Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 6641857, 17 pages
https://doi.org/10.1155/2021/6641857

*Research Article*

# An Improved Harmony Search Algorithm for Proactive Routing Protocol in VANET

**Ravie Chandren Muniyandi,[1] Mohammad Kamrul Hasan,[1] Mustafa Raad Hammoodi,[1] and Ali Maroosi[2]**

[1]*Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), 43600 Bangi, Selangor, Malaysia*
[2]*Department of Computer Engineering, University of Torbat Heydarieh, Torbat Heydarieh, Iran*

Correspondence should be addressed to Mohammad Kamrul Hasan; hasankamrul@ieee.org

Received 27 October 2020; Revised 28 December 2020; Accepted 4 February 2021; Published 28 February 2021

Academic Editor: Muhammad Arif



Vehicular ad-hoc network (VANET) is the direct application of mobile ad-hoc network (MANET) in which the nodes represent vehicles moving in a city or highway scenario. The deployment of VANET relies on routing protocols to transmit the information between the nodes. Different routing protocols that have been designed for MANET were proposed to be applied in VANET. However, the real-time implementation is still facing challenges to fulfill the quality of service (QoS) of VANET. Therefore, this study mainly focuses on the well-known MANET proactive optimized link state routing (OLSR) protocol. The OLSR in VANET gives a moderate performance; this is due to its necessity of maintaining an updated routing table for all possible routes. The performance of OLSR is highly dependent on its parameter. Thus, finding optimal parameter configurations that best fit VANET features and improve its quality of services is essential before its deployment. The harmony search (HS) is an emerging metaheuristic optimization algorithm with features of simplicity and exploration efficiency. Therefore, this paper aims to propose an improved harmony search optimization (EHSO) algorithm that considers the configuration of the OLSR parameters by coupling two stages, a procedure for optimization carried out by the EHSO algorithm based on embedding two popular selection methods in its memory, namely, roulette wheel selection and tournament selection. The experimental analysis shows that the proposed approach has achieved the QoS requirement, compared to the existing algorithms.

## 1. Introduction

Vehicular ad hoc network (VANET) is considered as an emerged class of the well-known mobile ad hoc network (MANET) [1, 2]. However, VANET and MANET can be distinguished with their features given in Table 1. The idea of vehicles communicating between them and forming a network to support different applications has attracted many researchers in both academic and manufacturer communities. Nodes in VANET are vehicles, and they communicate between each other using vehicle to vehicle (V2V) or vehicle to infrastructure (V2I) communication, where the first one enables vehicles to communicate with each other to share their information and the second one enables the vehicles to communicate with Road Side Units (RSU) that gather and broadcast information. These communications have led to the emergence of intelligent transportation systems (ITS), which can be used to transmit different kinds of information, thereby making driving more efficient and safe. For instance, in the case of an accident, it is useful to broadcast a message indicating the accident and warning drivers to slow down or take another route before reaching the accident area; for this reason, it's an interesting field of study for the researchers to work on and develop these kind of applications, which can save humans lives [3–5]. Figure 1 demonstrates A VANET network.

The VANET is a multi-hop network where the nodes can communicate with each other even when the sender node and the destination node are not in the same radio range. Its highly dynamic topology characterizes it, and network

Table 1: Summary of VANET and MANET features.

| Property | VANET | MANET |
|---|---|---|
| Node's mobility | High, nonrandom | Random |
| Network size | Large | Medium |
| Energy limitations | Low | Very high |
| Node's computation power | High | Low |
| Location dependency | Very high | Low |

fragmentation frequently happens as the vehicles can move at very high speeds. Thus, it's challenging to design a routing protocol that delivers the packets through the network with such unique features, especially with the absence of a central entity and the road's obstacles [6, 7]. Many routing protocols that were designed for MANET were proposed to be applied in VANET. According to their working nature, these routing protocols can be classified into proactive protocols, reactive protocols, and hybrid protocols. Proactive protocols are table-driven that need to update the routing tables and maintain network topology by sending periodic control messages, such as Destination-sequenced-Distance-Vector (DSDV) and Optimized Link State Routing Protocol (OLSR). Reactive protocols need to update their routing table by sending control messages whenever a new route is necessary. Dynamic Source Routing (DSR) and Location-Aided routing protocol (LAR) are examples of this type of routing protocol [8, 9]. Optimized Link State Routing protocol was mainly designed for MANET by the REF community [10, 11].

This paper mainly contributes to solving the OLSR optimization and network quality challenges by using the proposed improved Harmony Search Optimization (EHSO) algorithm. We have defined the optimal parameter values for OLSR optimization that significantly improves the network quality by maintaining the standard QoS.

The rest of the paper is organized as follows: Section 2 discusses the recent related works on the OLSR protocol and harmony search. Section 3 focuses on the proposed approach and its design consideration. Section 4 describes the simulation setup and parameter configuration of the simulation environment, data generation, and the node's mobility model. It then displays the performance metrics used to evaluate the results. Section 5 presents the results analysis and discussion of the proposed work. Section 6 concludes this work.

*1.1. Optimized Link State Routing Protocol.* The IETF community is credited for the design of the Optimized Link State Routing (OLSR) protocol, and it is mentioned as an experimental protocol in the Request for Comments 3626. It applies the shortest path routing algorithm, which is an extended traditional link-state approach for decreasing the overhead of link-state updates, particularly in dense ad hoc networks.

The exchange of HELLO messages amongst nodes allow them to learn their local vicinity and the link status with their neighbor (i.e. considering the link to be bidirectional or unidirectional). Through periodic topology control messages, this local information is distributed throughout the entire network. After obtaining the information through TC and HELLO messages, each node creates its view of the network topology independently and runs the Dijkstra algorithm to select the shortest routes to the potential destination. Analogous to DSDV, each protocol message is tagged by the OLSR with a sequence number to differentiate between fresh and stale information. In terms of network resources, flooding with TC messages can be a costly operation. Each node forwards a copy of the message through regular blind flooding. The OSLR uses the MPR technique to check on the cost of forwarding flooded messages, which also keeps the number of nodes required to deliver a message to a minimum, even though it still reaches the whole nonpartitioned section of the network. In the MPR approach, a node N is assumed to have knowledge of its 2-hop neighborhood. In the OLSR, this is achieved by employing neighborhood information to enrich the HELLO messages. Then, a subset of relays is selected by N amongst its 1-hop neighbors, which covers the same 2-hop nodes similar to that of the complete 1-hop neighborhood. This subset is referred to as the MPR set of N, where the MPR selector of each node in the set is defined by N. If a message has to reach the whole 2-hop neighborhood, forwarding of messages occurs for only those nodes that were selected by the source as MPRs. Figure 2 presents an example where the message reaches all the nodes through only four retransmissions, instead of the required eight transmissions in the case of blind flooding. When applying the same behavior to bigger networks, forwarding occurs for only those broadcast messages that were received through an MPR selector.

The source selects gray nodes as MPRs. Here, to reach all the nodes, just four out of eight retransmissions are required. The OLSR assumes that accessing the medium turns out to be costlier for transmitting several packets than putting more bytes in a given packet. This holds true for most wireless data link layers. Thus, the aggregation of the OLSR messages is maintained as long as possible into control packets. The trade-off between probability of reception and medium access is biased to the former, since large messages are more prone to suffering from transmission errors. Let us analyze an example of the functioning of the OLSR. The HELLO messages issued by the nodes are presented in Figure 3 in an ad hoc network. The list of neighbors is embedded in the HELLO messages regarding the sending node, and a link code that indicates the type of link is present between them: symmetric, asymmetric, or MPR.

Consequently, the basis of computing the MPR is defined by the set of HELLO mechanism to inform a neighbor if it has been selected as an MPR. In the figure, the optimal MPR is computed by $D$, which is $\{A\}$. The nonoptimal MPR set $\{A, C\}$ is sometimes selected to add redundancy.

Furthermore, it is not necessary for the optimal MPR to be unique. For instance, $C$'s optimal MPR set can either be $\{B\}$ or $\{C\}$, as the whole 2-hop neighborhood $\{A\}$ is covered by both the sets with the same number of elements. Each node can set up optimal routes (in terms of the number of hops) for all 1-hop and 2-hop neighbors by utilizing only the information obtained from the HELLO messages, even

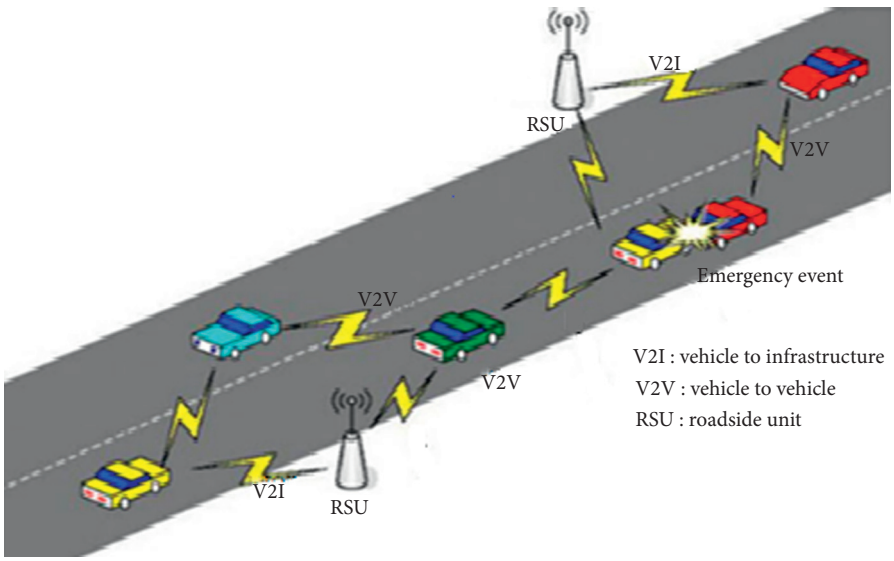


Figure 1: VANET communication architecture.

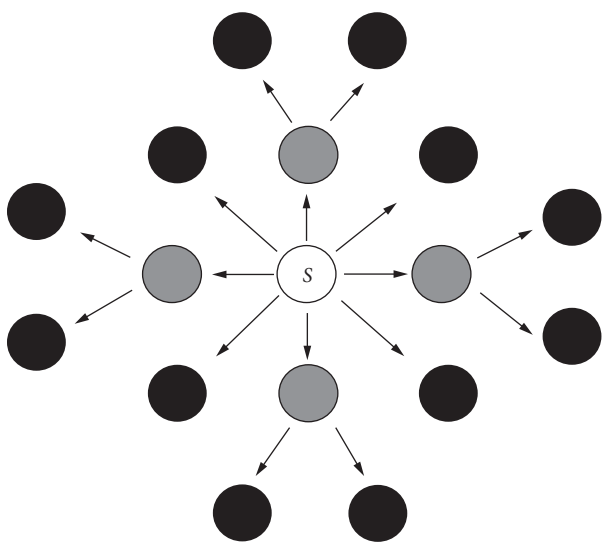


Figure 2: MPR selection and the working principle of the OLSR [8, 9, 12–14].

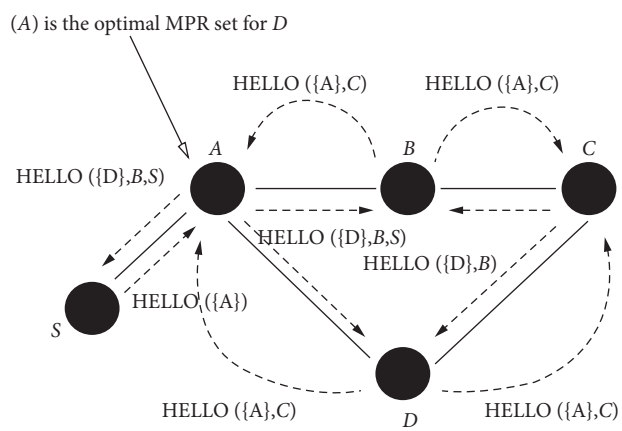


Figure 3: The working principle of the OLSR [8, 9, 12–14].

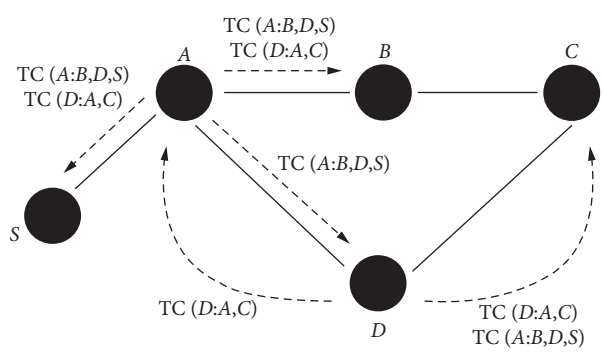


Figure 4: Continuation of the working principle of the OLSR [8, 9, 12–14].

though link states need to be exchanged via TC messages with the remaining nodes in the network in case there are more distant nodes.

Following the same example, in Figure 4, it can be seen that just $A$ and $D$ are in charge of sending TC messages and the nodes have to be selected as the MPR by at least one neighbor [8, 9, 12–14]. Besides, the MPR selectors are the only ones that forward these, i.e., messages from $D$ are forwarded by $A$ since the former selects the latter as the MPR, and vice versa. And finally, within the TC, an announcement is required for only those links with the MPR selectors. This information is enough to compute optimal routes. For instance, the HELLO exchange facilitates node B to gain information and decide that it has a direct 1-hop route to A and C.

It recognizes $S$ and $D$ to be neighbors of $A$, which helps configuring routes to them with two hops each with $A$ being the next hop. A node identifies if a neighbor is not reachable, which is achieved when either the link layer informs that it cannot deliver packets or several periodic HELLO messages were not received from the neighbor. In that case, the new topology is inserted in the HELLO and TC messages to facilitate immediate propagation. Even though the MPRs allow for achieving more significant optimization, large

networks still face flooding TC messages. In addition, in the case of highly mobile conditions experiencing frequent topology changes, the protocol overhead increases, triggered by TC messages' issue to prevent loops.

*1.2. OLSR Parameter Tuning.* The OLSR's standard configuration provides a moderate quality of services when it is applied in VANETs. Hence, after considering the effect of the parameter configuration on the whole network's performance, we have examined the optimal OLSR parameter tuning problem to establish the best protocol configuration for the deployment of VANET. Definition of the standard OLSR parameters was done without having any exact values for their ranges. Here, each parameter range of importance has been defined based on the OLSR restrictions. The aim was to avoid having pointless configurations.

Based on this, the OLSR parameters can be utilized to define a solution vector of real variables, with each one used to represent a particular OLSR parameter (see Table 2). The harmony search algorithm as an optimization technique can be used to automatically fine-tune a solution vector to obtain efficient OLSR parameter configurations that can be used for the VANETs.

The OLSR mechanisms are regulated by a set of parameters predefined in the OLSR RFC, and they are the timeouts before resending HELLO, MID, and TC messages (HELLO-INTERVAL, MID-INTERVAL, and TC-INTERVAL, respectively); the "validity time" of the information received via these three message types, which are NEIGHB HOLD TIME (HELLO), MID HOLD TIME (MID), and TOP HOLD TIME (TC); the WILLINGNESS of a node to act as an MPR (to carry and forward traffic to other nodes); and DUP HOLD TIME, which represents the time during which the MPRs record information about the forwarded packets [15].

## 2. Related Works

In the VANETs, some of the challenging features include the high node mobility, the limitation of the WiFi in the coverage and the channels' capacity, the presence of many obstacles that generate a data packet loss, topology changes, and network fragmentation [16]. Hence, Toutouh et al. [17] and Muniyandi et al. [18] proposed several optimization techniques like the Differential Evolution (DE), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and the Simulated Annealing (SA) for improving the OLSR performance in the VANETs. They did so after defining the optimization problem for tackling the optimal parameter settings. Also, they expressed a set of representative VANET scenarios (for the city of Málaga) for an accurate evaluation of the OLSR network performance using NS2. In their experiments, the authors noted that the tuned OLSR algorithm showed a better QoS as compared to the standard scheme for parameters like end to end (E2E) Delay, packet delivery ratio (PDR), and the overhead metrics [19]. However, the bio-inspired optimization algorithms have the fitness density and complexity issue that needs to be considered.

Karimi et al. [20] pointed that network lifetime was an essential factor in the design of Wireless Sensor Networks, and it depended on the sensor nodes' energy, which in turn was limited by the node's battery. Clustering is considered to be a strategy in energy management for wireless sensor networks. Consequently, Leach (Low-Energy Adaptive Clustering Hierarchy) [20] is regarded as one of the most popular clustering mechanisms. In this method, random cluster selection resulted in its inefficiency. Karimi et al. [20] proposed two algorithms—HS-leach and GP-leach. They were able to enhance the energy consumption through network partitioning. They also utilized the evolutionary algorithms to optimize the selection of the cluster heads while taking into consideration the position information and the residual energy of the WSN nodes. The simulation results obtained using MATLAB revealed that their proposed algorithms had more efficiency and increased the network's lifetime. However, the overhead, the packet loss ratio, and the complexity can be challenges for such a proposal.

Furthermore, Patil and Dhage [21] addressed the network load issue for the OLSR algorithm in the VANETs. The authors suggested using the Necessity First Algorithm (NFA) for selecting the MPR in the OLSR algorithm rather than the basic greedy algorithm after auto-tuning the different OLSR parameters. The NFA was seen to improve the routing protocols' performance and enhanced the network data rates and reduced the network load. After that, they developed an optimization technique for fine-tuning the OLSR protocol after automatically obtaining the configuration that best fits the VANET characteristics. This optimization problem was defined using the search space and quality or fitness functions. They used a simulation process for assigning the quantitative quality values (fitness) for determining the OLSR algorithm performance for the computed configurations with regard to the communication costs. The automatic selection of the best available configurations affected the fitness values. The research was carried out using the NS2. Patil and Dhage [21] observed that their proposed OLSR implemented with the novel automated selection of the optimized configuration offered better performance and further reduced the network load and was more suited for the higher density networks.

Kots and Kumar [22] proposed a heuristic approach for the effective selection of the MPRs in the OLSR routing protocol. The MPR selection was seen to be an essential function of the OLSR protocol. The researchers proposed using a novel Fuzzy logic-based routing metric for the choice of the MPRs. Their proposed process helped select the QMPRs (quality multipoint relays) for the OLSR protocol in the MANET environment. They used the fuzzy-based algorithm for predicting the quality nodes in the OLSR routing protocol in the MANET setting. For determining the node quality within the network, they considered the total sum of the quality factors like stability, energy, and buffer occupancy. They captured these metrics during the OLSR protocol's initialization, and these values were regularly updated whenever a new MPR was chosen from the nodes. This method was validated with the help of the predicted values' statistical study, while the approach was verified in

Table 2: Predefined OLSR parameters [15].

| Parameter | Standard value | Range |
|---|---|---|
| HELLO_INTERVAL | 2.0 s | $\mathbb{R} \in [1.0, 30.0]$ |
| MID_INTERVAL | 2.0 s | $\mathbb{R} \in [1.0, 30.0]$ |
| TC_INTERVAL | 5.0 s | $\mathbb{R} \in [1.0, 30.0]$ |
| WILLINGNESS | 3 | $\mathbb{R} \in [7]$ |
| NEIGHB_HOLD_TIME | 3 × HELLO_INTERVAL | $\mathbb{R} \in [3.0, 100.0]$ |
| MID_HOLD_TIME | 3 × TC_MID | $\mathbb{R} \in [3.0, 100.0]$ |
| TOP_HOLD_TIME | 3 × TC_INTERVAL | $\mathbb{R} \in [3.0, 100.0]$ |
| DUP_HOLD_TIME | 30.0 s | $\mathbb{R} \in [3.0, 100.0]$ |

MATLAB. The method ensured an improved network and lesser power consumption in a MANET setting and enhanced network efficiency. However, an experimental study is mandatory to test network efficiency and energy consumptions.

In VANET, frequent topology changes happen because of the high mobility of the nodes and the limitation of signal coverage [23]. However, the reactive routing protocol's scope in VANET is limited due to VANET's topology instability. On the other hand, proactive routing protocols like the OLSR, which are mainly designed for the MANET, also fail to satisfy the wide range of data services that have been intended for the VANET. This is due to the existing OLSR protocol's inability to predict channel overload and sense channel conditions. To obtain better routing efficiency, the network has to gain some cognitive capacity that will allow it to select an optimal path that considers both the channel information and the link-state, thereby helping it overcome the problem of channel incapacity [13]. Hasan et al. [24] attempted to improve the OLSR routing by utilizing cognitive processes that obtain and store knowledge on routing strategies to determine and select the most suitable route and channel for transmission. The focus of this research was on optimizing the OLSR routing protocol to adapt to the VANET's high mobility. First, it utilized cognitive radio to address the issue of channel inadequacy. Next, the optimal next-hop neighbor was chosen to strengthen the COG-OLSR or the link connectivity time. Based on the simulation experiments, OLSR and Cog-OLSR were compared based on performance measures such as end-to-end delay for the urban roadmap and packet delivery ratio. The former was found to have successfully estimated the projected lifetime of the link that was connected to the next hop. It was found out that the projected lifetime was close to the actual connectivity time.

Recently, it has been seen that the energy-aware and green communication protocols have garnered a lot of interest in the scientific community, specifically when using wireless mobile networks. Toutouh et al. [14] attempted to deal with decreasing the power consumption in the OLSR routing protocols used for vehicular networks. This can be accomplished by changing the standard parameter configurations and thereby reducing the routing overhead. Therefore, in this regard, meta-heuristics are a promising tool for determining the precise energy-aware OLSR configurations within a short period, despite considering a huge parameter set. Toutouh et al. [14] proposed the application of an automated configuration of the main OLSR parameters after using the parallel Evaluation Annealing (EA). The primary objectives of this study were: (i) improving the efficiency of the OLSR algorithm in the VANETs by reducing the energy consumption when standard OLSR configurations were used and (ii) scaling down the time needed for performing the automatic configurations for studying huge and realistic VANET-based scenarios. Toutouh et al. [14] studied the search space for probable combinations of 8 parameter values, which could define the OLSR routing, with GA's help.

Furthermore, the power consumed because of the data exchange for every OLSR configuration was also investigated based on the available data after carrying out the VANET simulations using the NS2. As all simulations needed a long time for their execution, Toutouh et al. [14] also parallelly used the GA to reduce the search time. Their results showed a significant improvement compared to the standard configuration with regard to the power consumed, with no consequential loss in the performance metrics.

Wahab et al. [25] had been applying the VANET QoS-OLSR protocol, which could maintain vehicular network stability while fulfilling the QoS requirements. This protocol was composed of three components: (1) QoS-based clustering with the help of the Ant Colony Optimization, (2) MPR recovery algorithm, and (3) the cheating prevention process. For ensuring cluster stability, Wahab et al. [25] also included the distance and the velocity of the vehicles, representing mobility metrics within the QoS functions. Then, the protocol selected the cluster-heads based on the local maximal QoS values. The cluster heads chose a set of optimized MPRs that could satisfy the mobility and the routing constraints as per the Ant Colony Optimization algorithm. For guaranteeing that the selection procedure was reliable and fair, Wahab et al. [25] also included a cheating prevention mechanism. Finally, they used the MPR recovery algorithm to select alternative MPRs and maintain network connections to avoid link failure. The simulation results and the performance analysis showed that the application of the newly proposed protocol could extend the life of the network by 12%, decrease the percentage of the chosen MPRs by 20%, show a 10% improvement in the PDR, and also reduce the path length by two hops.

Park et al. [26], Al-Terri et al. [27], and Prakash et al. [28, 29] addressed the issue of the MPR node disconnection because of the mobility present in the VANETs with the QoS-OLSR

protocol's help. This protocol used the MPR nodes for communicating within the clusters. The MPR disconnection is a big challenge in the VANET because of the frequent changes in the network topology. Consequently, the routing protocol performance would decline as it could significantly affect the network's connectivity. Hence, the authors proposed a novel cluster-based QoS-OLSR protocol, which was based on the Intelligent Water Drop (IWD) algorithm. This IWD-QoS-OLSR protocol was seen to improve the network connectivity and consisted of three components: the cluster formation, the MPR selection, and the MPR failure management. They carried out the cluster formation with the help of the algorithm described in the QoS-OLSR. The MPR selection was carried out using the IWD algorithm, which helped select the best path within a limited time. After that, the authors used an MPR failure management algorithm for tackling the link failure scenarios. The algorithm allowed the cluster heads to choose alternative MPR sets after the MPR was detached from the network because of their mobility. The authors used the MATLAB and Mobisim software for simulating the IWD-QoS-OLSR algorithm. However, the study conducted mainly theoretical simulation-based experiments to improve the connectivity issue. Nonetheless, in the actual experimental testing, the packet delivery ratio and the packet losses might be affected.

Mehra et al. [30] and Li et al. [31] suggested a clustering approach that was based on the routing protocol for the VANETs. Their proposed algorithm was a type of the distributed clustering algorithm along with the OLSR routing protocol and possessed an excellent data dissemination rate. The output of the protocol defined this data dissemination. This proposal could minimize the end to end (E2E) delay; however, the throughput and overhead are still a concern. Moreover, the cluster stability is another challenge for the vehicle's position (determined by the vehicle GPS), and its velocity changes with its dynamicity. Therefore, improving the mobility issue can provide ample scope to consider the clustering process between the cluster heads and the members. Moreover, the real-time experimental scenario is mandatory to measure the required QoS performance, such as E2E Delay, PDR, packet loss ratio, and throughput.

The harmony search algorithm was firstly proposed by Geem et al. [26, 32, 33]. This algorithm simulates the behavior of musicians that compose the harmony. For example, consider a group of musicians who are a part of one orchestra. The appropriate blend of organized notes that are performed by each musician is finally turned into beautiful music. Every composition of notes played by the musicians is referred to as the harmony. After production, each harmony should be esthetically checked to determine if the harmony obtained is the same. Successive exercises are performed until the harmony in consideration is produced. Musicians at any given time will practice and repeat their performance to deliver better harmonies. Consequently, each of the descriptions given about orchestra performance can be considered similar to the Harmony Search Algorithm components. Every musician in an orchestra can be used to represent a single variable in the Harmony Search Algorithm. How each musician performs notes in specific time intervals could mean how each variable is chosen from particular intervals. The harmony produced by the orchestra can be considered as one answer vector. In the same way that each harmony created by the orchestra should be checked, each solution produced by the Harmony Search Algorithm should also be evaluated based on the fitness function. Since musicians strive to improve their music quality at every practice, it means that new harmonies should be better than the previous ones. Every musician in an orchestra has three options that they can choose from for playing notes: (1) Play the notes strictly based on their memory. (2) Play notes that are somewhat similar (with minimal changes in the notes). (3) Play a random note. In the first case, a musical orchestra player can take notes from past practice sessions to improve. A variable within the harmony search algorithm can also perform a similar process using information currently stored in the harmony's memory. Each variable's position for all the rows of the matrix memory harmony remains the same. Thus, when a variable chooses the first case, one of the variables found within the memory column harmony is chosen. The selection of variable rate depends on the Harmony Memory Considerating Rate (HMCR). For the second case, a musician in a musical orchestra can implement some changes to the notes that he/she will play. This is referred to as the Pitch Adjustment Rate (PAR). Similarly, in a harmony search algorithm, a variable can slightly alter the chosen HMCR parameter's value. For the third case, a musician playing in an orchestra can also choose to play random notes. Similarly, a harmony search algorithm can also randomly select a variable's value. For this third case, the randomized term is utilized. Music improvisation refers to a process of searching for better harmonies by trying out different pitch combinations that should adhere to any of the three rules below:

(i) Playing any pitch based on memory

(ii) Playing a pitch that is adjacent to one pitch based on memory

(iii) Playing a random pitch that belongs and is a part of the possible range

This process is imitated every time the HS algorithm selects a variable. Similarly, any of the three rules presented below should be followed:

(i) Selecting any value taken from the HS memory.

(ii) Selecting an adjacent value taken from the HS memory.

(iii) Selecting a random value given the possible value range. The HS algorithm's three rules are effectively directed with the use of two vital parameters: pitch adjusting rate (PAR) and harmony memory considering rate (HMCR).

Various investigations have been conducted on the HS algorithm by modifying its structure or hybridizing it with other meta-heuristic methods to solve different optimization problems in different fields. The HS algorithm has three essential parameters, which are harmony memory

consideration rate (HMCR), pitch adjustment rate (PAR), and bandwidth (BW).

An improved version of the global-best harmony search (IGHS) algorithm was proposed by Wang and Huang [34]. The IGHS algorithm combines initialization conferring to opposition-based learning to enhance the quality of solution in the initial harmony memory, a new improvisation scheme that employs differential evolution to improve the local search capability, a modified random consideration for decreasing the randomness of the global-best harmony search (GHS) algorithm that uses the artificial bee colony algorithm, and two perturbation schemes that are used to avert premature convergence. Moreover, pitch adjusting rate and harmony memory consideration rate were two parameters of IGHS that dynamically updated according to the composite function consisting of a periodic function, a linear time-varying function, and a sign function with regard to the estimated periodicity of evolution in nature. Twenty-eight benchmark functions were tested, the results of which specify far better performance through IGHS compared with the basic harmony search (HS) algorithm, and eight popular GHS meta-heuristics were also compared with IGHS. The results indicated that IGHS was either better than or at least similar to those methods on most of the test functions.

The Hybrid Binary Differential Evolution Harmony Search Algorithm (HBDEHS) was used in Wireless Sensor Networks [35]. Wang et al. [35] focused on the Industrial Wireless Sensor Networks' (IWSNs) optimal node placement problem. As opposed to the nonindustrial Wireless Sensor Networks, IWSNs had a vital requirement on the networks' reliability. Thus, they developed a new model for node placement in IWSNs. This model was able to take into account the cost, reliability, load constraint, and scalability. Given its NP-hard quality, a new hybrid Binary Differential Evolution Harmony Search Algorithm (HBDEHS) was chosen to address the problem of optimal sensor deployment. Four large-scale node deployment problems were selected as the benchmarks and used to verify the optimization algorithm and the proposed model.

In addition, the other five binary optimization algorithms, i.e., Modified Binary Differential Evolution Algorithm (MBDE), Global Harmony Search Algorithm (NGHS), Discrete Binary PSO algorithm (DBPSO), Discrete Binary Harmony Search Algorithm (DBHS), and Simple Genetic Algorithm (SGA), were also implemented to address the problems for a comparison. Based on the experimental results, it was established that all the algorithms could determine the feasible solutions. This signified the proposed model's validity and its capacity to address the optimal node placement problem effectively. Furthermore, the comparison results also showed that HBDEHS possesses the best global search ability and that it was able to perform better than DBHS, DBPSO, MBDE, NGHS, and SGA in terms of convergence speed and search accuracy.

Amin et al. [36] came up with a modified HS with diversity injection and multi sub-harmony memories (sub-HMs) to solve the MANETs' dynamic SP problem. In the problem, the least cost solutions for bandwidth-delay-constrained systems were being sought for. Both links with minimum bandwidth and end-to-end delay were used as quality-of-service metrics to guarantee real-time applications' best performance. It exploited a set of experiments to make comparisons between the restart HS (RHS), modified HS (MHS), and standard HS (SHS). The results revealed that the MHS that was proposed performed better compared to both RHS and SHS. Furthermore, the MHS could recover from topological changes and converge rapidly to reasonable solutions before a new topological change occurs.

Moh'd Alia et al. [37] introduced a clustering algorithm based on the Harmony Search Algorithm. They developed cluster-based, energy-efficiency protocols that can be used in the Wireless Sensor Networks (WSNs). One challenging issue was determining how to organize the sensors dynamically and into a wireless communication network. They also routed the sensed information from the field sensors and sent it to a remote base station so that the lifetime of WSNs is prolonged. The authors proposed a clustering algorithm for WSNs that was energy-efficient, dynamic, and able to automatically organize the sensors into a suitable number of clusters that could be used in the network. This algorithm was based on the Harmony Search Algorithm, which was considered a meta-heuristic, music-based optimization algorithm, which got rid of the need to pre-set the number of clusters. Moreover, in the cluster head selection algorithm, a multi-objective approach was utilized to choose the best cluster headset. The simulation results revealed that the proposed algorithm was able to achieve an optimal number of clusters, prolong the lifetime of the network, and increase the delivery of data at the base station compared to other popular clustering-based routing protocols.

A differential harmony search (DHS) algorithm by blending differential evolution and harmony search processes was introduced to address the economic loading dispatch [38]. To start with, the pitch adjustment operation of the actual HS was collaborated with the differential mutation operation to improve exploitation capability. Next, the memory consideration and the improved pitch adjustment operation were deployed to boost the exploration capability. In comparison to the pure HS, the utilization of differential mutation and crossover achieved the potential to improve the exploitation in the DHS As against the pure DE; the DHS might obtain elements from as many individuals as its quantity of dimensions when creating a new individual for improving the exploration capability. Third, the researchers recommended a repair process and three straightforward selection rules to deal with the optimization problem's limitations. Simulation tests were conducted using Visual Studio 2008. The DHS outcomes were matched against a few of the current algorithms, particularly with a few other HS and DE variants. The results depicted the efficacy, competence, and sturdiness of the recommended HS when collaborating with DE to deal with large-scale optimization problems. Furthermore, the DHS was nearly robust on several parameters.

A new variant of the HS algorithm called Geometric Selective Harmony Search (GSHS) was developed by Castelli et al. [39] and Li et al. [40]. The key dissimilarities between

the actual HS and GSHS were: (i) the presence of a selection process stimulated by the tournament selection of genetic programming and genetic algorithms; (2) the definition of a new mutation operator that significantly enhances the evolvability of improved solutions due to their geometric attributes; and (3) the definition of a new memory consideration procedure, which was centered on the utilization of a recombination operator. Mainly, by construction, a geometric semantic crossover generates an offspring that was not inferior to the lowest of its parents. Geometric semantic mutation triggers a perturbation on the semantics of solutions, whose scale was regulated through a parameter. A comparison of GSHS had been made with the actual HS and another variant, the improved HS (IHS). The comparison of the algorithms had been made on 20 benchmark problems from the CEC 2010 suite. The outcomes indicated that GSHS outdoes the other approaches with statistically significant dissimilarities in nearly all cases.

Furthermore, scrutiny of the parameters initiated by the approach had been carried out. The outcomes indicated that the tournament size's small values were the most apposite. With regard to the mutation stage, it was impossible to come up with a generic inference: few of the problems perform well with a small mutation step, whereas others entail a higher value. The impact of the mutation operator was noted to be marginal when a comparison was made with the performance enhancements rising out of the other alterations.

A modified harmony search (MHS) algorithm had been presented that consists of an intersect mutation operator and local cellular search to allow addressing continuous function optimization problems [41]. During the searching process, the MHS algorithm distributes all harmonies in the harmony memory to be classified as a worse part and a better part based on their fitness, rather than just concentrating on the intelligent tuning of parameters. Harmony vectors were produced through the novel intersect mutation operation. Moreover, MHS now includes a local cellular search as well, which helps in enhancing optimization performance through the exploration of a vast search space in the early run phase to prevent premature convergence, and in the later run phase, to exploit a small region to Polish the final solutions. For the proposed MHS algorithm, an orthogonal test and a range analysis method were employed to determine the effects of parameters for achieving better parameter settings. Finally, to test and examine the proposed MHS algorithm's performance, two sets of famous benchmark functions were employed. In these benchmark sets, functions include various characteristics to give a comprehensive assessment of the performance of the MHS. Based on the experimental results, the proposed algorithm provided better performance than those state-of-the-art HS variants but compared with other famous meta-heuristic algorithms according to the solution's efficiency and accuracy.

Al-Betar et al. put forward three new versions of the harmony search algorithm [26]. The harmony search algorithm had been projecting the variation of the natural principle, the survival of the fittest, in the update process to enhance diversity. Natural Proportional Harmony Search (NPHS) was the first version, which allocated a probability for each individual in the HM [42]. There was a higher probability of the worst fitness values being replaced by the new harmony solution. Natural Tournament Harmony Search (NTHS) was the second HS version, which picked a set of individuals determined through the HM's tournament size, where, if better, the new harmony replaced the worst individual among them. Natural Rank Harmony Search (NRHS) was the third version of HS, which ranked individuals according to their fitness values to identify the best to the worst. Four different economic loading dispatch problems were employed to examine the convergence characteristics of the three HS versions. Based on the results, proposed HS variations were seen to have the potential to get fruitful results for all Electronic Logging Device problems. Indeed, applying a natural update process in HS could be a promising research direction that could result in numerous future discoveries.

Shankar et al. [43] addressed the energy efficiency challenge in wireless sensor networks by considering sensor nodes as battery-operated devices. For transmitting data in an energy-efficient manner, clustering-based techniques were utilized via data aggregation. This was performed to balance the consumption of energy among the network's sensor nodes. The current clustering techniques use Harmony Search Algorithm (HS), distinct Low-Energy Adaptive Clustering Hierarchy (LEACH), and Particle swarm optimization (PSO) algorithms. Nevertheless, individually, these algorithms faced local search and exploration–exploitation trade-off (PSO) constraints [24, 44]. To achieve a global search that has faster convergence, an HSA and PSO algorithm hybrid was proposed to conduct an energy-efficient selection of the cluster head. The proposed algorithm possesses the HS's high search efficiency and the PSO's dynamic capability. These qualities improved the sensor nodes' lifetime. The hybrid algorithm's performance was assessed based on the number of dead nodes, the number of live nodes, and throughput and residual energy. The authors evaluated the proposed approach using MATLAB. Based on the results, the hybrid HSA-PSO algorithm achieved an improvement in throughput and residual energy by 29.00% and 83.89%, respectively. This was in comparison with the PSO algorithm.

A multi-objective optimization harmony search parallel algorithm aligned with cloud computing was developed by Li et al. [31] to address traditional harmony search issues in complex function multi-objective optimization, like slow convergence, low precision, and easy to fall under local optimum. First, a single harmony library was employed to store and process memory harmony, a characteristic of the traditional harmony search algorithm, which was segmented into numerous harmony sub-libraries based on various harmonies. Simultaneously, the dynamic trade-off factor and roulette selection strategies were employed to dynamically set harmony memory library value-taking probability, fine-tuning pitch bandwidth, fine-tuning pitch probability, and other parameters that were relied upon by traditional harmony search algorithm. The MapReduce programming model was then utilized to set up the Map and Reduce core

parallel computing functions to build the parallel algorithm of dynamic parameter harmony search centered on cloud computing [45]. Finally, a Hadoop platform was used to perform algorithm optimization comparison tests and compare them with other existing optimal harmony search algorithms [46–49]. This helped improve this algorithm's searching precision, achieving a linear acceleration ratio in parallel and a decrease in the iteration number with the convergence speed. A higher optimization efficiency was performed based on the experimental results through this algorithm than with various existing optimal harmony search algorithms.

## 3. Proposed Approach and Design Consideration

To improve the performance of the OLSR in the VANET, an optimization method, which is the improved harmony search (IHS) algorithm, is applied to the routing protocol OLSR, by defining and solving an optimization problem. The use of this method aims to address the standard OLSR parameters that were implemented without having any exact values for their ranges and seek to avoid having pointless configurations. The scenario evaluated in this section will demonstrate the employment of the method, as applied on moving cars in a highway scenario. Due to the constant changing of the network topology, the high speed of the nodes, and the continuous exchanging of messages throughout the network that causes a high overhead and dropping of packets, here we explore the scenario and apply the proposed IHS algorithm to address these issues.

*3.1. Harmony Search Algorithm with Selection Methods.* In this section, two selection schemes incorporated together in the memory consideration are presented, including the roulette wheel and tournament, as shown in Figure 5. The selection schemes proposed are altered in a way that applies to HS. These selection schemes were adopted in the memory consideration phase in a way that selects a variable from a solution in the harmony memory based on its fitness. The selection is performed according to the HMCR.

Step 1: initialize the problem and algorithm parameters. The optimization problem is specified as follows:

min $\{f(x) \mid x \in X\}$, where $f(x)$ is the objective function; $x = \{x_i \mid i = 1, \ldots, N\}$ is the set of decision variables; $X = \{X_i = 1, \ldots, N\}$ is the possible value range for each decision variable where $LB_i$, and $UB_i$ are the lower and upper bounds for decision variables, respectively. Besides, the HS algorithm parameters are specified in this step. These parameters are harmony solution vectors in the harmony memory; harmony memory considering rate (HMCR); pitch adjusting rate (PAR); distance bandwidth (BW); the number of decision variables ($n$) and the number of improvisations (NI), or the stopping criterion; memory size (HMS).

The optimization problem that has been considered in this research is the communication cost. The proposed optimization approach considers the quality-of-services-aware communications, so the main component of the objective function is to maximize the PDR and minimize both the E2E delay and overhead, and the solutions are guided by reducing the communication cost during the search process. As has been mentioned earlier, the OSLR is governed by its configuration parameters. Thus, the solutions are encoded as vectors with three parameters, which represent the timeouts before resending HELLO, Multiple Interface Declaration (MID), and Topology Control (TC) messages that describe the decision variables. These parameters have been chosen since they have the most impact on the protocol's functionality, as has been shown in the conducted related studies. The lower and upper bounds of these vectors have been set to be 1 and 15, respectively. The goal of these bounds is to adjust the timeouts runtime within these values according to the conditions of the network. For example, in a high mobility network (with frequent topology changes), it is desirable not to use high values for runtime before resending HELLO, TC, and MID to detect the changes in the network quickly. Therefore, we have chosen these bounds for the parameter to accommodate the high mobility of the highway environment by updating the routing table of the OLSR.

Step 2: initialize the harmony memory. The harmony memory is an augmented matrix of size $N\times$ HMS, which consists of sets of solution vectors determined by the HMS, where the value of HMS in this research is 40,

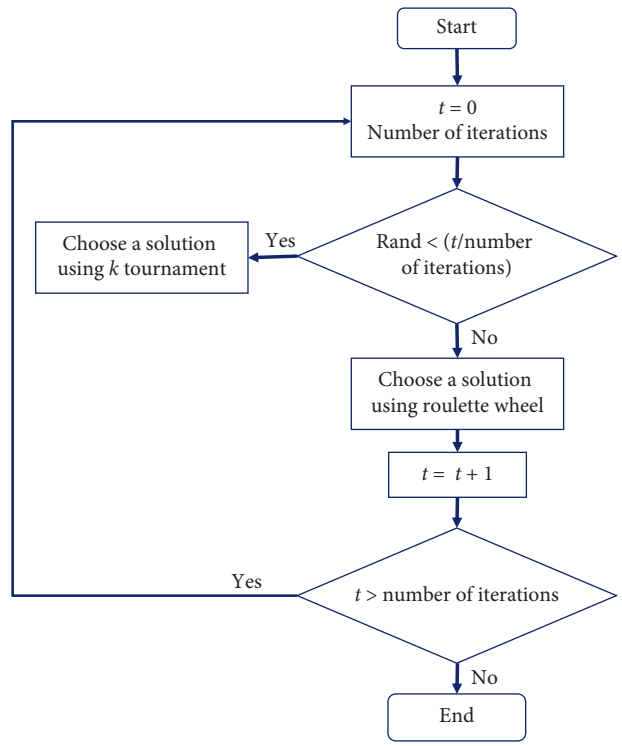


Figure 5: Selection of solutions using roulette and k-tournament methods.

and the number of $N$ is 3 (OLSR timeouts before resending parameters). In this step, these vectors are randomly generated as follows: $x_i^j = \mathrm{LB}_i + (\mathrm{UB}_i - \mathrm{LB}_i) \times \mathrm{Rand}[0, 1]$, and the rand returns a random number between 0 and 1, and $\mathrm{LB}_i$ and $\mathrm{UB}_i$ represent the lower and upper bounds of the decision parameter, respectively. The generated solutions are stored in the HM according to their objective function values.

$$\mathrm{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_N^2 \\ \vdots & \vdots & \cdots & \vdots \\ x_1^{\mathrm{HMS}} & x_2^{\mathrm{HMS}} & \ldots & x_N^{\mathrm{HMS}} \end{bmatrix}. \quad (1)$$

Step 3: improvise a new harmony; in the improvisation stage of the HS algorithm, there is no way to control the quality of the harmony selected from the HM; accordingly, any harmony of the HM can be a nominee. However, this way may harm the HS performance. To provide an effective way to select a note from the HM, two aspects must be considered. From the standpoint of the first aspect, using good harmonies for the improvisation, the process increases the probability of generating a new harmony with better quality. The second aspect takes into account the fact that harmonies with worse qualities may include some information in which the algorithm can be quickly converged to the global optimum. Therefore, in this research, a harmony selection base is proposed. The selection schemes proposed are altered in a way that applies to the HS. These selection schemes were adopted in the memory consideration phase in a way that selects a variable from a solution in the harmony memory. The selection is performed according to the HMCR, and it is used to choose one variable from the solutions in the harmony memory. The improvised solution is based on the HMCR and PAR values. First, a random number is generated in the range, from 0 to 1. If the rand is bigger than the HMCR value, then improvise randomly. Else select the rand based on roulette and k-tournament selection features. The following flow chart will demonstrate the selection process of the pitches where t is the current iteration.

The condition of improvising a new solution when the random solution is lower than the HMCR; the improvisation of a solution based on choosing between the two selection methods depends on the current number of iteration. At the beginning of the execution, the number of the iteration is low, which means that the condition of the random solution is bigger than the current iteration, which will not be valid. As a result, the selection will be based on tournament selection. In this case, $K$ (which is set to 10 in our research) individuals are sampled at random to enter the tournament. The HS will select the best solution from the solutions that participated in the tournament, where tournament size determines the selective pressure; the higher the value of $K$, the stronger the selective pressure. These selection methods have the advantage of being insensitive to fitness scaling problems and the sense of optimization. After a while, the number of iteration will be increased, which means that the random solution will be chosen based on the roulette wheel. The probability of selecting an individual is proportional to its fitness. This selection technique allows the selection of the best individuals with greater probability, but at the same time, the worst ones could be selected too. In this way, exploration space will be more considered since the roulette wheel tends to choose the solutions based on their fitness values. This is effective in selecting the best solution. Still, the best solution might be found in the neighborhood, and not be global-based. Therefore, to avoid local minima, the adoption of k-tournament at the beginning of the execution of the optimization method that enables more random selection of the solutions to a certain number of iterations will lead to the exploration for better solutions in the search space. The roulette wheel is adopted to converge toward the best solution. It means that adopting the two selection methods will enable better exploration and convergence to the solutions in the memory.

Step 4: update the harmony memory. In this step, the decision on whether the new harmony vector improvised in step 3 should be included in the harmony memory is made judged by the objective function (communication cost). If the fitness of the new harmony vector is better than the worst harmony vector in the current harmony memory, it will be included in the HM. Meanwhile, the existing worst harmony will be excluded from the HM.

Step 5: repeat Steps 3 and 4 until the stopping criterion is met.

*3.2. Harmony Search Optimization with Optimized Link State Routing Protocol.* As mentioned earlier, the optimization strategy that was utilized to obtain efficient OLSR parameter configurations automatically was performed by coupling two different stages: (1) a procedure for optimization and (2) a simulation stage. A metaheuristic method was carried out by an optimization block, which in this case is the IHS algorithm, which has been developed to search for optimal (or near-optimal) solutions within continuous search spaces. In this case, that is represented by this research. A simulation procedure was utilized to assign a quantitative quality value (fitness) to the computed configurations' (refresh intervals) OLSR performance in terms of the communication cost. For this, MATLAB network simulation was used to perform this procedure. For this particular research, MATLAB was modified so that it can automatically interact with the optimization procedure and therefore accept new routing parameters. When the meta-heuristic process is used, it requires an assessment of the solution; the tentative OLSR configuration's simulation procedure is invoked over the defined VANET scenario. MATLAB was then started so that it can evaluate the VANET based on the circumstances that

have been defined by the OLSR routing parameters, which are shown in Figure 6. These circumstances and parameters were generated using the optimization algorithm. Once the simulation is completed, MATLAB can then produce global information about the packet overhead, the PDR, and the E2E delay of the entire mobile vehicular network scenario. In turn, this information is used to calculate the communication cost (comm_cost) function based on the following:

$$\text{minimize comm}_{\text{cost}} = w3 \cdot \text{E2E Delay} + w2 \cdot \text{Overhead} - w1 \cdot \text{PDR}. \tag{2}$$

The communication cost function is used to represent the fitness function of this research's optimization problem. To enhance the QoS, one will have to maximize the PDR and minimize both the E2E Delay and overhead. As seen in equation (2), an aggregative minimizing function was used. Because of this, the formulation of the PDR was done with a negative sign. In addition, for this Equation, factors $w1$, $w2$, and $w3$ were utilized to weigh each metric's influence on the resultant fitness value. Since the goal of this research is to promote the PDR for the sake of efficient packet communication, it was decided that different biased weights will be used in the fitness function, being $w1 = 0.5$, $w2 = 0.2$, and $w3 = 0.3$. With this, the PDR will have more priority compared to packet overhead and E2E delay [14, 17].

## 4. Simulation Setup and Parameter Configuration

One of the most important and critical tasks to solve the problem is to obtain the metrics necessary to evaluate the performance of a particular routing protocol configuration. This evaluation has been carried out by using the simulator MATLAB (Matrix Laboratory), a high-performance language aimed at technical computing, assimilating computation, visualization, and programming amid a simple-to-use environment where problems, as well as solutions, are articulated in a familiar mathematical notation.

To obtain results close to the real world, we have defined a simulation VANET scenario (instances) from an area of Bangi in Malaysia (Figure 7), creating highway scenarios. We have described this distinct scenario since the characteristics of the movement of vehicles for any scenario are different enough to affect the transfer of files. For example, in urban areas, vehicle density is higher, and these vehicles travel at lower speeds than in interurban environments, increasing the likelihood that the transfers are carried out successfully in urban areas than in the highways. Therefore, we can analyze in our highway scenario the behavior and performance of the compared algorithms, as well as the differences in the resulting OLSR configurations in terms of communication efficiency. Furthermore, we can compare these automatically generated configurations against the ones used in previously conducted studies and the original configurations.

The highway scenario coverage of 500×500 $m_2$ area was considered to inspect the configured OLSR parameters' performance in high density and velocity. The fast-changing topology nature has done the characterization, and we have set the number of nodes to be 70. Each vehicle moves at speeds ranging between 70 and 100 km/h, with nonrandom movement patterns, as the constraints of the user map dictate the nodes.

Furthermore, the simulation time for the conducted simulation is set to be 1500 seconds, with a data counter of 60 bits for nodes transmitting packets and the routing table size being set to be 100 bits, where the communications between the vehicles being conditioned by the use of IEEE 802.11b standard network interfaces. Finally, nodes communicate within a radius of 150 m. The other related parameters are presented in Table 3.

*4.1. Performance Metrics.* Performance metrics are presented in several ways, such as two basic performance metrics that run by a delivery fraction of packet and E2E delay that has been proposed [50, 51]. Apart from that, with the consideration of the mobility pattern of nodes, Manjula et al. suggested using the random waypoint mobility model in terms of delay and packet drop [52].

*4.1.1. Packet Delivery Ratio.* The packet delivery ratio or the fraction of data packets that come from an application is delivered completely and correctly by the destination. The PDR plays a vital role in every routing protocol, as there is no margin for errors to have occurred, especially in real-world environments. For evaluating the HSO, with the OLSR in terms of PDR, equation (3) is used:

$$\text{PDR} = \frac{\text{total number of recieved packets}}{\text{total number of sent packets}} \times 100. \tag{3}$$

*4.1.2. End to End Delay (E2E Delay).* E2E refers to the difference between the time that an application sends a data packet and the time that the destination receives this packet. To evaluate the effectiveness of the HSO with the OLSR in terms of E2E Delay, equation (4) is used:

$$\text{AVG.E2E (ms)} = \frac{\text{total E2E}}{\text{number of sent packets}}. \tag{4}$$

*4.1.3. Overhead.* Overhead is the network routing load or the ratio of transmissions of the administrative routing packet and the data packets delivered where every hop is separately counted. For evaluating the effectiveness of the HSO with the OLSR in terms of the network overhead ratio, equation (5) is used:

$$\text{overhead} = \frac{\text{number of packets sent}}{\text{E2E}} \times 100. \tag{5}$$

*4.1.4. Energy Consumption.* The variance of energy is how much energy each node in the network consumes in its

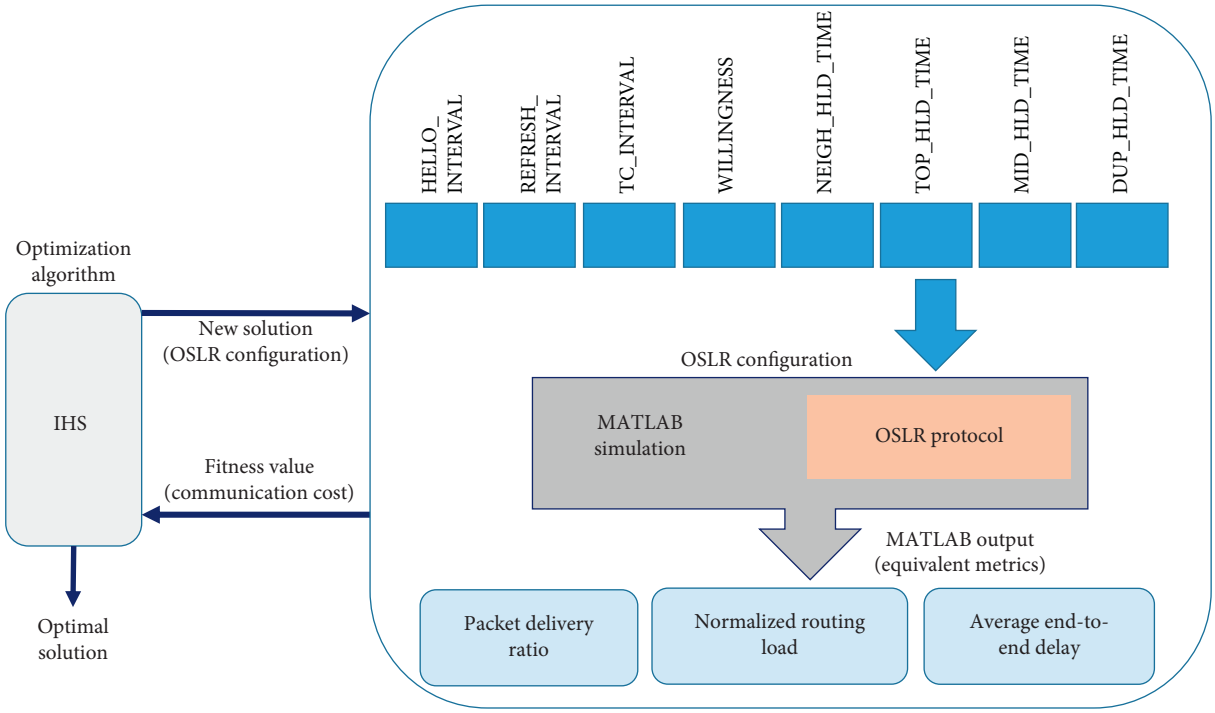


Figure 6: OLSR with IHS in VANET.


Figure 7: The selected area of Bangi in Malaysia for the experimental simulation scenario.

Table 3: Parameterization of HS.

| Parameter | Value |
|---|---|
| Pitch adjustment rate | 0.3 |
| Harmony memory consideration rate | 0.9 |
| Harmony memory size | 5 |
| Lower bounds | 1 |
| Higher bounds | 15 |
| Decision variables | 3 |
| Distance bandwidth | 0.5 |
| Number of iterations | 1 |

lifetime. For evaluating the effectiveness of HSO with the OLSR in terms of energy consumption, equation (6) is used:

$$\mu = \text{mean}(x),$$
$$\mu(X) = \frac{\sum (xi - \mu)^2}{N - 1}, \tag{6}$$

where $N$ is nodes number, $x_i$ is consumed energy by the $i^{\text{th}}$ node, and $\mu$ is the mean of energy consumptions for all nodes.

## 5. Results Analysis and Discussion

We compare the OLSR configurations in terms of the network's quality of services (PDR, E2E, Overhead, and Variance of energy). To start with, we can see in Table 4 the OLSR parameters settings considered for comparison in this analysis. In this table, column 2 contains the OLSR configurations of the standard OLSR proposed by the RFC 3626; and columns 3–5 contain the OLSR configurations obtained by each one of the meta-heurisitc algorithms studied in this research:improved harmony search (OLSR-IHS), harmony search algorithm (OLSR-HS), and particle swarm optimization, respectively. As it has been mentioned earlier, the OLSR is regulated by these configurations, where the first three intervals are responsible for the timeout values before resending hello, topology control, and mid-times, which are responsible for updating the protocol routing table in case there are any changes in the topology and link breakage. The second three intervals are the validity timeout values that are responsible for the willingness of nodes of rerouting packets and how long it takes before dropping these packets.

Figure 8 shows the PDR (the total number of correctly delivered packets from source to destination) results obtained by simulating the original OLSR parameter configurations and the obtained parameter configurations by our OLSR_IHS, OLSR, the OLSR_HS, and PSO-OLSR on 70 nodes moving in our defined highway scenario. It can be seen that a lag had occurred during the simulation time by a 0.48 ratio that caused the $Y$-axis not to make contact. This is due to the time needed to collect the data and generate the curve for the PDR.

It can be seen that the OLSR with the optimized configurations obtained by our OLSR-IHS gives a high ratio of PDR at the beginning of the simulation. It decreases gradually to about 0.4625 ratios at time 150 sec and remains between 0.425 and 0.465 ratios until the end of the simulation time. Meanwhile, the results obtained by the OLSR parameter configurations achieve about a 0.46 ratio at the beginning of the simulation. Then it decreases, reaching below 0.43 ratio and starts increasing, not exceeding 0.45 ratio until the end of the simulation time. The final PDR results are obtained by simulating the HS-OLSR based on the optimized parameter configurations returned by the basic HS, where it achieves about 0.40 ratio at the beginning of the simulation and decreases gradually to about 0.35 ratio. Then, it increases gradually, not exceeding 0.35 ratio until the end of the simulation time. It can be observed that the PDR of the OLSR with our optimized parameter configurations returned by our approach has outperformed both the original OLSR and the HS-OLSR in terms of the total delivered packets; however, the PSO-OLSR has achieved better results than OLSR-IHS since the time for refresh intervals is the shortest. The original OLSR parameters and the optimized parameters obtained by the IHS and PSO are better than those of HS-OLSR since the refresh intervals, and MID-H-T is the longest, which means that its obtained parameters are not effective for routing and neighboring discovery.

Table 4: Original OLSR and the obtained OLSR parameters using optimization methods.

| Parameter | OLSR | OLSR-IHS | OLSR-HS | OLSR-PSO |
|---|---|---|---|---|
| HELLO-INT | 2 | 1 | 2 | 1 |
| MID-INT | 2 | 5 | 5 | 4 |
| TC-INT | 5 | 15 | 14 | 5 |
| WILLING | 3 | 3 | 3 | 3 |
| NEIG_H_T | 6 | 3 | 6 | 3 |
| TOP_H_T | 15 | 45 | 42 | 15 |
| MID_H_T | 15 | 45 | 42 | 15 |
| DUP_H_T | 30 | 30 | 30 | 30 |

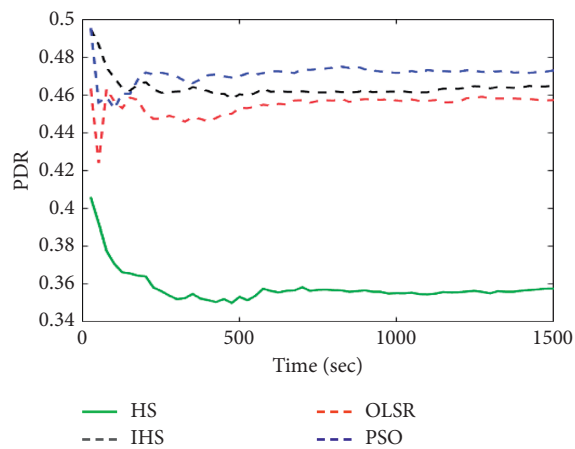


Figure 8: PDR simulation results.

Figure 9 shows the E2E delay (the time needed for the packets to be delivered) results obtained by simulating the original OLSR parameters along with the optimized parameter configurations returned by our IHS-OLSR, PSO-OLSR, and HS-OLSR. A lag had occurred during the simulation due to the time needed to collect and generate the results, which resulted in a disconnected $Y$-axis at a 0.48 ratio.

The E2E Delay performance for the OLSR with the optimized parameters configuration obtained by the HIS. It starts with a high E2E delay at the beginning of the simulation by 0.39 ms. Then, it decreases gradually until the time of simulation is about 750 sec, achieving 0.13 ms. After a while, it again keeps reducing simulation time, reaching 0.09 ms at the end. Meanwhile, the results of the E2E delay obtained by the time intervals returned by the OLSR start at 0.23 ms at the beginning of the simulation. It starts decreasing, achieving 0.06 ms, and it stays between 0.06 ms and 0.05 ms until the end of the simulation time. Finally the E2E delay performance for the OLSR-based optimized parameters configuration has done by HS, where the minimum E2E Delay was 0.9ms at the very beginning. It profoundly decreases at the time between 200 sec and 500 sec achieving 0.11 ms, and it keeps dropping until the end of the simulation time. According to the E2E Delay metric, it can be seen that the configurations by the OLSR and the PSO-OLSR have achieved the minimum time needed for the packets to

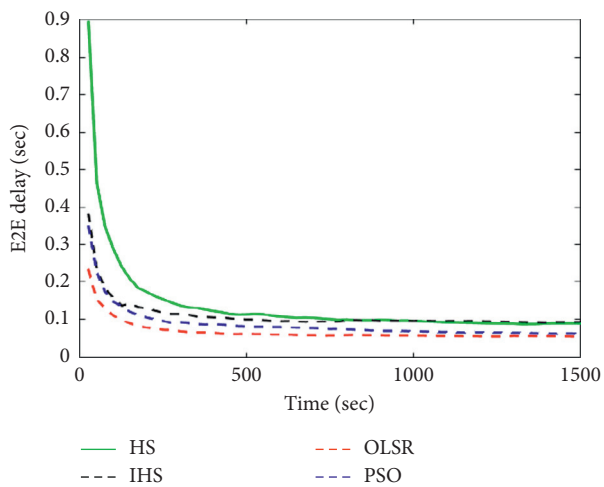


Figure 9: E2E Delay simulation results.

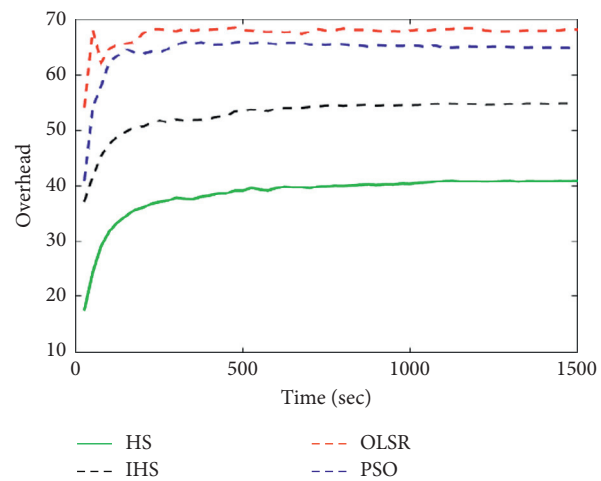


Figure 10: Overhead simulation results.

be transmitted from a source to the destination when compared with our approach and the HS; this is due to the short time intervals of Hello, TC, and MID messages, and specifically the Hello messages. Note that the mobility of the network in our scenario is high, as the link-changing rate is constant; therefore, Hello messages need to be resent more periodically. Thus, the PSO has outperformed the existing approaches to detect the link failure and discovering the alternative links to forward the packets whenever packets are dropped.

Figure 10 shows the overhead (the total number of generated packets in the network) obtained results for the OLSR based on the original OLSR parameter configurations and the optimized parameter configurations returned by OLSR-IHS, PSO-OLSR, and HS-OLSR. It can be seen that a lag had occurred during the simulation at a 0.48 ratio, resulting in $Y$-axis not making contact due to the time needed to collect and generate the results.

As can be seen from the figure, the OLSR based on the configurations returned by our approach has outperformed the original OLSR and the PSO-OLSR, as it achieves about 36 at the beginning of the simulation and then starts increasing gradually, reaching its highest peak at 54. Meanwhile, the original OLSR curve begins at 55, and it reaches its peak above 68 at the time of 25 sec. At time 100 sec, the curve shows the overhead increases going to 66 at time 250 sec, and it stays between 66 or 68 until the end of the simulation. The parameter configurations of the OLSR returned by the HS has outperformed the configurations of both the original OLSR and the OLSR-HIS. It achieved less than 20 at the beginning of the simulation until it reached the peak, resulting in an overhead above 40 at the end of the simulation time; this is due to the constant generation of Hello and TC messages, which has led to network congestion. Note that the network is continuously changing. Thus, the number of generated Hello and TC messages is more to adapt to these changes. Since the HS has obtained the longest refresh intervals (the timeout values before resending), it generated fewer packets whenever a change or failure occurred. Thus, it has achieved the least overhead when compared with the other approaches.

Figure 11 shows the results of simulating the optimized parameters obtained by OLSR-IHS, HS-OLSR, PSO-OLSR, and the original OLSR parameter configurations in terms of the variance of energy metrics by showing how much energy is consumed by the nodes in the network. It can be observed that a lag had occurred during the simulating of the results by 0.48 ratio due to the time needed to collect and generate the data, resulting in the $Y$-axis not making contact.

It can be seen that the optimized OLSR parameters obtained by the HS outperformed the parameters returned by our approach and the original OLSR parameters and the PSO-OLSR. This is due to the fact that it had acquired the worst results regarding the PDR and the E2E delay since the refresh intervals parameters are the longest, which means it did not generate many HELLO, TC, and MID messages. Thus, the nodes did not consume much energy. Regarding the parameters returned by our approach, it can be seen that the energy consumed is lesser than the OLSR and the PSO-OLSR, since the new configurations of the OLSR-IHS achieved a better trade-off between the delivered packets and the network's overhead when the time difference between its refresh time intervals are longer than both the OLSR and the PSO-OLSR.

In summary, we can say that the optimized OLSR configurations obtained by the improved harmony search algorithm have given promising results compared with the original OLSR configurations and the configurations obtained by the basic harmony search and particle swarm optimization methods. Also, it was able to achieve the best trade-off between the network performance metrics (PDR, E2E, overhead, and variance of energy). Concerning the PDR metric, it can be observed that the proposed approach has outperformed both the original OLSR and the basic harmony search. This is because the proposed approach could obtain the parameters configuration for accommodating fast-changing in the network topology. In addition, the Harmony search has achieved the lowest delivered

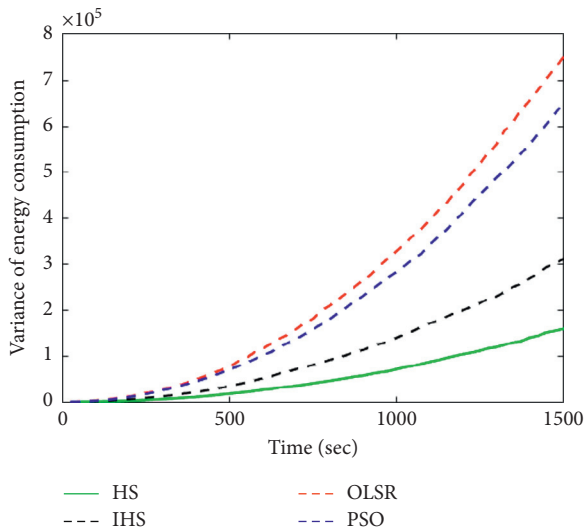


Figure 11: The variance of energy consumption.

packets throughout the network. This is because the basic algorithm as an optimization method has failed to obtain a set of configurations that can improve the routing protocol's performance in a highway scenario. It has randomly chosen the parameters with no guided process, which led it to choose the nonfittest set of variables. In contrast, the proposed approach has overcome this issue by selecting the fittest solutions (parameters) based on the principle of choosing solutions with a higher probability of generating new harmonies (solutions). Hence, it reacted to sending more HELLO messages to detect the network changes by obtaining lower values of resending timeouts. Finally, when comparing the particle swarm with our approach in terms of packet delivery, E2E delay, and energy consumption achieved better results in low simulation time. This is because the PSO has obtained the lowest HELLO-interval when compared with the other approaches. Thus, the configurations were more successful in generating more packets whenever a change in the topology or a link breakage occurred by sending more hello messages to detect them. Furthermore, concerning how long the packets have taken to reach the network's destinations, the configurations obtained by the proposed approach, the PSO and the OLSR, were able to outperform the basic harmony configurations search. This is because the harmony search needed the longest times to hold the packets since the parameters' validity times are the highest. The nodes held the packets for longer before sending any updates on the topology changing, and link failures as an invalid link in the OLSR cause more delay to the routing table's recalculation, which is the main feature of the simulated scenario that has been used in this research. Moreover, when it comes to the amount of overhead that causes traffic in the network, the configurations of our approach have achieved better performance compared with the configurations obtained by the PSO and the OLSR, given that our approach properly maintained the trade-off between the total number of generated packets and the corrected packets. This is because the improved harmony search was able to react better to the topology changes by acquiring the best set of configurations regarding the times needed to send HELLO messages and TC messages. Thus, the nodes were able to maintain the updates of their routing by producing lesser packets. Simultaneously, the calculations of forwarding these packets were better compared to the approaches. Finally, the energy consumptions by the nodes in the network obtained optimized configurations by our proposed approach. It was able to outperform the configurations of the PSO and the original OLSR, as the optimization method achieved lesser overhead compared with the other approaches. However, the configurations obtained by the basic harmony search helped achieve the least consumed energy, since it produced the minimum number of packets as the nodes did not react to the changes in the topology and failures to resending the packets due to link breakage. Generally, the improved harmony search is suitable for highway scenarios. It has obtained promising results, generating the packets and routing them correctly without consuming the network resources by achieving short times to route the packets and updating the nodes routing the tables. The network is undergoing constant changes. We conclude that parameters related to discovery intervals have a far more significant impact on route quality from the results obtained. Even within the limits of available network bandwidth, we see that it can improve the route availability in the OLSR protocol, just by decreasing the refresh intervals for the HELLO messages and the TC messages. The HELLO interval has more effect on improving the route availability than the TC interval among the refresh intervals. The HELLO interval and the TC interval decide the rate at which the HELLO messages and the TC messages are exchanged in the network. We found this to be good news from the implementation perspective because decreasing the HELLO interval alone is likely to affect the network bandwidth compared to decreasing the TC interval. The HELLO messages are only transmitted in the neighborhood. In contrast, the TC messages are transmitted throughout the network.

## 6. Conclusions

This work investigates the evaluation performance of the OLSR routing protocol to auto-tune the parameters of this routing protocol and finding the best configurations suited for improving the performance of routing protocols for vehicular ad hoc networks, which is known for their unique and challenging features. Due to these features, it is essential to design an efficient and reliable routing strategy, as it is one of the most challenging problems in this field for this kind of network. A method for flexible routing is required because of the dynamic nature of the VANET, such as network topology, the density of the nodes, and the high speed of the nodes. This research has developed an improved metaheuristic algorithm, Harmony Search Optimization, based on coupling two selection methods and investigates using it to optimize the OLSR parameters for the VANET. The simulation results, implemented on a highway scenario, showed that the IHS could find better OLSR parameter

configurations. Besides, three other experiments using the original OLSR and the optimized OLSR using the basic harmony search algorithms and particle swarm optimization have been used as a benchmark. The results have shown an overall improvement in comparison with the other algorithms. The proposed improved harmony search algorithm outperformed the original HS and the original OLSR in the packet delivery ratio and has less overhead than the OLSR and particle swarm optimization algorithm.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research was supported by the Universiti Kebangsaan Malaysia (UKM) under the research grants GGPM 2020-028 and GGP-2019-023.

## References

[1] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13–64, 2003.

[2] P. Ghosekar, G. Katkar, and P. Ghorpade, "Mobile ad hoc networking: imperatives and challenges," *IJCA Special Issue on MANETs*, vol. 3, pp. 153–158, 2010.

[3] V. Kumar, S. Mishra, and N. Chand, "Applications of VANETs: present & future communications and network," *Communications and Networks*, vol. 5, no. 1, pp. 12–15, 2013.

[4] R. Hussain, J. Lee, and S. Zeadally, "Trust in VANET: a survey of current solutions and future research opportunities," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–19, 2020.

[5] M. Oche, A. B. Tambuwal, C. Chemebe, R. M. Noor, and S. Distefano, "VANETs QoS-based routing protocols based on multi-constrained ability to support ITS infotainment services," *Wireless Networks*, vol. 26, no. 3, pp. 1685–1715, 2020.

[6] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular ad hoc networks (VANETs): challenges and perspectives," in *Proceedings of the 2006 6th International Conference on ITS Telecommunications*, pp. 761–766, Chengdu, China, June 2006.

[7] B. Yu and C. Xu, "Vehicular ad-hoc networks: an information-centric perspective," *ZTE Communications*, vol. 8, pp. 42–49, 2020.

[8] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: a survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007.

[9] A. Awang, K. Husain, N. Kamel, and S. Aissa, "Routing in vehicular ad-hoc networks: a survey on single- and cross-layer design techniques, and perspectives," *IEEE Access*, vol. 5, pp. 9497–9517, 2017.

[10] C. Serrano-Cinca, Y. Fuertes-Callén, and C. Mar-Molinero, "Measuring DEA efficiency in internet companies," *Decision Support Systems*, vol. 38, no. 4, pp. 557–573, 2005.

[11] K. Chen, Y.-Y. Song, J.-F. Pan, and G.-L. Yang, "Measuring destocking performance of the Chinese real estate industry: a DEA-Malmquist approach," *Socio-Economic Planning Sciences*, vol. 69, Article ID 100691, 2020.

[12] T. Clausen, P. Jacquet, C. Adjih et al., "Optimized link state routing protocol (OLSR)," 2003.

[13] M. T. Lwin, J. Yim, and Y.-B. Ko, "Blockchain-based lightweight trust management in mobile ad-hoc networks," *Sensors*, vol. 20, no. 3, p. 698, 2020.

[14] J. Toutouh, J. García-Nieto, and E. Alba, "Intelligent OLSR routing protocol optimization for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1884–1894, 2012.

[15] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proceedings of the IEEE International Multi Topic Conference*, Lahore, Pakistan, December 2001.

[16] S. Goudarzi, M. H. Anisi, H. Ahmadi, and L. Mousavian, "Dynamic resource allocation model for distribution operations using SDN," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 976–988, 2020.

[17] J. Toutouh, S. Nesmachnow, and E. Alba, "Fast energy-aware OLSR routing in VANETs by means of a parallel evolutionary algorithm," *Cluster Computing*, vol. 16, no. 3, pp. 435–450, 2013.

[18] R. C. Muniyandi, F. Qamar, and A. N. Jasim, "Genetic optimized location aided routing protocol for VANET based on rectangular estimation of position," *Applied Sciences*, vol. 10, no. 17, p. 5759, 2020.

[19] S. Islam, O. O. Khalifa, A. H. A. Hashim, M. K. Hasan, M. A. Razzaque, and B. Pandey, "Design and evaluation of a multihoming-based mobility management scheme to support inter technology handoff in PNEMO," *Wireless Personal Communications*, vol. 114, no. 2, pp. 1133–1153, 2020.

[20] M. Karimi, H. R. Naji, and S. Golestani, "Optimizing clusterhead selection in wireless sensor networks using genetic algorithm and harmony search algorithm," in *Proceedings of the 20th Iranian conference on electrical engineering (ICEE 2012)*, pp. 706–710, Tehran, Iran, May 2012.

[21] K. V. Patil and M. Dhage, "The adaptive optimized routing protocol for vehicular ad-hoc networks," *IJCSN International Journal of Computer Science and Network*, vol. 2, pp. 67–70, 2013.

[22] A. Kots and M. Kumar, "The fuzzy based QMPR selection for OLSR routing protocol," *Wireless Networks*, vol. 20, no. 1, pp. 1–10, 2014.

[23] T. Stephan and K. Karuppanan, "Cognitive inspired optimal routing of OLSR in VANET," in *Proceedings of the 2013 International Conference on Recent Trends in Information Technology (ICRTIT)*, pp. 283–289, Chennai, India, July 2013.

[24] M. K. Hasan, A. F. Ismail, S. Islam, W. Hashim, M. M. Ahmed, and I. Memon, "A novel HGBBDSA-CTI approach for subcarrier allocation in heterogeneous network," *Telecommunication Systems*, vol. 70, no. 2, pp. 245–262, 2019.

[25] O. A. Wahab, H. Otrok, and A. Mourad, "VANET QoS-OLSR: QoS-based clustering protocol for vehicular ad hoc networks," *Computer Communications*, vol. 36, no. 13, pp. 1422–1435, 2013.

[26] S. H. Park, Y.-H. Jang, Z. W. Geem, and S.-H. Lee, "CityGML-based road information model for route optimization of snow-removal vehicle," *ISPRS International Journal of Geo-Information*, vol. 8, no. 12, p. 588, 2019.

[27] D. Al-Terri, H. Otrok, H. Barada, M. Al-Qutayri, R. M. Shubair, and Y. Al-Hammadi, "QoS-OLSR protocol

based on intelligent water drop for vehicular ad-hoc networks," in *Proceedings of the 2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1352–1357, Dubrovnik, Croatia, August 2015.

[28] K. Prakash, P. C. Philip, R. Paulus, and A. Kumar, "A packet fluctuation-based OLSR and efficient parameters-based OLSR routing protocols for urban vehicular ad hoc networks," in *Recent Trends in Communication and Intelligent Systems*, pp. 79–87, Springer, Berlin, Germany, 2020.

[29] K. Prakash, P. C. Philip, R. Paulus, and A. Kumar, "A packet fluctuation-based OLSR and efficient parameters-based OLSR," *Recent Trends in Communication and Intelligent Systems: Proceedings of ICRTCIS 2019*, p. 79, Springer, Berlin, Germany, 2020.

[30] R. Mehra, R. S. Bali, and P. Kaur, "Efficient clustering based OLSR routing protocol for VANET," in *Proceedings of the 2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, pp. 1–7, Indore, India, March 2016.

[31] W. Li, W. Du, W. Tang, Y. Pan, J. Zhou, and Z. Lin, "Parallel algorithm of multiobjective optimization harmony search based on cloud computing," *Journal of Algorithms & Computational Technology*, vol. 11, no. 3, pp. 301–313, 2017.

[32] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, pp. 60–68, 2001.

[33] T. Zhang and Z. W. Geem, "Review of harmony search with respect to algorithm structure," *Swarm and Evolutionary Computation*, vol. 48, pp. 31–43, 2019.

[34] C.-M. Wang and Y.-F. Huang, "Self-adaptive harmony search algorithm for optimization," *Expert Systems with Applications*, vol. 37, no. 4, pp. 2826–2837, 2010.

[35] L. Wang, W. Ye, Y. Mao, P. G. Georgiev, H. Wang, and M. Fei, "The node placement of large-scale industrial wireless sensor networks based on binary differential evolution harmony search algorithm," *International Journal of Innovative Computing, Information and Control*, vol. 9, pp. 955–970, 2013.

[36] A. I. Amin, H. S. Hamza, and I. A. Saroit, "Harmony search based algorithm for dynamic shortest path problem in mobile ad hoc networks," in *Proceedings of the Third International Conference on Digital Information and Communication Technology and Its Applications (DICTAP2013)*, Ostrava, Czech Republic, July 2013.

[37] O. Moh'd Alia, Z. Shaaban, A. Basheer, A. Al-Ajouri, and A. Alsswey, "Musicians'-inspired clustering protocol for efficient energy wireless sensor networks," in *Proceedings of the Fourth International Conference on Communications and Networking, ComNet-2014*, pp. 1–6, Ḥammāmāt, Tunisia, March 2014.

[38] L. Wang and L.-P. Li, "An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems," *International Journal of Electrical Power & Energy Systems*, vol. 44, no. 1, pp. 832–843, 2013.

[39] M. Castelli, S. Silva, L. Manzoni, and L. Vanneschi, "Geometric selective harmony search," *Information Sciences*, vol. 279, pp. 468–482, 2014.

[40] H. Li, P.-C. Shih, X. Zhou, C. Ye, and L. Huang, "An improved novel global harmony search algorithm based on selective acceptance," *Applied Sciences*, vol. 10, no. 6, p. 1910, 2020.

[41] J. Yi, L. Gao, X. Li, and J. Gao, "An efficient modified harmony search algorithm with intersect mutation operator and cellular local search for continuous function optimization problems," *Applied Intelligence*, vol. 44, no. 3, pp. 725–753, 2016.

[42] M. A. Al-Betar, M. A. Awadallah, A. T. Khader, A. L. a. Bolaji, and A. Almomani, "Economic load dispatch problems with valve-point loading using natural updated harmony search," *Neural Computing and Applications*, vol. 29, no. 10, pp. 767–781, 2018.

[43] T. Shankar, S. Shanmugavel, and A. Rajesh, "Hybrid HSA and PSO algorithm for energy efficient cluster head selection in wireless sensor networks," *Swarm and Evolutionary Computation*, vol. 30, pp. 1–10, 2016.

[44] M. K. Hasan, M. M. Ahmed, A. H. A. Hashim, A. Razzaque, S. Islam, and B. Pandey, "A novel artificial intelligence based timing synchronization scheme for smart grid applications," *Wireless Personal Communications*, vol. 114, no. 2, pp. 1067–1084, 2020.

[45] H. Sallehudin, R. Razak, M. Ismail, A. Fadzil, and R. Baker, "Cloud computing implementation in the public sector: factors and impact," *Asia-Pacific Journal of Information Technology and Multimedia*, vol. 7, pp. 27–42, 2019.

[46] A. Ghazvini, S. N. H. S. Abdullah, M. Kamrul Hasan, and D. Z. A. Bin Kasim, "Crime spatiotemporal prediction with fused objective function in time delay neural network," *IEEE Access*, vol. 8, pp. 115167–115183, 2020.

[47] H. Q. Ghadhban and R. C. Muniyandi, "Improved Kalman filter based LAR in vehicular ad hoc network," *International Review on Modelling and Simulations (IREMOS)*, vol. 9, no. 5, pp. 361–366, 2016.

[48] M. Ali, R. C. Muniyandi, E. Sundararajan, and A. M. Zin, "A parallel membrane inspired harmony search for optimization problems: a case study based on a flexible job scheduling problem," *Applied Soft Computing*, vol. 49, pp. 120–136, 2016.

[49] M. Raad Hammoodi and R. Chandren Muniyand, "An improved harmony search algorithm for optimized link state routing protocol in vehicular ad hoc network," *International Journal of Engineering & Technology*, vol. 7, no. 2, pp. 177–181, 2018.

[50] K. Al-Begain, I. Awan, and D. D. Kouvatsos, "Analysis of GSM/GPRS cell with multiple data service classes," *Wireless Personal Communications*, vol. 25, no. 1, pp. 41–57, 2003.

[51] M. Sundberg, N. Johansson, P. Schliwa-Bertling, and J. W. Diachina, "Wireless communications—dynamic coverage class update and aligning coverage class paging groups," Google Patents, 2019.

[52] S. Manjula, C. Abhilash, K. Shaila, K. Venugopal, and L. Patnaik, "Performance of AODV routing protocol using group and entity mobility models in wireless sensor networks," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, pp. 1212–1217, Kowloon, Hong Kong, March 2008.