performance validation plots of the OGSO-ALO algorithm for various nodes are represented in figure 4.

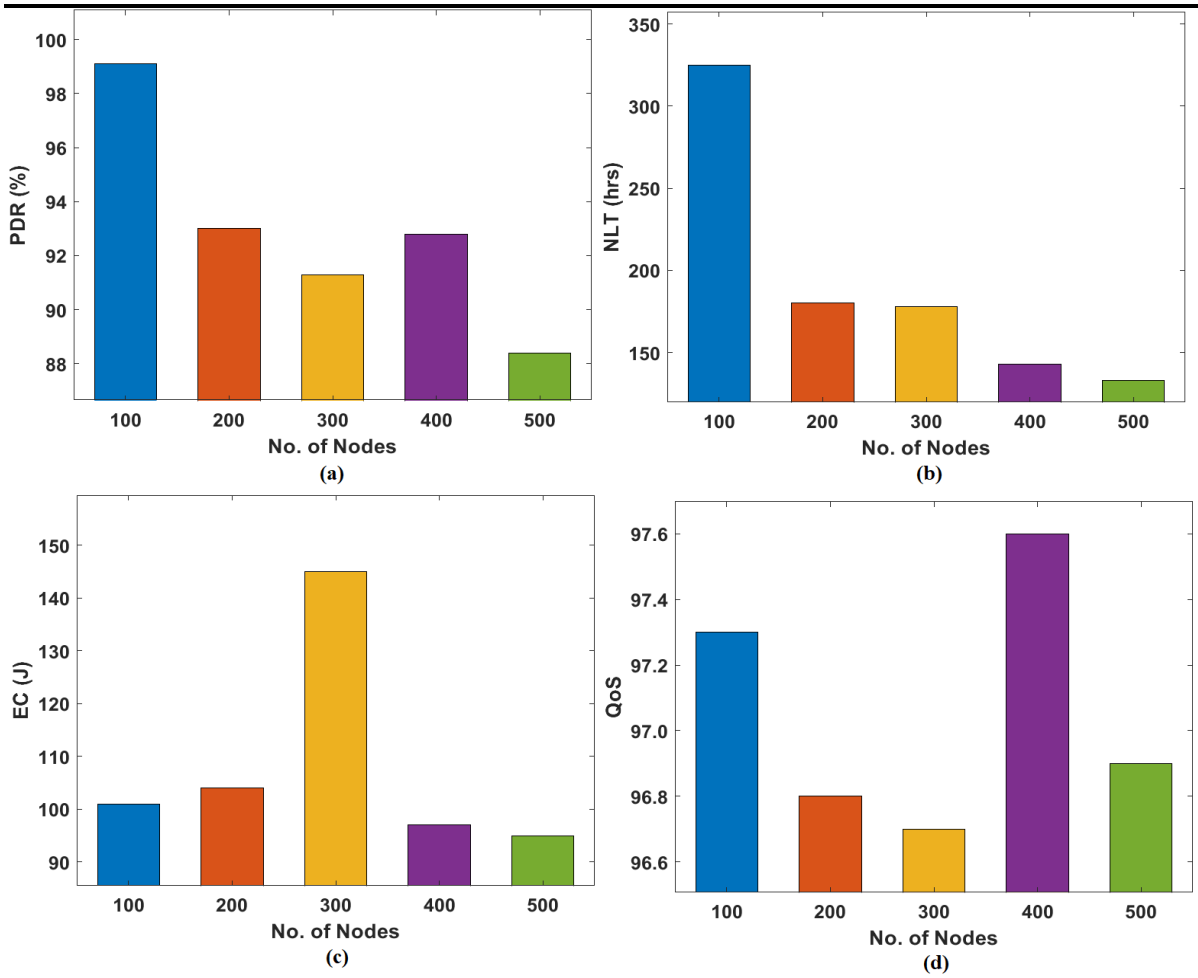

Figure 4: Performance validation of the OGSO-ALO algorithm a) PDR b) NLT c) EC d) QoS

The PDR and NLT of the OGSO-ALO algorithm were found to be superior and the reduced EC with maximum QoS favors its utilization in real-time application. For comparative analysis, the OGSO-ALO algorithm was compared with the HGGSA, IHSO, KHO and BPSO algorithms.

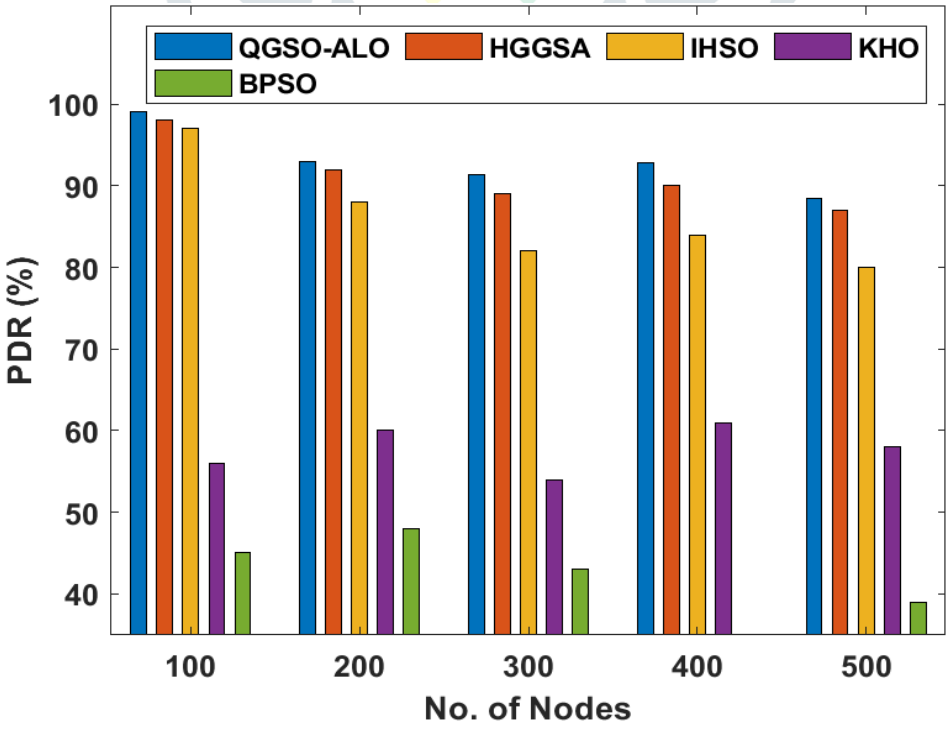

Fig. 5. PDR analysis of QGSO-ALO model with other models concerning variable node values

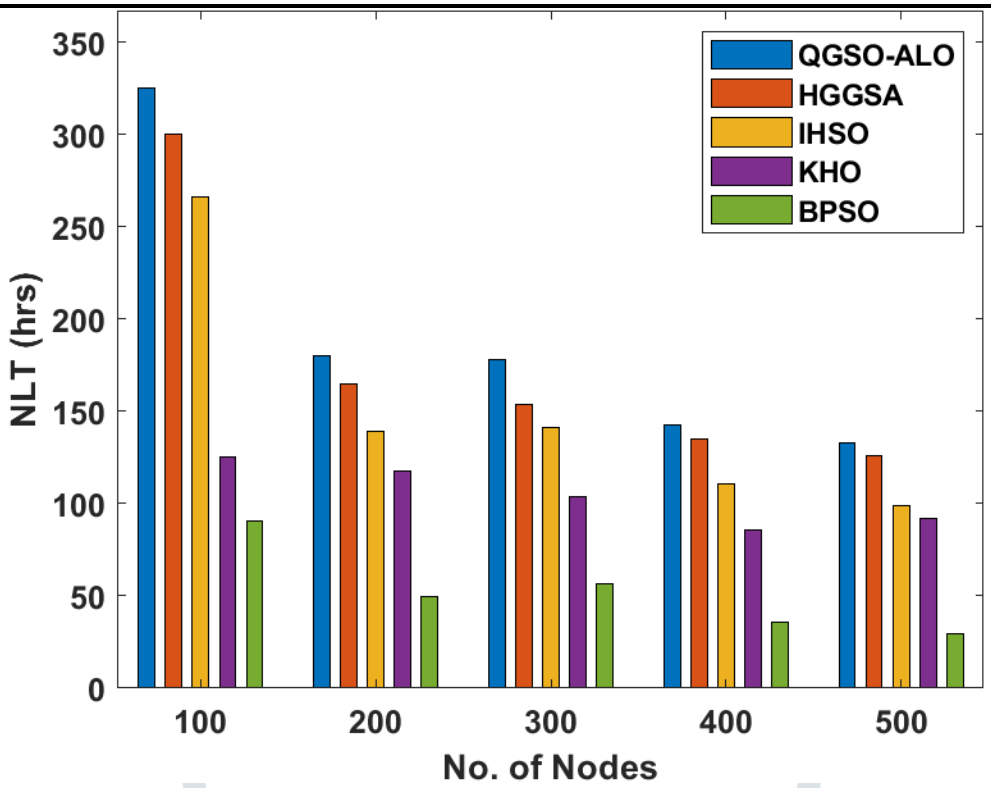

Fig. 6. NLT analysis of QGSO-ALO model with respect to variable node values

The network lifetime of the QGSO- ALO algorithm was found to be proficient when compared with the IHSO, BPSO, HGGSA, and KHO algorithms.

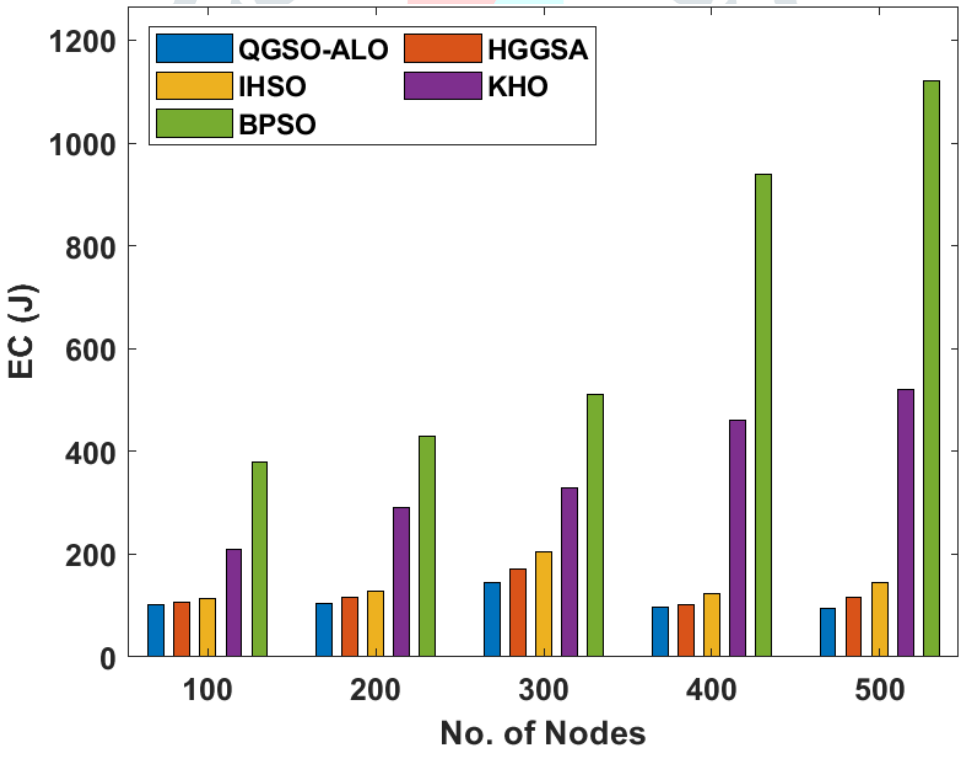

Fig. 7. EC analysis of QGSO-ALO model concerning variable node values

The energy consumption of the algorithms is plotted in figure 7, the QGSO-ALO algorithm has the lowest energy consumption when compared with the IHSO, BPSO, HGGSA and KHO algorithms.

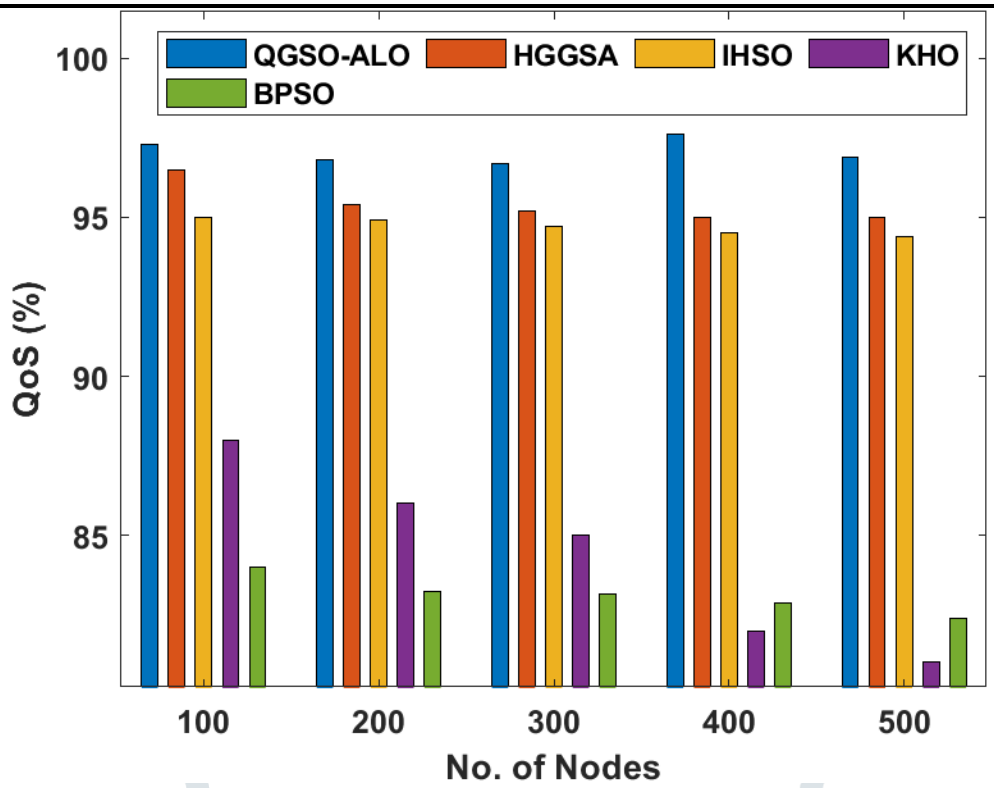

Fig. 8. QoS analysis of QQSO-ALO model concerning variable node values

The QoS plot is depicted in figure 8, the QGSO-ALO algorithm has a high value of QoS when compared with the other algorithms.

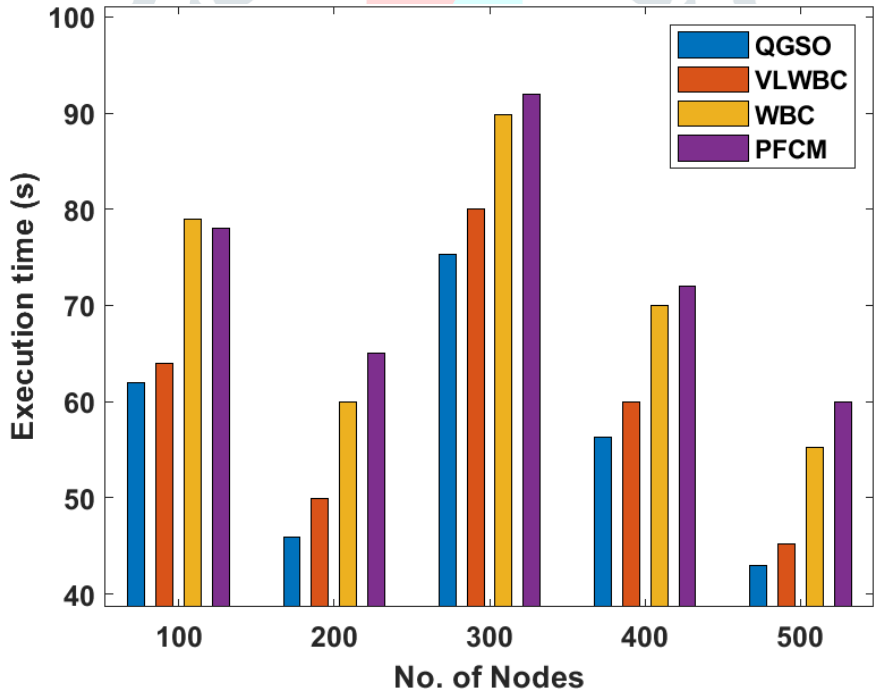

Fig. 9. Execution time analysis of QGSO model concerning variable node values

The execution time of the algorithms is depicted in figure 9. The execution time of the QGSO algorithm was found to be less when compared with the other algorithms.

4. CONCLUSION

The metaheuristic optimization-based cluster node identification and routing were proposed in this research work. A hybrid combination of optimization algorithms was utilized in this research work comprising of quantum glow swarm optimization for cluster node formulation and cluster head identification and ant lion optimization was employed for trusted node identification. The proposed cluster identification and routing

protocols were found to be superior when compared with the classical algorithms in terms of the packet delivery ratio, energy consumption, and network routing load.

REFERENCES

1. Veeraiah N, Khalaf OI, Prasad CV, Alotaibi Y, Alsufyani A, Alghamdi SA, Alsufyani N. Trust aware secure energy-efficient hybrid protocol for manet. IEEE Access. 2021 Aug 30;9:120996-1005.

2. Valayapalayam Kittusamy SR, Elhoseny M, Kathiresan S. An enhanced whale optimization algorithm for vehicular communication networks. International Journal of Communication Systems. 2019:e3953.

3. Subramanian P, Sahayaraj JM, Senthilkumar S, Alex DS. A hybrid grey wolf and crow search optimization algorithm-based optimal cluster head selection scheme for wireless sensor networks. Wireless Personal Communications. 2020 Jul;113(2):905-25.

4. Elhoseny M, Shankar K. Reliable data transmission model for mobile ad hoc network using signcryption technique. IEEE Transactions on Reliability. 2019 Jun 6;69(3):1077-86.

5. Hamza F, Vigila SM. Cluster Head Selection Algorithm for MANETs Using Hybrid Particle Swarm Optimization-Genetic Algorithm. Int. J. Comput. Netw. Appl. 2021 Apr 27;8(2):119-29.

6. Devika B, Sudha PN. Power optimization in MANET using topology management. Engineering Science and Technology, an International Journal. 2020 Jun 1;23(3):565-75.

7. Visalakshi P, Prabakaran S. Detection and prevention of spoofing attacks in mobile adhoc networks using hybrid optimization algorithm. Journal of Intelligent & Fuzzy Systems. 2020 Jan 1;38(2):1691-704.

8. Alazab M, Lakshmanna K, Reddy T, Pham QV, Maddikunta PK. Multi-objective cluster head selection using fitness averaged rider optimization algorithm for IoT networks in smart cities. Sustainable Energy Technologies and Assessments. 2021 Feb 1;43:100973.

9. Banumathi J, Kanthavel R. Node Failure Aware Broadcasting Mechanism in Mobile Adhoc Network Environment. Programming and Computer Software. 2018 Nov;44(6):371-80.

10. Choi J, Oh S, Kim J. Energy-efficient cluster head selection via quantum approximate optimization. Electronics. 2020 Oct;9(10):1669.