A Quality of Service (Qos) Load Balanced Hybrid Simulated Annealing -Stochastic Diffusion Search (Sa-Sds) Network Backbone for Manet

KAVITHA V.R (✉ vr.kavitha.cs@gmail.com)

Prathyusha Engineering College <https://orcid.org/0000-0003-2201-2734>

Moorthi M

Saveetha Engineering College

Research Article

Keywords: Mobile Ad hoc Networks (MANET), Multi-path Routing, Quality of Service (QoS), Connected Dominating Set (CDS), Minimum Dominating Set (MDS), Simulated Annealing (SA) with Stochastic Diffusion Search (SDS)

Posted Date: March 8th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-173370/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

A QUALITY OF SERVICE (QoS) LOAD BALANCED HYBRID SIMULATED ANNEALING -STOCHASTIC DIFFUSION SEARCH (SA-SDS) NETWORK BACKBONE FOR MANET

Dr. V.R.Kavitha Corresponding author

Associate Professor

Computer Science and Engineering

Prathyusha Engineering College

Tiruvallur, Tamil Nadu, India.

E.Mail: vr.kavitha.cs@gmail.com

Dr.M.Moorthi

Professor

Department of Medical Electronics - ECE

Saveetha Engineering College,

Chennai, Tamil Nadu, India

E-Mail : moorthidmp@gmail.com

ABSTRACT

The Mobile Ad hoc Networks (MANET) are those networks that do not have the infrastructure and are formed dynamically by means of an autonomous system of some mobile nodes connected through wireless links. All routers are left free to be able to randomly move and arbitrarily organize themselves. So, the wireless topology of the network can have unpredictable and rapid changes. In these types of networks, the provisioning of services based on Quality of Service (QoS) can pose to be very challenging. The work further presented a newer approach that was based on a hybrid Simulated Annealing (SA) along with a Stochastic Diffusion Search (SDS) based multi-path routing network which backbones in giving support to the enhanced QoS in the MANETs. This multipath routing had the objective of improving the dependability and the throughput along with load balancing. The SA is used for solving the problem of the Minimum Dominating Set (MDS). This SDS heuristic gives an algorithm which is simple in its structure and also provides a high exploration level along with fast convergence in comparison with the other algorithms. SA algorithms are also used for improving the diversity of agent and for avoiding it from being trapped within the local optimum. The results of the experiment proved that the SA-SDS method proposed had a better performance compared to the Connected Dominating Set (CDS)-SA.

Keywords: *Mobile Ad hoc Networks (MANET), Multi-path Routing, Quality of Service (QoS), Connected Dominating Set (CDS), Minimum Dominating Set (MDS), Simulated Annealing (SA) with Stochastic Diffusion Search (SDS).*

1 INTRODUCTION

The Mobile Ad hoc Network (MANET) denotes a collection of various mobile nodes that are connected through wireless links thus forming a network without infrastructure or a central administrator. The applications in which the MANET is employed are data

acquisition, emergence and search rescue in the military field. The network topology will also change in a dynamic manner as the wireless nodes can move in an arbitrary manner. All these mobile nodes act to be the sender, the receiver and the intermediate router that depends on the network and its situation. Every node contains a limited power in terms of battery and this further gets reduced with time. The primary traits of the MANET were that it operates without any central coordinator and is rapidly deployable with constraint resources (such as the lifetime of battery, computing power and bandwidth) breakage of a link owing to mobile nodes, multi-hop communication and is also self-configuring. Thus, the primary challenges in the routing protocol found in the MANET was that it had to be distributed fully, it should be adaptive to frequent change in topology, have easy maintenance and computation, should have a loop free route and be optimal and finally should provide the QoS with a minimum collision.

There have been several issues that need consideration while deploying MANETs. Given below are some main issues. 1. The environmental unpredictability: there are ad hoc networks that are deployed in some of the unknown terrains, the hazardous conditions and some hostile environments in which either tampering or actual destruction of nodes are imminent. Based on the environment, there may be frequent node failures. 2. Wireless medium unreliability: any communication using a wireless medium was unreliable and was also subject to errors. Further, owing to the environmental conditions like the high Electro-Magnetic Interference (EMI) levels or inclement weather, the wireless link quality gets unpredictable. 3. Nodes that are resource constrained: the nodes found in a MANET are ideally powered by a battery and also has limited processing and storage capacity. Furthermore, they can also be located in areas where recharge may not be possible and thus have a limited lifetime. 4. Dynamic topology: Topology in the case of an ad hoc network can change constantly owing to node mobility. As there is an in and out movement by the nodes in terms of the range, there are some links that break when there are newer links being created between nodes [1].

Owing to the requirements of the application and for a data transfer that is reliable, load balancing becomes a key area of research in the MANETs. In the MANET, completing a job can get complex and whenever there is a large load that is given to the nodes that have low capabilities of processing they may not have the means of load sharing. Possibilities of load imbalance are owing to the processing or computing power of systems that is non-uniform. There are some more situations in which there are only a few nodes that remain idle and some may be overloaded. The node that has the highest processing power will complete its work fast and will have a low load or no load. Thus, in the case of loads being under-loaded and idle, the requirement of an over-loaded node may become objectionable. There are plenty of lots in routing approaches that have been developed for the purpose of load balancing in the MANETs. Any routing protocol found in the MANET will have to distribute fairly all routing tasks among its mobile host. If the load distribution or traffic is unbalanced, it can result in degradation of performance of the network. Owing to this unbalancing aspect, there are a few nodes within this network that are loaded with some routing duties that result in large-sized queues, high consumption of power, a high ratio of packet loss and a high

packet delay. The problem can result in a new algorithm of load balancing routing in the MANET.

Recently, the approaches to multipath routing have been introduced for overcoming limitations in single path routing. There are multiple routes found between a source destination pair. The benefits of multipath approaches are that it has a bandwidth of higher utilization, and end-to-end delay that is lower, a network life that is higher and higher throughput. It further had a provision to employ load balancing by means of carrying traffic using multiple paths. This brings down the congestion to a network and also protects against any route failure. Aside from these benefits, the multipath is connected to certain issues that are: (i) how multiple paths can be discovered and (ii) how load can be distributed among multiple paths. In the case of multipath approach there are many similarities to a single path approach. The protocols of multipath routing have a preference to disjoint paths. The paths may be found disjoint in two different ways. They are: (a) the link-disjoint, and (b) the node-disjoint. The node disjoint paths will not have any common nodes except in the case of a source and a destination. They do not have any common links as well. Contrastingly, the link-disjoint will not have common links but can have some common nodes. On the basis of the criteria of path selection, the chosen paths may be alternately or simultaneously move to the network traffic [2].

Owing to the MANETs and their dynamic nature, the designing of networking and communication protocols for the network can be a challenging task. Another very important aspect in the process of communication was designing routing protocols that are used for establishing and maintaining the multi-hop routes that permit communication of data among nodes. Plenty of research has been conducted in the area and several protocols of multi-hop routing were developed. Most protocols like the Temporally Ordered Routing Protocol (TORA), the Ad Hoc on Demand Distance Vector protocol (AODV), and the Dynamic Source Routing protocol (DSR) were established for maintaining routes. This was quite sufficient for a particular class of the MANET applications and is not sufficient for supporting the demanding applications like a multimedia video or audio. These applications need the provision of guarantees on a QoS [3].

There has been plenty of research that is directed in the provision of the QoS found in ad hoc networks. There is a major role to play in the routing protocol of the QoS where there is a major part found in the QoS mechanism in the ad hoc network. The QoS routing has a major role in this QoS mechanism. In spite of the nodes in the MANET like personal digital assistants and laptops which are normally limited in terms of resources, the basic challenge was in designing the routing protocol development where certain key elements like conservation of energy, load balancing and robustness are considered. Recently, there are several routing protocols in the QoS that have some distinguishing features which are proposed. Most protocols of QoS routing are found to be extensions of the currently existing routing protocols and are thus classified into two categories: the on-demand (reactive and the table-driven (proactive). For an on-demand routing protocol, all routes were discovered between the source and its destination. This gives a reduced overhead and is found to be quite scalable and this is more common in a high-density network that is large. At the same time,

for a routing protocol that is table-driven, the nodes will have to maintain all routing information within the network and periodically update it for purposes of communication. For the purpose of this approach, all nodes will have to maintain this as the latency of path finding is quite small and the overhead is quite high, as a path that has not been used for a very long time will continue to be updated [4].

The actual purpose behind the routing protocols that are QoS aware was the calculation of the parameters of the QoS and for choosing a path from the source to the destination which can satisfy the QoS applications and their requirements. The selection or optimization of routing protocols has a major role to play in improving the ad hoc network QoS. There are some metaheuristic approaches that have provided some suitable solutions to this [5]. One such very promising option which could be applied to the MANETs was the SDS and the SA. For the purpose of this work, there was a hybrid SA-SDS algorithm that had been proposed. The rest of the investigation has been organized thus. Section 2 explains all related work found in the literature. The different methods used have been discussed in Section 3. The experimental results have been discussed in Section 4 and Section 5 has concluded the work.

2 RELATED WORKS

Kalaiselvi and Radhakrishnan [6] had made a presentation of a multi-constrained QoS Routing (QoSR) that was based on Differentially Guided Krill Herd (DGKH) based algorithms in the MANETs. The QoSR was found to be a very significant problem which is NP-complete having many challenges in the determination of optimum paths which can simultaneously be able to satisfy different MANET constraints even in changing topology. There are several other heuristic algorithms used widely for solving such issues that have to compromise on different complexities of computation or low performance. The work further proposed a new algorithm based on krill herd known as the DGKH, in which krill individuals did not update their position compared to the one-to-one aspects but make use of information from different krill individuals and then looked out to determine a path which was feasible.

For the purpose of enhancing QoS communication for the MANETs, there was an exponential Genetic Algorithm (GA)-based Stable and Load-Aware QoS Routing protocol (SLAQR) that had been proposed by Rao et al.,. The primary focus of this work was on enhancing the routing algorithm that was GA-based by means of including an exponential function incorporating the metrics of the QoS such as the static resource capacity of the node, its dynamic resource availability, link quality and quality of the neighbourhood. This protocol's originality is from the very fact that there were multiple parameters that had been introduced within the computation of route quality and this also integrates an exponential function within the GA.

Kout et al., [7] had proposed another novel protocol in routing that had its inspiration from the Cuckoo Search (CS) method which implemented the NS-2 protocol for routing. It also selected the Random WayPoint model as its model of mobility. For validation of this work, there was a comparison made to the routing protocol known as AODV and the

Destination Sequence Distance Vector (DSDV) along with a new bio-inspired protocol known as AntHocNet which was in connection to the parameters of the QoS: the end-to-end delay and the packet delivery ratio.

Prabaharan and Ponnusamy had made use of a three-opt variant of the ACO that had a very high emphasis to exploration. This three-opt variant had been operating by means of the identification of its shortest path once each ant that was part of the system was able to identify a complete path. The SA was used for identifying an ideal path from these sets. The primary concentration here was to get a point of equilibrium for ideal time and efficiency. There were many experiments that were conducted by making use of many other datasets and both randomness and efficiency were observed in the paths.

Newton and Mohideen [8] had brought about a proposal for a mechanism of an Improved Ant Colony Optimization that included Least Interference Routing (LIRANT). The technique was implemented by means of the ACO with the least interference technique of routing. The results of simulation proved that once it was incorporated, it improved the least interference routing along with the ACO where the MANET performs better with increased throughput.

Jayavenkatesan and Mariappan [9] had made a new proposal of another hybrid optimization technique which combined the Ant Colony Optimization (ACO) along with the Fitness Distance Ratio Particle Swarm Optimization (FDR PSO) (ACO-FDR PSO) for optimizing throughput. The ACO will identify its unique path within the network which is dependent on the higher concentration of pheromone. The FDR PSO will bring down the throughput that is based on the Packet Delivery Factor (PDF). As the FDR PSO uses the N-best particle in the equation of velocity updating, the main drawback of the PSO was premature convergence and being stuck in the local optima. By means of the results it was proved that this technique had better results compared to the ACO, the PSO, the ACO-PSO or the AODV techniques.

The Multicast routing has proven to be yet another effective scheme in routing. Singh et al., [10] proposed the Genetic oriented QoS Multicast Routing (GA-QMR) algorithm. This was an algorithm which was efficient in finding an optimal multicast tree that can satisfy various parameters of the QoS compared to the other two algorithms. It has the ability to be able to explore several paths at the same time and choose the best one on the basis of the parameters of quality. These routes will be needed for satisfying any end-to-end delay, bandwidth, packet loss rate, jitter or packet success rate.

In recent times, there are several metaheuristic algorithms that formulate the problem of multicast routing which is taken as a problem of a single objective. Wei et al., [11] had been considering Multi-Objective Multicast Routing- Differential Evolution algorithm (MOMR-DE), its bandwidth, jitter, delay, cost and lifetime of the network as the five objectives. Additionally, the authors also modified the mutation and crossover operators for building the path that was the shortest to the multicast tree for maximizing the lifetime of the network, jitter, delay, bandwidth and minimizing cost. The results of the simulation proved

that the method was capable of getting a faster range of convergence and is preferred for MANET multicast routing.

3 METHODOLOGY

The Ad Hoc Multipath Distance Vector (AOMDV) was another loop free extension of that of the AODV demanding the least link disjoint of multiple paths. For supporting this multipath routing, there are route tables that contain three paths for every destination [12]. The paths that are given to one destination have been assigned a similar destination sequence number. As soon as a route request having a higher number is got, all the other routes having older sequence numbers will be removed. The attributes of the route table will be the previous hop and its hop count. There are different predecessors that ensure the disjointness of the links. There is a QoS-based routing metric used for the MANETs that need to incorporate a minimum bandwidth available along with an end-to-end latency with congestion that is around a link. The congestion is related to the quality of the channel and this is dependent on its Medium Access Control (MAC) based access contention and the reliability of a channel. For the purpose of this section, the SDS, the SA-CDS, the SA, the CDS and hybrid SA-SDS methods have been discussed.

3.1 Connected Dominating Set (CDS)

In the case of all wireless ad hoc networks, there has been no pre-defined or fixed infrastructure. The nodes found in the wireless network will communicate through a shared medium which is either a single or multi-hop. Even though there is no physical backbone infrastructure, there is a virtual backbone that is formed by means of construction of the CDS. When given in an undirected graph $G = (V; E)$, the subset $V' \subseteq V$ will be a CDS of that of G for every node $u \in V$, u will be either in V' or there may be another node $v \in V'$ so that $uv \in E$ and here the sub-graph that is induced by V' , i.e., $G(V')$, will be connected. All nodes found in the CDS will be known as the dominators and the others the dominates. Using the CDS, routing will become easier and can adapt itself to changes in topology. For the purpose of reducing traffic at the time of communication and for simplification of management of connectivity, it may be ideal to construct another Minimum CDS (MCDS) [13].

The problem of the CDS was duly investigated within the Unit Disk Graph (UDG), wherein every node will be found inside the same range of transmission. The problem of the MCDS problem in the UDG is NP-hard. For building the CDS, all current algorithms will have to identify a Maximal Independent Set (MIS) I of G , after which it has to connect all the nodes in I to get a CDS. Independent set I will be a subset of that of V so that for every two nodes $u, v \in I$, $uv \notin E$. This further means the nodes in i will be nonadjacent pairwise. There is also a maximal independent set made to ensure no more nodes are added to the non-adjacency property.

3.2 Simulated Annealing (SA) Algorithm

The SA is an ideal example of the modern metaheuristic algorithms which were developed by Vecchi, Gelatt and Kirkpatrick in the year 1983. This was inspired by the metals and their annealing process when treated in heat and the Metropolis algorithms for the simulations of Monte Carlo. The primary ideal of this SA algorithm was very similar to the dropping of certain bouncing balls in a landscape. When these balls bounce for some time and lose their energy they settle down in the local minima. In case the balls are permitted to bounce enough number of times, they may lose energy slowly but will fall into their lowest locations eventually thus reaching the minimum. But it may use several other balls (that are parallel to annealing) or make use of one single ball for tracing its due trajectory (standard SA) [14].

The process of optimization will ideally start from one initial guess that has a higher energy level. It will then move to the other locations in a random manner with slightly lower energy. This move will be accepted in case the new state has even lower energy and the solution will then improve since it has a better objective or a value that is lower for minimization and its objective function. But, in case there is no improvement to the solution, it may still be accepted with the below mentioned probability as per (1):

$$p = \exp\left[-\frac{\delta E}{kT}\right] \quad (1)$$

This is a probability distribution of a Boltzmann-type where T denotes the system's temperature and k denotes the Boltzmann constant that is taken to be 1 for purposes of simplicity. The difference of energy δE will often be connected to its objective function which is $f(x)$ for being optimized. A trajectory found in the Simulated Annealing is a path which is piecewise and is the Markov chain in the new state (or the new solution) that is dependent on its present state or solution with a transition probability p.

For this, the diversification is through the randomization that produced some more new solutions (or locations) and whether this new solution has been accepted will only be determined by means of a probability. In case T is found to be too low ($T \rightarrow 0$), any other $\delta E > 0$ (the worse solution) gets accepted rarely as the $p \rightarrow 0$, where the diversity of these solutions will be limited subsequently. At the same time, if T is very high, the system will also be in a state of high energy and more new changes will get accepted. Thus the temperature T will normally be the control of the balance of both intensification and diversification. For changing T, the cooling schedule is adopted [15].

Identified are two different categories of these cooling scheduled: for the purpose of monotonic cooling, the geometric cooling schedule is widely used: $T(t) = T_0 \alpha^t$, in which t denotes its time step or the iteration counter and the T_0 denotes its initial temperature. For this, there is no need for a determination of its final temperature and the α will be within a range of (0, 1). The primary disadvantage was in the case of small values that correspond to Simulated Quenching (SQ), where there is a risk to the system to be able to freeze very fast and can also get trapped in local optima. In case α is found to be approaching 1, its

convergence will be slow. A Monotonic cooling schedule can be applied to ensure the system is elevated to a higher energy state when needed. In case the temperature is raised many times, it can adversely affect the convergence. This further demonstrates the fact that there exists an ideal balance between that of its diversification and its intensification.

3.3 CDS-SA Algorithm

As according to the standard SA, the SA-MDS will work by choosing trial solutions within the neighbourhood for the particular trial solution. There is an evaluation of the quality of both these solutions by making use of the objective function that has been defined as per equation (2), and for the subsequent iterate solution, the choice is made from two different solutions.

$$f(x) = \frac{n}{|V|} + \frac{1}{|V| \gamma_x(G)} \quad (2)$$

In case n denotes the node number that is covered using solution x , and $\gamma_x(G)$ denotes the actual number of nodes that are contained in x . The objective function normally contains two parts the first one being $n/|V|$, which reflects the actual size of the domination on the G by x . In case x represents the dominating set, this part will be equal to 1. At the same time, the second part $\frac{1}{|V| \gamma_x(G)}$ will distinguish between various solutions with similar values of that of the first part that is based on the actual number of nodes in every one of them [16].

The trial solutions that are better will generally be accepted but the ones that are worse may only be accepted with a probability $p = \exp\left(\frac{\Delta f}{T}\right)$; wherein the Δf denotes the actual amount of bringing down the objective value which is caused by a downhill move (i.e., $\Delta f \leq 0$), and the T denotes a parameter that is referred to as an “annealing temperature”. The primary idea behind this was to find it better to be able to accept a penalty of a short-term hoping to find some significant rewards for a longer term. This means, a search may be able to accept a solution that is inferior in order to escape from solutions that are locally optimal. There is also a control parameter (which is generally the temperature T which is owing to an analogy it has with the process of physical annealing) which is used for controlling any acceptance of trial solutions that are inferior. In the beginning of the search, its temperature $T_{\max}$ will be high thus permitting an unrestricted movement within the search space. Temperature $T_{\max}$ will gradually be reduced while the search takes place and this constrains the acceptance of any trial solutions that are inferior. There are some steps to the SA-MDS method which shows the manner in which the SA acceptance and cooling schedule are applied with the $T_{\min}$ as the temperature at its lower limit.

A formal description for the SA-MDS is as per the below-mentioned algorithm.

1. *Initialization. Choose the cooling schedule parameters :*

initial temperature $T_{\max}$, final temperature $T_{\min}$,

cooling ratio $\lambda \in (0, 1)$, and the epoch length M .

Set $T := T_{\max}$, and generate an initial solution x^0 .

Set $x^{best} := x^0$, and $k := 0$.

2. *Evaluation. Evaluate the objective function at x^k by using eq.2.*

3. *Main Loop. Repeat the following Steps 3.1–3.4 M times.*

3.1. *Set $y = x^k$.*

3.2. *Node-Reduction. If $f(x^k) \geq 1$, then go to Step*

3.2.1. Otherwise, go to Step 3.3.

3.2.1. *Randomly select a component y_i with value 1 as in Step 3 of the Procedure 1, set $y_i = 0$.*

3.2.2. *If $\Delta f(x) = f(y) - f(x^k) > 0$, set $x^{k+1} := y$, update x^{best} , set $k := k + 1$, and go to step 4.*

3.2.3. *The trial solution y is accepted with the probability*

$$p = \exp\left(\frac{\Delta f}{f}\right), \quad (3)$$

where $\Delta f(x) = f(y) - f(x^k) \leq 0$.

3.2.4. *If y is accepted, then set $x^{k+1} := y$. Otherwise, set $x^{k+1} := x^k$.*

3.2.5. *Set $k := k + 1$, and go to Step 4.*

3.3. *Node-Addition.*

3.3.1. *Randomly select a component y_i with value 0 as in Step 4 of the Procedure 1, set $y_i = 1$.*

3.3.2. *If $f(y) > f(x^k)$, set $x^{k+1} := y$, update x^{best} , set $k := k + 1$, and go to Step 4.*

3.4. *Node Swapping. If $f(y) < f(x^k)$, then set $z = x^k$.*

3.4.1. *Randomly replace a component z_i with value 1 with a component z_j with value 0 as in Step 5 of the Procedure 1, i.e., $z_i = 0, z_j = 1$.*

3.4.2. *If $f(y) > f(x^k)$, set $x^{k+1} := z$, update x^{best} , set $k := k + 1$, and go to Step 4.*

3.4.3. *Accept z with the probability similar to p which is given by Eq. 3.*

3.4.4. *If z is accepted, then set $x^{k+1} := z$. Otherwise, set $x^{k+1} := x^k$. Set $k := k + 1$, in both cases.*

4. *Epoch Length Condition. If the epoch length M is attained, then go to Step 5. Otherwise, go to Step 3.*

5. *Stopping Condition. If $T > T_{\min}$, then set $T := \lambda T$, and go to Step 3. Otherwise, terminate.*

The SA-MDS tends to follow the SA and its main framework. So, there is an initial solution which is randomly chosen within its search space which is x^0 . For every iteration k , there is a trial solution y that is generated within the neighbourhood for the present iterate solution which is x^k . This process of generation is completed by means of the cases below [17].

– In case x^k denotes the dominating set (i.e. $f(x^k) \geq 1$), then the y will be generated in a way which brings down the cardinality of the x^k as per Step 3 of Procedure 1. This process is known as the “Node-Reduction”.

– In case the x^k fails to represent the dominating set (i.e. $f(x^k) < 1$), y will be generated in a way to increase the nodes covered by the x^k as per Step 4 of Procedure 1. This process of generation is known as the “Node-Addition”.

– In case this process of Node-Addition is not able to bring about an improvement to x^k , there is one more process known as “Node-Swapping” that has been applied as per Step 5 of the Procedure 1.

For all of these processes of generation, there is an annealing acceptance mechanism that has been applied for either accepting or rejecting the trial solution y which is generated.

3.4 Stochastic Diffusion Search (SDS) Algorithm

An SDS is that multi-agent global search algorithm for optimization based on a simple agent interaction. There is also a high level description of the SDS that has been presented as a social metaphor that demonstrates the process by means of which the SDS will allocate its resources. It acts as a population based optimisation and global search algorithm which is a distributed mode to compute the utilization of any interaction among simple agents [18].

This SDS algorithm begins either a search or an optimization by means of being able to initialize population (such as miners in the metaphor of mining games). For any type of SDS search, every agent will maintain a new hypothesis which is h , that defines any possible solution to a problem. In the analogy of mining game, a hill is identified by the agent hypothesis. Once initialization is complete two phases follow:

- The Test Phase (such as testing the availability of gold)
- The Diffusion Phase (such as congregation or information exchange)

SDS algorithm is as below:

Initialising agents ()
While (stopping condition is not met)
Testing hypotheses ()
Diffusion hypotheses ()
End

While in a test phase, the SDS will check if the agent hypothesis is successful by means of a partial hypothesis evaluation that gives a Boolean value. After this, being contingent on the employed strategy for recruitment, there can be some successful hypotheses that diffuse across the population and there is information on solutions which are potentially good which get spread throughout the whole agent population. For the Test phase, an agent will perform the partial function evaluation, pFE, that in a certain function of the hypothesis of the agent is employed; $pFE = f(h)$. For the mining game, there is a partial function evaluation which entails any randomly chosen region that has been defined by the hypothesis of the agent (as opposed to mining the regions on the same hill). In the case of the diffusion phase, every agent will recruit for the interaction and also the communication of the hypothesis. Within the metaphor of the mining game, there is diffusion performed by means of communicating the hill hypothesis.

3.5 Proposed Hybrid SA-SDS Algorithm

Unlike in the case of several algorithms inspired by nature, the SDS has a stronger mathematical framework describing the algorithm and its behaviour by means of an investigation of the allocation of resources, its convergence to a global optimum, its linear time complexity, minimal convergence criteria and its robustness. The primary disadvantage in the case of the SDS was for the search spaces that have been heavily distorted by noise and the activity's diffusion owing to the disturbances that may bring down the average number of its inactive agents that are part of a random search and this will in effect increase time required for reaching their steady state.

An SA algorithm has been employed for the purpose of avoiding getting trapped within the local minimum and also for increasing the diversity of the agents. A hybrid SA-SDS algorithm has as its parameters, an $N \times N$ integer matrix (along with $N = \#edges$) containing edges between an edge i and j . The $P \times N$ integer matrix known as an agent will then be implemented. Its P value will correspond to the actual number of agents that are used in this algorithm. Every row in this matrix will correspond to a vision of an agent on every city and also its neighbours [19]. Here, every row will denote a feasible solution used for a Travelling Salesman Problem (TSP). Every row here is taken to be a cycle vector. Step 1 of a hybrid algorithm will be the initialization of agents. For this purpose, every agent (a row within the agent matrix) will be initialized by making use of a random value for every such position. One unique constraint here was assigning a value which is in $[0, N-1]$ interval and this is not repeated within the row again.

After this, the SA will be introduced for obtaining another good solution from each such agent. For the purpose of this, an energy function E had been considered so that $f(S_i)$

indicates the actual length of the tour within agent S_i . Then the $\Delta f = f(S_{new}) - f(S_{old}) = \Delta E$, has been defined and this will correspond to the gap in energy between an agent (S_{old}) within the swarm and also its neighbour (S_{new}). The neighbour will then be generated by making use of a simple and random swap move that happened between both these edges.

Either the acceptance or the rejection of that of the neighbour and the update of an agent tour which is dependent on the below-mentioned expression that corresponds to the equation and its implementation (4):

$$P = \begin{cases} \exp\left[\frac{-\Delta f}{t_k}\right] & \text{if } \Delta f > 0 \\ 1 & \text{if } \Delta f \leq 0 \end{cases} \quad (4)$$

As soon as every agent is able to find the best neighbour tour $S_i^{bestNeighbor}$ (by employing the SA strategy) the agent that is best will be chosen from among its agent matrix, $S_{global}^{bestNeighbor}$. After this, every agent S_i , found within the agent matrix, will be updated by means of applying the crossing operation on both $S_i^{bestNeighbor}$ and the $S_{global}^{bestNeighbor}$. Lastly, temperature T is computed with the equation below (5):

$$T = \alpha^i T_0 + T_\lambda \quad (5)$$

In which the cooling coefficient α was a random constant in range $[0, 1]$, i denotes the iteration number and T_0 the initial temperature, T_λ denotes the end criterion limit value.

As discussed above, the primary difference between an algorithm that was proposed by Fang was the frequency found in using the SA algorithm for increasing the level of diversification. In the SA, it made use of only the first agent search. On the other side, a hybrid SA-SDS algorithm makes use of an SA algorithm along the iterations of the SDS algorithms. Furthermore, for the purpose of this algorithm there was a simple sort strategy that had been used for edges in which the S_i agent will be updated. For the edges that had been considered to be in an order that is the same as in an array (agent). The Bubble sort algorithm had been used to order edges from agent S_i : firstly, all these edges that are nearest to the chosen edge and lastly those edges that are the farthest the chosen edge. Improvements that are important here have been related to the difference and they are the time taken for execution and the solution's quality which is obtained by a hybrid SA-SDS algorithm.

4 RESULTS AND DISCUSSION

In this section, the CDS-SA and hybrid SA-SDS methods are used. Experiments are carried out using 100 to 500 number of nodes. The average packet loss, average end to end delay, jitter and control packet overhead as shown in tables 1 to 4 and figures 1 to 4.

Table 1 Average Packet Loss (%) for Hybrid SA-SDS

Number of nodes	CDS-SA	Hybrid SA-SDS
100	11.2	11
200	12.4	11.6
300	15.5	14.1
400	15.2	15.2
500	17.9	17

Figure 1 Average Packet Loss (%) for Hybrid SA-SDS

From the figure 1, it can be observed that the hybrid SA-SDS has lower average packet loss by 1.8% for 100 number of nodes, by 6.67% for 200 number of nodes, by 9.46% for 300 number of nodes, by same value for 400 number of nodes and by 5.16% for 500 number of nodes when compared with CDS-SA respectively.

Table 2 Average End to End Delay (Second) for Hybrid SA-SDS

Number of nodes	CDS-SA	Hybrid SA-SDS
100	0.02032	0.02018
200	0.02578	0.02478
300	0.03277	0.03095
400	0.0355	0.03334
500	0.0402	0.0378

Figure 2 Average End to End Delay (Second) for Hybrid SA-SDS

From the figure 2, it can be observed that the hybrid SA-SDS has lower average end to end delay by 0.69% for 100 number of nodes, by 3.95% for 200 number of nodes, by 5.71% for 300 number of nodes, by 6.27% for 400 number of nodes and by 6.15% for 500 number of nodes when compared with CDS-SA respectively.

Table 3 Jitter (Millisecond) for Hybrid SA-SDS

Number of nodes	CDS-SA	Hybrid SA-SDS
100	0.00013	0.00009
200	0.00009	0.00017
300	0.00001	0.00028
400	0.00022	0.00028
500	0.00007	0.00003

Figure 3 Jitter (Millisecond) for Hybrid SA-SDS

From the figure 3, it can be observed that the hybrid SA-SDS has higher jitter by 61.54% for 200 number of nodes, by 186.21% for 300 number of nodes, by 24% for 400 number of nodes, but lower jitter by 36.36% for 100 number of nodes and by 80% for 500 number of nodes when compared with CDS-SA respectively.

Table 4 Control Packet Overhead for Hybrid SA-SDS

Number of nodes	CDS-SA	Hybrid SA-SDS
100	4.68	5.45
200	4.87	6.96
300	8.09	5.65
400	7.28	6.9
500	10.07	6.59

Figure 4 Control Packet Overhead for Hybrid SA-SDS

From the figure 4, it can be observed that the hybrid SA-SDS has higher control packet overhead by 15.2% for 100 number of nodes, by 35.33% for 200 number of nodes, but lower control packet overhead by 35.52% for 300 number of nodes, by 5.36% for 400 number of nodes and by 41.78% for 500 number of nodes when compared with CDS-SA respectively.

5 CONCLUSION

Another key area that is related to research in the MANETs is Load balancing. The QoS provides certain guaranteed services like the rate of packet delivery to users, delay jitter, delay and bandwidth. When supporting more of these QoS constraints the QoS problem is NP-complete. Normally, the multipath routing algorithms tend to face a new challenge as to how the traffic volume can be distributed to a certain path. On the basis of the feedback of measurements from real-time traffic, it had been designed in a very efficient and simple metaheuristic algorithm. The SA is a metaheuristic method which is well-known and this is

applied in a successful manner to the problems in combinatorial optimization. The SDS is taken to be a global search based on a multi-agent population and this is a distributed mode to compute the interaction which exists between some simple agents. In recent times, the SA-SDS hybrid algorithm had been employed to be used in order to solve a multipath routing problem and a classical TSP. The primary difference that exists between algorithms was the frequency with which the SA algorithm was used to increase the level of diversification: In the case of the SA this is used only in a first agent search. On the other side, SA-SDS algorithm makes use of the SA algorithm with the iterations of the SDS algorithm. Furthermore, considering a simple sort algorithm that is used it can perform the agent movement more efficiently. The results proved that a hybrid SA-SDS had a higher jitter by about 61.54% for the 200 number of nodes, by about 186.21% for the 300 number of nodes, by about 24% for the 400 number of nodes, but a lower jitter by about 36.36% for the 100 number of nodes and finally by about 80% for the 500 number of nodes on being compared to the CDS-SA.

Funding Not applicable.

Availability of Data and Material Not applicable.

Compliance with Ethical Standards

Conflict of interest We authors not having any conflict of interest among ourselves to submit and publish our articles in Wireless Personal Communications journal.

Code Availability Not Applicable.

REFERENCES

1. Mueller, S., Tsang, R. P., & Ghosal, D. (2004). Multipath routing in mobile ad hoc networks: Issues and challenges. In *Performance tools and applications to networked systems* (pp. 209-234). Springer, Berlin, Heidelberg.
2. Manohari, P. K., & Ray, N. (2016, February). Multipath routing protocols in MANETs: A study. In *Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, 2016 International Conference on (pp. 91-96). IEEE.
3. Jawhar, I., & Wu, J. (2005). Quality of service routing in mobile ad hoc networks. In *Resource Management in Wireless Networking* (pp. 365-400). Springer, Boston, MA.
4. Gulati, M. K., & Kumar, K. (2012). QoS routing protocols for mobile ad hoc networks: a survey. *International journal of wireless and mobile computing*, 5(2), 107-118.
5. Al-Ani, A. D., & Seitz, J. (2016). QoS-aware routing in multi-rate ad hoc networks based on ant colony optimization. *Network Protocols and Algorithms*, 7(4), 1-25.
6. Kalaiselvi, D., & Radhakrishnan, R. (2015). Multiconstrained QoS routing using a differentially guided krill herd algorithm in mobile ad hoc networks. *Mathematical Problems in Engineering*, 2015.
7. Kout, A., Labed, S., & Chikhi, S. (2018). AODVCS, a new bio-inspired routing protocol based on cuckoo search algorithm for mobile ad hoc networks. *Wireless Networks*, 24(7), 2509-2519.
8. Newton, P. C., & Mohideen, M. S. K. (2018). LIRANT: An Improved Ant Colony Optimization mechanism with Least Interference Routing for Enhancing Throughput of MANET. *International Journal of Computer Sciences and Engineering Open Access*, Vol-6, Special Issue-2, 38-42.
9. Jayavenkatesan, R., & Mariappan, A. (2018). A Hybrid Optimization for Improving Network Performance in Scalable Mobile Ad Hoc Network. *Indian Journal of Science and Technology*, 10(41).
10. Singh, S., Koslia, M., & Poonia, R. C. (2018). A GA-QMR: Genetic Algorithm Oriented MANET QoS Multicast Routing. *Recent Patents on Computer Science*, 11(4), 268-275.
11. Wei, W., Qin, Y., & Cai, Z. (2018). A multi-objective multicast routing optimization based on differential evolution in MANET. *International Journal of Intelligent Computing and Cybernetics*, 11(1), 121-140.
12. Tarique, M., Tepe, K. E., Adibi, S., & Erfani, S. (2009). Survey of multipath routing protocols for mobile ad hoc networks. *Journal of network and computer applications*, 32(6), 1125-1143.
13. Thai, M. T., Wang, F., Liu, D., Zhu, S., & Du, D. Z. (2007). Connected dominating sets in wireless networks with different transmission ranges. *IEEE transactions on mobile computing*, 6(7), 721-730.
14. Yang, X. S. (2009). Harmony search as a metaheuristic algorithm. In *Music-inspired harmony search algorithm* (pp. 1-14). Springer, Berlin, Heidelberg.

15. Román-Román, P., & Torres-Ruiz, F. (2015). A stochastic model related to the Richards-type growth curve. Estimation by means of simulated annealing and variable neighborhood search. *Applied Mathematics and Computation*, 266, 579-598.
16. Hedar, A. R., Ismail, R., El Sayed, G. A., & Khayyat, K. M. J. (2015, July). Two meta-heuristics for the minimum connected dominating set problem with an application in wireless networks. In *Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence (ACIT-CSI), 2015 3rd International Conference on* (pp. 355-362). IEEE.
17. Hedar, A. R., & Ismail, R. (2012). Simulated annealing with stochastic local search for minimum dominating set problem. *International Journal of Machine Learning and Cybernetics*, 3(2), 97-109.
18. Al-Rifaie, M. M., Bishop, M. J., & Blackwell, T. (2011, July). An investigation into the merger of stochastic diffusion search and particle swarm optimisation. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation* (pp. 37-44). ACM.
19. Cabrera, G., Roncagliolo, S., Riquelme, J. P., Cubillos, C., & Soto, R. (2012). A hybrid particle swarm optimization-simulated annealing algorithm for the probabilistic travelling salesman problem. *Studies in Informatics and Control*, 21(1), 49-58.

Figures

Figure 1

Average Packet Loss (%) for Hybrid SA-SDS

Figure 2

Average End to End Delay (Second) for Hybrid SA-SDS

Figure 3

Jitter (Millisecond) for Hybrid SA-SDS

Figure 4

Control Packet Overhead for Hybrid SA-SDS