Reactive Topology-based Routing for VANETs: A New Bio-inspired Solution

An Evolutionary Algorithm-based Vehicular Reactive Protocol With Route Break Prediction Mechanism

Youcef Azzoug
Faculty of Electronic and Informatic
USTHB University, Bp 32 El Alia 16111 Bab Ezzouar
Algiers, Algeria
e-mail: yazzoug@usthb.dz

Abdelmadjid Boukra
Faculty of Electronic and Informatic
USTHB University, Bp 32 El Alia 16111 Bab Ezzouar
Algiers, Algeria
e-mail: aboukra@usthb.dz

***Abstract*—Vehicular Ad-hoc Network (VANET) routing took its early bases from Mobile Ad-hoc Network (MANET) originated routing protocols starting especially by enlarging topology-based protocols that have been conceived for MANETs, which suffered numerous modifications to make them applicable on vehicular mobility patterns. Topology-based routing is divided between proactive and reactive approaches. The latter came to fulfill the shortages of the former. In this paper, the inspiration is got from few enhanced versions of vehicular reactive topology-based routing in their two forms hop-by-hop and source routing, particularly from Ad-hoc On-demand Distance Vector Routing (AODV) and Dynamic Source Routing (DSR) originated routing protocols such as the Prior AODV and Enhanced DSR, to combine their advantages. Following this axis, an evolved Evolutionary Algorithm (EA) is exploited, in this case the Bull Optimization Algorithm (BOA): an ameliorated Genetic Algorithm (GA) incorporated with few conventional modifications inspired from the aforementioned protocols to improve the Quality of Service (QoS) routing performances. The suggested routing algorithm offers a new solution for assisting the source routing forwarding for vehicular networks.**

***Keywords*—source routing; VANETs; P-AODV; G-NET; Bull Optimization Algorithm.**

I. INTRODUCTION

The reactive routing destined for VANETs [1] has been the context of a wide research derivation, tending for improving the operational routing mechanisms, for instance the route discovery, route repair process, neighboring table optimization, beaconing, route refreshing among others. This routing model suggests two types of on-demand forwarding strategies: hop-by-hop (or next-hop) routing and source routing. For the first, each data packet tracks the destination by keeping two addresses in its header: the next-hop and destination's Identifiers (IDs). These two IDs are recorded in the routing table. This mechanism helps to perform an efficient periodic refreshing using beacons [2] to the best routes to any destination, thus an easy adaptability to quick topology changes. On the other hand, hop-by-hop forwarding may accumulate congesting routing overheads due to beaconing which is responsible for tracking neighborhood connectivity evolution. In the source routing, every data packet loads its route hops' IDs in its header to attain the destination, that does not require beaconing, the fact that packet forwarding depends mainly on intermediate hops' available full routing paths. This mechanism entails an alleviated bandwidth occupation due to elimination of beaconing-originated overheads, but such routing policy becomes less reliable when confronting to quick mobility of nodes, which characterize vehicular networks, especially for longer routes where topology changes are more frequent, hence higher number of dropped packets. AODV [3] is a typical hop-by-hop reactive protocol, whereas DSR protocol [4] is a classic routing algorithm of source routing.

This paper proposes a new bio-inspired routing algorithm to enhance the reactive routing under VANET environments based on an enhanced EA, in this case the BOA metaheuristic [5] that seeks generating better genetically-generated routing paths after the route discovery to manage data forwarding better under unpredictable vehicular mobility and drastic network density changes. The suggested solution is conceived on a multipath DSR-based route discovery procedure and completed with few AODV-evolved routing up-to-date operations.

The following context of this paper is divided as follows: Section 2 spreads the notable related contributions from both conventional and nature-inspired reactive protocols in their two types: hop-by-hop and source routing. This section clarifies the impact of bio-inspired optimization algorithms on reactive routing. Section 3 introduces the BOA. Section 4 opens out onto the suggested solution's modules. Section 5 spreads in details BVRRP's modules describing the exploited qualities of BOA metaheuristic. Finally, Section 6 regroups the conclusion and a perspective for further contributing modifications. The configuration settings adopted for the solution's codification and simulation phase are also cited in this section.

II. RELATED WORK

This section spreads some notable conventional and bio-inspired protocols in the studied context, based mostly on VANET-destined AODV-based and DSR-based protocols.

A. Conventional reactive topology-based protocols

[6] discussed numerous enhanced versions of AODV for VANETs, notably:

Improved-AODV [7] seeks route stability and less network overheads by setting two steps for route discovery and route selection phases respectively. In the first, Route Requests (RREQs) are broadcasted to the more stable nodes, i.e., the vehicles merging toward the same direction and having clause speed. In fact, more stable vehicles engender more stable route links. In the second, two selection strategies are adopted for multiple source-destination trajectories determining more

 ISBN: 978-1-61208-720-7

stable routing paths: the first deducts stable routes by calculating the Route Expiration Time (RET), while the second calculates the route's total weight. RET represents the link having the minimum Link Expiration Time (LET) between all route links. LET is calculated using link hops' velocity, direction and communication range. Total route weight is calculated by adding up all links' weights, where each link's weight calculated by considering the velocity and direction of each link's upstream and downstream hops. It is worth noting that a predefined maximum threshold of selected neighbors following their link weight must not be surpassed so as bandwidth consumption is reduced. Finally, routes having a minimized weight and longer RET are more likely to be chosen for packet forwarding.

AODV-VANET [8] modifies both Route Request (RREQ) and Route Reply (RREP) of default AODV where each node's velocity, acceleration and direction values are added to RREQ header for route weight calculation. The latter is initialized by the source node. Route weight is the sum of all link weights passing by all hops liaising the source and destination nodes, where each current hop uses its predecessor's information to quantify the link chaining them. For RREP, the accumulated built route weight is added to its header plus the route itself and sent back straight to the source. The latter chooses among numerous received Route Replies (RREPs) the route having the minimum weight which indicates the more reliable one.

AODV-BD [9] keeps request broadcasting and data packets forwarding while route repair is being proceeded, seeking by that the dissemination of data packets to their destinations in reduced delays through fresher routes. That is, instead of launching route rediscovery, which wastes more time, data packets are rebroadcasted simultaneously with RREQs to neighboring nodes, and the one having a path to the destination takes charge of forwarding data packets to their final hop. RREP in this case is sent back to the source node when its related routed data packets reach the destination. This policy allows to shorten routing delays the fact considering the elimination of the required average time that necessitates for the reestablishment of new routes.

Prior-AODV (P-AODV) [10] seeks a higher route profitability by establishing a limited number of routes having fewer hops and fewer broken links. Thus, P-AODV restricts the number of RREQs and neighbors. In the first, for each node except the source, the number of distributed RREQs is limited to a predefined packet number threshold so that not all node's neighbors are notified. For the second, the number of neighbors is reduced, based on their distance from the current hop. These two restrictions seek next-hops that reduce the probability of link breaks. According to P-AODV, any node's surrounding neighbors are classified into two categories by considering the distance separating between the node and every neighbor: prior neighbors, which concern generally 2-hop nodes and more away, to which RREQs are sent, and overheads neighbors, which are generally 1-hop away from the current hop, whose RREQs are sent in second priority after prior zone if the aforementioned threshold allows. That is because overhead nodes share common neighbors with current hop in the prior zone. As a result, routing overheads are generally decreased.

Concerning source routing, numerous conventional protocols exist like DSR, RBVT-R [11] among others. Several DSR-modified versions have been realized, initially for MANETs and VANETs afterwards. Few cases are exposed below:

Enhanced-DSR (E-DSR) [12] adds two modifications to the default DSR which consists of: replacing the request broadcasting by multicasting to cut down overheads, and adopting a route hop limit to shorten long paths. Indeed, each node avoids distributing RREQ copies among the neighbors which are included in the in-progress request path or those sending RREQ copies to the current hop. This method helps to reduce the number of RREPs and that of control packet overheads consequently. To cope with DSR limits toward large networks, E-DSR sets a Time-To-Live (TTL) hops variable for destinations that are distant more than a customized TTL threshold from the related source node. This TTL variable is reinitialized by each node preceding the destination on expiration. Such node truncates the past hops, saves it in its route cache, reset the TTL variable, and forward the remaining path's hops until meeting the destination. This mechanism allows to alleviate data packet header and splitting data dissemination towards distant destinations into short-path forwarding sequences, hence gathering more reliable routing.

TE-DSR [13] selects route paths based on their trust evaluation, since each node saves a trust value for each of its neighbors. The degree of interactivity between these neighbors determines how these values are updated. This trust mechanism is composed of three modules. The first module includes three parts: a trust unit that involves an initializer which assigns low trust values for new unknown nodes, an upgrader for updating trust values depending on the previous experience values, and an administrator for interfacing between the previous two components and DSR protocol by storing trust information during runtime. The second is a router module which uses trust values of inter-node relationships for route evaluation and selection. The third is a monitor module which adjusts trust values for either received or missing acknowledgments. Any route discovery's received RREPs are sorted on the basis of this trust system.

For the proposed solution's context, numerous modified versions of VANET-destined DSR-originated protocols have been proposed, targeting more secured and rapid forwarding such as:

Modified DSR (M-DSR) [14] integrates a breakage prediction mechanism to the original DSR, based on tracking the Received Signal Strength Indicator (RSSI). Each node that anticipates a potential link break, when detecting the decreasing tendency of RSSI, engenders a Soon Link Breakage Warning (SLBW) packet, an equivalent of the Route Error (RERR) in default DSR, to unicast it to the source node. The latter marks with Breakage Prediction (RBP) the route having breakable link(s) with low RSSI state. Then, the source checks for another available route for data forwarding, i.e., the one that does not match with the RBP-marked routes. If it is not the case, this source node triggers a route discovery

 ISBN: 978-1-61208-720-7

by broadcasting a Modified RREQ (MRREQ) whose header is split into RREQ header and source route header. The latter stores the RBP-marked route hops' IDs to prevent from building new paths with potential link breaks.

B. Bio-inspired reactive routing related protocols

Earlier, DSR and AODV have been ameliorated using bio-inspired metaheuristics for VANETs. Few notable works in this field are discussed below:

AODVCS [15] a Cuckoo Search (CSA)-inspired protocol for VANETs taking bases from AODV. AODVCS exploits the aggressive egg reproduction behavior of Cuckoo particles to formulate an optimized source-destination routing path among a number of established routes. This is done by the implementation the CSA [16] in optimizing the route distance toward the destination by involving levy flights, a random walk of a given cuckoo which generates new solutions, i.e., finds better nests with superior reproduction. This mechanism is applied in each node to find its next-hop which concludes to form shorten routes in term of hop count. Each RREQ is modified by adding an additional field recording the past IDs between the source and destination to use it for fitness calculation and taking hop count as the sole fitness factor. Also, an adjacency matrix is set to record all links between source and destination. This matrix is used to repair the invalid solutions that can be engendered in the Lévy flight calculation.

FA-AODV [17] a Firefly Algorithm (FA) [18] routing protocol consisting of an FA-assisted partial route selection on AODV. FA-AODV estimates the stochastic brightness of nodes for next-hop selection, where the node having more brightness, which reflects in its calculated reachability degree, is prioritized to be chosen as FA routing packet's next-hop. FA routing packets act better than RREQs and RREPs since fireflies require less messages for forming paths. As a consequence, less overheads are engendered.

Several researchers provided different nature-inspired reactive routing algorithms for VANETs. Few notable solutions are discussed below:

G-NET [19] sets a DSR-based route discovery and a GA-inspired route optimization and maintenance. G-NET mobilizes two control packets for route discovery phase: G-NET_REQUEST and G-NET_RESPONSE. GA is executed on the destination node to optimize, by genetic recombination, the collected set of chromosome request paths which forms the GA' initial population. Each chromosome is codified in a set of genomes representing route hops. The fitness formula, which evaluates each population's solution quality, considers the route latency as sole evaluation parameter since it shows the adaptation level of individuals to the vehicular environment. G-NET performs classic GA tournament selection phase, one-point crossover and mutation operators for routing optimization. The route repair procedure follows after GA mutation by eliminating the reappearing genomes within the same route so that route loops are avoided. Finally, the stopping condition is fixed to four generations to stop running the GA.

Genetic AODV (G-AODV) [20] a secure backup GA-enhanced reactive hop-by-hop topology-based routing protocol that improves participating nodes' reliability. G-AODV seeks reducing RREQs using the GA. To do so, G-AODV performs an AODV-like route discovery where the source node avoids broadcasting RREQs to all its neighbors to avoid bandwidth wastage. Instead, it utilizes the GA to discover three routes for any solicited destination. It is worth noting that G-AODV applies a terminating condition for each RREQ using three user-defined evaluation thresholds (p1, p2 and p3) that any candidate solution must surpass to stop running the GA. Finally, formed routes tend to have a reasonable cost in term of forwarding time.

III. The Bull Optimization Algorithm (BOA)

Oguz Findik suggested the BOA, an enhanced version of the GA for continuous optimization problems [5]. BOA is a GA-based metaheuristic that implements the genetic crossover and mutation operators. BOA seeks surpassing GA's research depth limitations which reside in two parts, in this case: the decreasing stochastic search abilities when selecting the best individuals since early stages, and the quick convergence to local optima caused by the random selection of partial solutions having good fitness. Thus, BOA eliminates the selection phase which is usually behind the aforementioned weaknesses. Also, BOA increases the mutation percentage which imposes an influent randomization impact so that the convergence to local minima induced by the GA's low mutation rate is avoided. BOA keeps use of initial population during all process to exploit worse solutions to generate better global optima. To summarize, BOA is expected to offer less computation complexity than GA. The BOA cycle is illustrated in Figure.1.

IV. Our suggestion: BOA-assisted Vehicular Reactive Routing Protocol (BVRRP)

Numerous conventional enhanced protocols of AODV and DSR gave effective ideas on optimizing beaconing process, shortening the route discovery-repair sequence, alleviating request packet headers and so on. This helps to reduce forwarding delays and overheads and raise up packet delivery ratio. Meanwhile, it is noticed that reactive topology-based routing suffers from difficulties to maintain longer routing paths due to topology changes and high routing overheads generated by the beaconing. Also, such protocols are solution-orientated, since each protocol tries to solve a particular problem. For instance, the case of P-AODV which focuses on request broadcasting optimization or the case of M-DSR which concentrates on route break prediction optimization.

BVRRP protocol is destined for source routing assistance. Its purpose is to use the collected routes from the route discovery as an initial population for proceeding a BOA-based stochastic recombination of request paths. This protocol combines few foundations suggested in few DSR-modified and hop-by-hop routing algorithms with route optimization inspired from G-NET.

The strategy of BVRRP is resumed in the principles below:

- BVRRP is a source routing protocols, the case of multi-path DSR [21] and DSR-based protocols.
- BVRRP builds multiple link-disjoint routes for the same route demand, where routes must share one or numerous nodes, which is mandatory for BOA's genetic operators.

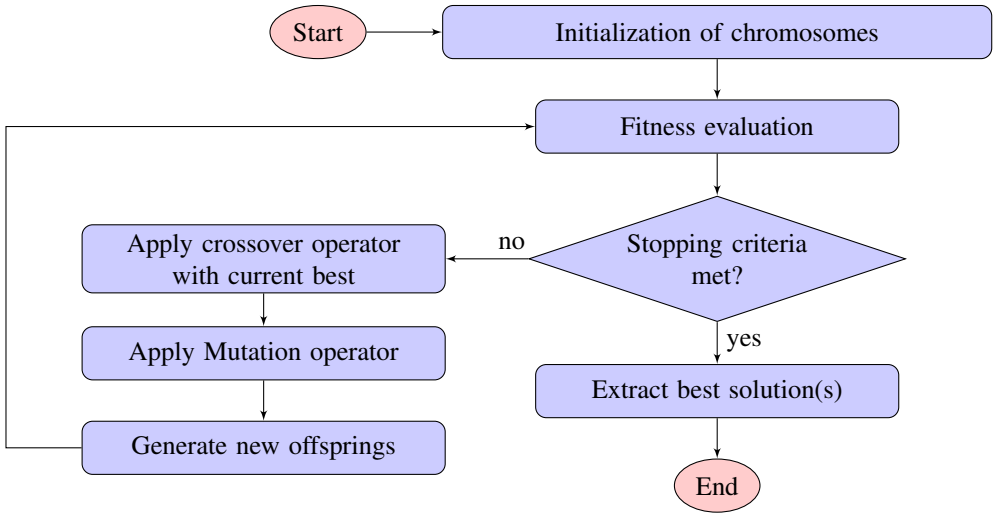

Figure 1. BOA lifecycle.

- The route discovery in BVRRP is based on a default request broadcasting as performed in AOMDV [22] and multipath DSR except its duration is adapted seen the amount of collected requests.
- BVRRP adopts a restricted beaconing for refreshing both neighborhood and topology links information which is required for BOA application and route break prediction mechanism.
- BVRRP implements two fitness optimization formulae: the first considers the parameters that favors limited lifetime routes providing quick routing, whereas the second offers longer-living routes to fit with recovery routing conditions constrained with considerable structural topology changes.
- BOA's first fitness formula, which is conceived for ordinary forwarding, involves three parameters for its calculation, in this case: the request path's hop count, average neighborhood longevity of request path's hops and request delay. The generated routes prioritize short route length which entails a quicker data forwarding and avoids routing circumstances under unmanageable mobility changes. Thus, the probability of switching to route recovery mode is reduced.
- The second fitness formula is conceived for route break recovery and implies three parameters: request delay, average number of route breaks of request path's hops and average RSSI of request path's hops. The generated routes are tendentious to have a longer longevity rather than a shorter path length so that more route reliability is gathered in order to adapt to route recovery forwarding conditions.
- BVRRP shortens BOA runtime on the destination by advancing the stopping criteria, when the number of collected routes is under a predefined route number threshold, since the initial population is not filled enough to extend for a deeper optimization reach.
- Furthermore, BVRRP adjusts BOA runtime depending on a predefined average route hop count threshold. An extension of stopping criteria is set for longer routes while a restriction is reserved for shorter ones. This serves a better research profitability of BOA.
- Recovery forwarding: BVRRP runs route repair phase with data packet forwarding in a parallel paradigm when the regular forwarding fails. Thus, the best available BOA-generated path is used for recovery forwarding while RERR packet is sent back to the source by backward path for route break notification and routing cache updates.
- BVRRP sets a link break prediction mechanism performed on each BOA-built route notifying all its hops and their limited neighborhood area about the state indicators of potential breakable links. This is done by checking up-to-date RSSI values collected by hello packets on each hop.

V. Contribution Details

BVRRP is an intelligent reactive routing protocol implementing the BOA for route optimization, parallel route repair with recovery data fowrding, and route break prediction system as detail in this section, where the main modules for this protocol are described detailing the major set of modifications, including the positioning of BOA in the routing proceedings:

A. Route discovery phase:

BVRRP's route discovery is an approach founded from the general principle of multipath routing, implemented in AOMDV and multipath DSR, based on collecting numerous routes for the same destination using source route discovery. Such routes can share one or numerous hops to cope with the requirements of BOA's genetic operators. Whereas, paths are link-disjoint since sharing route links degrades the stochastic impact of crossover and mutation operators to produce better paths. Every RREQ saves all passed nodes to form a request path. All collected paths from the received RREQs in the destination node will constitute the BOA's initial population. The routing discovery duration is limited by a Request Time-To-Live (RTTL) which marks how many RREQs are allowed to be received by the destination. The RTTL can be:

- Number of RREQs threshold (Nbr_{RREQ}): marks the maximum allowed number of received RREQs, i.e., represents the size of BOA's initial population.
- A discovery duration threshold ($Delay_{RD}$): this factor set an expiration time for the route discovery and it is enabled in case of the number of received RREQs is still under the predefined Nbr_{RREQ}.

BVRRP-conceived RREQ adds further fields to gather information required for checking link-disjointless property, evaluating route links quality, and regrouping necessary information for BOA's fitness evaluation mentioned in the previous section. Each RREQ saves, for each traversed hop, an entry including its average neighborhood longevity and its number of downstream link breaks. A request next-hop list is recorded as well containing each recorded node with its next hop. That will serve for BOA's mutation operator. Whereas, each RREP records one BOA-generated routing path. BVRRP's RREQ and RREP packet structure are illustrated in Table. I and Table. II, respectively.

Each hop receiving a RREQ performs the below-mentioned control operations to decide either accepting or rejecting it:

- If the current hop's ID is already recorded in the RREQ header's in-progress request route then this RREQ is discarded, since a loop over current hop is detected.
- Else, if the pair $\{$source ID, RREQ ID$\}$ is already passed by the current hop, then the received RREQ has been already treated, so this RREQ is discarded as a result.
- Else, if any of the links recorded in the received RREQ have been already frequented from an earlier RREQ by the current node for the same destination, then the received RREQ is discarded, since the route link-disjointless property is not guaranteed in this case.
- Else, if the current hop's ID is the destination's ID then if the route discovery time limit is not expired yet, by checking both Nbr_{RREQ} and $Delay_{RREQ}$ thresholds, then the build request path is added BOA's initial population, otherwise the destination rejects all new arriving RREQs for the corresponding source-destination route discovery and the BOA routing optimization is started.
- Otherwise, the received RREQ is the first copy of its source-destination entry, so it's ID is appended to the in-progress request route if its route hop count is under the predefined hop count limit (HTTL) threshold which marks the maximum allowed length for request chromosomes of BOA's initial population.

Furthermore, each valid received RREQ (accepted request) gets to add the following information when hosted by each intermediary node:

- Updating the average RSSI of all passed hops up to the host hop.
- Updating the average number of route link breaks of all passed hops including the current hop.
- Updating the average request path's neighborhood lifespan with the current hop's average neighbors' lifetime.
- Recording the last and current hop in the request next-hop list. As mentioned, this serves performing the BOA's mutation operator.
- Recording the accumulated delay since the sent of the RREQ from source node.

B. Restricted Beaconing

A restricted beaconing is executed to serve the BOA application and route break estimation in which hello notification messages are sent periodically to 1-hop neighboring nodes for:

- Recording reachable 1-hop neighbors: each node adds a new node on its neighbors table and records the timestamp when it was added. This permits to quantify the lifespan of the liaison between the two hops which will serve for BOA main route fitness evaluation. In our opinion, it is estimated that every vehicle having a longer average connection with its neighbors is supposed to perform more reliable routing.
- Recording timestamp and value of hello-collected RSSI for each neighbor. This measure serves for route break prediction mechanism, offering more recent information about each route links's connectivity state.
- Updating the neighbor table and what entails as changes in routing cache and request neighborhood list. Such updates are mandatory to perform BOA's fitness calculation.

The periodic interval that separates two hello packets' broadcasting is dynamically adapted depending on the node's solicitation degree which refers to the its participation level to data forwarding in the network. This indicator is approached following the below factors in any instant *t*:

- Number of active neighbors ($Neigh_{Count}$).
- Number of active request entries ($Active_{RREQs}$).
- Unreachability degree, measured by the historic number of route link breaks as downstream hop ($RouteBrks_{Count}$).

Any node's reachability value ($Reach_{Level}$) is calculated using (1).

$$Reach_{Level} = (0.3 \times Neigh_{Count}) + (0.6 \times Active_{RREQs}) + (0.1 \times RouteBrks_{Count}) \quad (1)$$

A solicitation threshold ($Solicitation_{Thresh}$) is predefined to decide whether accelerating or slowing down the beaconing frequency, where any node having a higher reachability value than $Solicitation_{Thresh}$ obliges more hello updates to avoid quick data aging (2).

$$Reach_{Level} \leq Solicitation_{Thresh} \quad (2)$$

Table. III presents the structure of the proposed neighbor table set on each node.

C. Application of BOA

The BOA optimization is performed in the destination node straight after expiration of the RTTL in case no notification has been received on the source by any intermediate node that recognizes the destination. The collected routes constitute the initial population of BOA which passes a limited number of iterations to produce few optimized routes. This iteration limit is adjusted basing on two factors:

- *Average Hop Count of initial population's individuals (AvgHC)*: based on request chromosomes's length since longer routes offer more recombination alternatives. This

TABLE I. BVRRP'S RREQ PACKET STRUCTURE.

Packet type	Request ID (Seq number)	Source ID	Destination ID
Path hops IDs	Hop Count	Total delay	Neighborhood list
Avg Nbr Route breaks	Avg Recv signal strength	Next-hop list	Topology links list

TABLE II. BVRRP's RREP PACKET STRUCTURE.

Packet type	Source ID	Destination ID	Route type (Main/Recovery)	Route's IDs

TABLE III. NEIGHBOR LIST'S ENTRY FIELDS.

1-hop neighbor ID	Neighbor record time	Neighbor's hello RSSI	RSSI timestamp	Neighbor Delay

parameter decides either reducing or extending BOA lifetime, eliminating BOA for main route (restricting BOA application to recovery routes), or setting a shortened BOA for either main or recovery routes optimization. Thus, a predefined hop count threshold ($Avg_{Hopcount}$) is set to decide how many generations BOA performs for both main route and recovery procedures.

- *Minimum number of RREQ routes for initial population (MinNbr$_{RREQ}$)*: decides the stopping criteria of BOA for both main route and recovery routes optimization. If the number of collected RREQ routes is under $MinNbr_{RREQ}$ within an expired $Delay_{RREQ}$ then the number of BOA generations is shortened for both main and recovery optimization procedures.

The full BOA cycle performed in the destination node is detailed below:

- *Solution codification*: Each chromosome represents a candidate source-destination route, where the genomes represent route hops. It is worth noting that BOA chromosomes do not share the same dimension, since routes can have different hop count. Two hop count limits for main path (HC_{Main}) and recovery paths ($HC_{Recovery}$) are set respectively to avoid extended routes, while the BOA genetic operators, i.e., crossover and mutation are customized to avoid any dysfunction that may entail erroneous paths. It is worth noting that HC_{Main} is shorter than $HC_{Recovery}$ to meet the above-described routing constraints.
- *Initialization step*: The initial population regroups all established node-shared paths collected from received RREQs that liaise the source and destination nodes. Individuals differ in term of hop count; hence initial population's chromosomes have variable length which forces few adjustments in crossover operator. Noting that initial chromosomes can be longer than HC_{Main} and $HC_{Recovery}$ to open the possibility for generating better offspring paths after crossover and mutation recombination. It is worth to mention that BOA's initial population is kept during all performed BOA generations.
- *Fitness functions*: each initial chromosome is evaluated with two fitness equations used for main and recovery procedures of BOA algorithm respectively. The $Fitness_{routing\_path}$ calculated for the main BOA route fitness and $Fitness_{recovery\_paths}$ calculated for recovery BOA route fitness.
- *Crossover operator*: BOA's crossover operator intervenes to combine the best parts of numerous chromosomes by the permutation of route pair portions that share one or several nodes. For BVRRP, two-point crossover is performed in all BOA generations by associating the best chromosome with every member of the initial population including the partial-best chromosome. Meanwhile, since the crossover operator in BVRRP solution is variable due to the dependency from request paths' genomes where it differs from a chromosome pair to another, depending on the position of common permutation points, either one-point crossover is switched in case there is one common node between chromosomes pair or no crossover if there are no common genomes.
- *Mutation operator*: BOA's mutation operator intervenes for tracking global-varied solutions. It consists of replacing a randomly-selected genome of a candidate solution with a random hop provided that it belongs to the next-hop list of the replaced genome's predecessor. A low mutation probability (M_{prob}) is set for BVRRP solution (M_{prob}= 0.02) to avoid high variety impact which risks converging to lower-quality solutions. Noting that a two-point mutation is applied for BVRRP and performed on initial population's individuals including the partial-best chromosome.
- *Stopping criteria*: two stopping criteria are set for BOA. The first is static and fixed to a limited number of iterations for both main and recovery procedures. In our case, it varies between 03 and 06 generations depending on the above-mentioned iteration limit factors, i.e., adapting to both $MinNbr_{RREQ}$ and AvgHC thresholds. The second is dynamic and considers a predefined BOA-progressing factor of the global best, set at an improvement of 60% of the initial best chromosome, that must be reached to stop the BOA for both the main path's procedure and the average fitness of best five (05) solutions for the recovery paths' procedure.
- *Solutions' extraction*: the best found path from main route algorithm, i.e., having the lowest fitness value, is selected for regular forwarding, while the best five (05) solutions from recovery routes algorithm, which have the five lowest fitness values ($Min\{Fitness_{recovery\_paths}\}$), are preserved for recovering any possible route break.

The fitness function of a given route **R** to a destination **D** used for post route discovery forwarding is given in (3):

$$Fitness_{routing\_path} = (p_{\alpha 1} \times path_{delay}) + (p_{\alpha 2} \times path_{hopcount}) - (p_{\alpha 3} \times path_{rss}) \quad (3)$$

Where:

- $p_{\alpha 1}$, $p_{\alpha 2}$ and $p_{\alpha 3}$ are user defined parameters which are set in a way that favors quick forwarding over route reliability.
- $path_{delay}$ is the total delay consumed between the source and destination nodes.
- $path_{hopcount}$ is the number of chromosomes's passed nodes (hops).
- $path_{rss}$ is the average RSSI of received RREQs of all path's hops.

The main fitness parameters are set as: $p_{\alpha 1}$= 0.45, $p_{\alpha 2}$= 0.3 and $p_{\alpha 3}$= 0.25.

The fitness function used for route recovery forwarding is given in (4):

$$Fitness_{recovery\_paths} = (p_{\beta 1} \times path_{delay}) + (p_{\beta 2} \times path_{nrb}) - (p_{\beta 3} \times path_{neighborhood}) - (p_{\beta 4} \times path_{rss}) \quad (4)$$

Where:

- $p_{\beta 1}$, $p_{\beta 2}$, $p_{\beta 3}$ and $p_{\beta 4}$ are user defined parameters whose values advantage extending route longevity with keeping reasonable routing delays.
- $path_{nrb}$ is the average number of route breaks that includes the average route breaks number of chromosome hops.
- $path_{neighborhood}$ is the average neighborhood longevity of all route nodes' neighbors.

The recovery fitness parameters are set as: $p_{\beta 1}$= 0.15, $p_{\beta 2}$= 0.3, $p_{\beta 3}$= 0.3 and $p_{\beta 4}$= 0.25.

A RREP is created to record each BOA-optimized path in its header including the path links and extracted partial routes'with its fitness values, and then returned to the source node. This serves keeping the best routes and provisioning intermediate nodes having buffered data packets for the same destination. The destination node empties its structures related to the BOA optimization process for the corresponding source-destination entry in the next-hop table, Request route neighborhood table, Request topology links table and BOA table.

Each intermediary hop from the BOA-built route which receives a RREP performs the following operations:

- Updating the routing table with extracted both main and recovery RREP's sub-paths that liaise the current hop to the destination.
- Checking if there are buffered data packets for the destination and the node preceding the destination. If so, these data packets are forwarded using the main routing path and then, route discovery is avoided.
- Discharging the routing data structures related to the concluded route discovery of the source-destination entry, in this case the seen links list and the seen requests list.

The full BOA's pseudo-code implemented for route optimization is presented in Algorithm. 1.

D. Route maintenance and route break prediction

BVRRP anticipates with a route recovery strategy to face any unexpected route break and keeps data packet forwarding while performing route repair tasks. BVRRP's route maintenance phase is composed from two major mechanisms:

- *Parallel route repair and recovery forwarding system*: this module is triggered when a route break event is stated. It performs simultaneously a route repair process while keeping data forwarding. Initially, a RERR copy is generated from the node detecting the link break generally the node preceding the broken link (upstream node), and then sent back to the source node. Each intermediary hop receiving this RERR performs the necessary updates to the destination entry of its routing cache. Second, the downstream node of broken link keeps forwarding both blocked and buffered data packets using its available recovery path. In the meantime, a new route discovery is triggered by the source after being notified by the RERR.
- *Route break prediction system*: this module is the post-BOA phase, when a prediction packet (PRED) is sent back to the source just after the RREP by the reverse path. In a first step, PRED notifies built-BOA path's hops of any changes in RSSI state to detect breakable route links of this path, so that every hop that hosts this PRED updates breakable link list and marks all routes in routing cache having breakable links to restrict their lifespan, as a result, route breaks are avoided. Also, the stored breakable links have a limited lifetime and expire once the established BOA path is broken or expired. Second measure, each passed hop belonging to the built-BOA route notifies, with a PRED copy, the nodes that belongs to its three-hop radius area. This measure is set to reduce the impact of breakable links in close network area of established route and anticipate recovery forwarding which shortens routing delays. A route break equation is defined to estimate route links' lifetime as coded in the route break estimation pseudo-code presented in Algorithm. 2.

VI. Conclusion and Future Work

In this paper, an EA-assisted V2V reactive routing protocol for VANETs is suggested named the BOA-based Vehicular Reactive Routing Protocol (BVRRP) that makes use of BOA metaheuristic to generate better genetically-optimized routes for both regular and recovery data forwarding. It is noticed through this proposition the possibility of spreading EAs family in route optimization for vehicular networks and its effective implementation with other modifications that touches the main functional parts of a typical ad-hoc reactive routing protocol such as route maintenance and route break prediction. This work is opened for other modifications such the implementation of nature-inspired metaphor-based algorithms, the passage to Vehicle-To-Infrastructure (V2I) routing through cloud computing technologies, or the combination with geography-assisted routing.

 ISBN: 978-1-61208-720-7

Algorithm 1 BOA pseudo-code of BVRRP

Define Src: source node;
Define $Dest$: destination node;
Define constant $main\_threshold$: loop limit for main route optimization;
Define constant $rec\_threshold$: loop limit for recovery routes optimization;
Define constant $main\_optimum$: minimum routing path fitness value for post route discovery forwarding;
Define constant $rec\_optimum$: minimum routing path fitness value for post route break forwarding;
Define constant $cross\_prob$: crossover probability;
Define constant mut_prob: mutation probability;
Declare $cross\_offspring$: post-crossover offspring variable;
Declare $mut\_offspring$: post-mutation offspring variable;
Declare $fittest\_recovery\_list$: vector for best 05 recovery paths;
Declare $current\_mbest$: current best main path variable;
▷ intialize populations and global bests
Set $main\_population$ = Initialize ($initial\_population$); ▷ a copy of initial population for main algorithm
Set $recovery\_population$ = Initialize ($initial\_population$); ▷ a copy of initial population for recovery algorithm
Set $fittest\_main$ = Best_Fitness ($main\_population$); ▷ get best initial chromosome for main population
Set $fittest\_recovery$ = Best_Fitness ($recovery\_population$); ▷ get best initial chromosome for recovery population
Set $main\_iteration$ = 1, $rec\_iteration$ = 1;
while $main\_iteration < main\_threshold$ and $fittest\_main > main\_optimum$ **do** ▷ perform BOA for regular forwarding
    $current\_mbest$ = extract_best ($main\_population$);
    for each individual k in $main\_population$ **do**
        $cross\_offspring$ = Crossover_operator ($k, cross\_prob$);
        $fittest\_main$ = Fitness ($cross\_offspring$);
        if $fittest\_main <$ Fitness ($current\_mbest$) **then**
            Update_best ($fittest\_main$);
        $mut\_offspring$ = Mutation_operator (k, mut_prob);
        $fittest\_main$ = Fitness ($mut\_offspring$);
        if $fittest\_main <$ Fitness ($current\_mbest$) **then**
            Update_best ($fittest\_main$);
    $main\_iteration$++;
if $main\_iteration \geq main\_threshold$ or $fittest\_main \leq main\_optimum$ **then** ▷ select best BOA main path
    Set $best\_main\_route$ = $current\_mbest$;
    Load_RREP ($best\_main\_route$); ▷ load new path in a RREP
    Unicast_RREP (Src, $best\_main\_route$); ▷ send RREP to the source node
    Route_break_prediction (Src, $Dest$, $best\_main\_route$); ▷ trigger route breaks prediction for sent path
while $rec\_iteration < rec\_threshold$ and $fittest\_recovery > rec\_optimum$) **do** ▷ perform BOA for recovery forwarding
    Set $current\_rbest$ = extract_best ($recovery\_population$);
    for each individual j in $recovery\_population$ **do**
        $cross\_offspring$ = crossover_recovery_operator (j, $cross\_prob$);
        $fittest\_recovery$ = Fitness ($cross\_offspring$);
        if $fittest\_recovery <$ Fitness ($current\_rbest$) **then**
            Update_bests ($fittest\_recovery\_list$, $cross\_offspring$); ▷ update best recovery routes list after crossover
        $mut\_offspring$ = mutation_recovery_operator (j, mut_prob);
        $fittest\_recovery$ = Fitness ($mut\_offspring$);
        if $fittest\_recovery <$ Fitness ($current\_rbest$) **then**
            Update_bests ($fittest\_recovery\_list$, $mut\_offspring$); ▷ update best recovery routes list after mutation
    $rec\_iteration$++;
if $rec\_iteration \geq rec\_threshold$ or $fittest\_recovery \leq rec\_optimum$) **then** ▷ extract best 05 BOA recovery paths
    Set $recovery\_routes\_list$ = Extract ($fittest\_recovery\_list$); ▷ order best 05 recovery routes
    for each recovery_path rp in $recovery\_routes\_list$ **do**
        Load_RREP (rp); ▷ load new path in a RREP
        Unicast_RREP (Src, rp); ▷ send RREP to source node
        Route_break_prediction (Src, $Dest$, rp); ▷ trigger route breaks prediction for sent path

 ISBN: 978-1-61208-720-7

Algorithm 2 BVRRP break prediction procedure of a post-BOA route for Current hop N

Define const $HSignThresh$: packet RSSI threshold;
Define const Exp_param: expiration parameter $\in$ [0,1];
Define const Ext_param: extension parameter $\in$ [0,1]; with $Ext\_param > Exp\_param$;
Define Boa_route: routing path loaded in the PRED;
Define Src: source node;
Define $Dest$: destination node;
for each passed hop $S \in$ post-BOA path positioned after N **do**
    Record HT: last received hello signal timestamp of the link [S-1, S]; ▷ (S-1) is the previous hop of S
    Record PT: current received prediction signal timestamp of [S-1, S];
    Record HSs: last received hello signal strength of [S-1, S];
    Record PSs: current received prediction signal strength of [S-1, S];
    if $HSs > PSs$ **then** ▷ (S-1) is less reachable by S
        if $PSs < HSignThresh$ **then** ▷ (S-1) is out of transmission range of S
            Remove_link ([S-1, S], $topology\_link\_list$); ▷ delete [S-1, S] from the topology links list
            Remove_Routes ([S-1, S], $route\_cache$); ▷ delete all paths having [S-1, S]
        else
            Set $time\_exp = (PT - HT) \times (HSs - PSs) \times Exp\_param$;
            Expire_link ([S-1, S], $time\_exp$); ▷ [S-1, S] link will expire to $time\_exp$
            Update_Routes ($route\_cache$, [S-1, S], $time\_exp$); ▷ shorten paths lifespan having [S-1, S] to $time\_exp$
    else ▷ [S-1, S] is more reliable than when on path creation timestamp
        Set $time\_ext = (PT - HT) \times (PSs - HSs) \times Ext\_param$;
        Extend_link ([S-1, S], $time\_ext$); ▷ [S-1, S] lifespan is extended to $time_ext$
        Update_Routes ($route\_cache$, [S-1, S], $time\_ext$); ▷ extend paths lifespan having [S-1, S] to $time\_ext$
    Notify_Neighborhood (); ▷ notify 3-hop neighbors of evaluated path
    if breakable_link [S-1, S] and $buffer\_cache$ not empty **then** ▷ start new route discovery for $Dest$
        Route_Discovery ($Dest$);
    if (S-1) is Src **then** ▷ end route prediction process
        Stop_Process ();

It is worth noting that the BVRRP is programmed using the GloMoSim simulator in C++ and set for comparison with P-AODV and G-NET on three QoS metrics, namely: the Average End-To-End Delay (AE2ED), Packet Delivery Ratio (PDR) and the Normalizd Routing Load (NRL). The BVRRP version of *"VEHICULAR 2019"* conference is reduced to only theoretical solution for conference oral lecture while the simulation results will be attached to the BVRRP manuscript for proceeding indexing.

REFERENCES

[1] P. K. Pagadala and N. M. S. Kumar, "A survey on Topology based Reactive Routing Protocols in Vanets," *Global Journal of Computer Science and Technology: ENetwork, Web & Security*, vol. 18, no. 4, Dec. 2018. [Online]. Available: https://computerresearch.org/index.php/computer/article/view/1765

[2] K. Z. Ghafoor, J. Lloret, K. A. Bakar, A. S. Sadiq, and S. A. B. Mussa, "Beaconing Approaches in Vehicular Ad Hoc Networks: A Survey," *Wireless Personal Communications*, vol. 73, no. 3, pp. 885–912, Dec. 2013.

[3] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," University of California, Tech. Rep. DEC-TR-506, Aug. 2003.

[4] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*. Springer, 1996, pp. 153–181.

[5] O. Findik, "Bull optimization algorithm based on genetic operators for continuous optimization problems," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 23, pp. 2225–2239, Sep. 2013.

[6] A. Joshi, P. Sirola, and K. C. Purohit, "Comparative Study of Enhanced AODV Routing Protocols in VANET," *International Journal of Computer Applications*, vol. 96, no. 18, pp. 22–27, 2014.

[7] B. Ding, Z. Chen, Y. Wang, and H. Yu, "An Improved AODV Routing Protocol for VANETs," in *IEEE 2011 International Conference on Wireless Communications and Signal Processing (WCSP)*, Nanjing, China, Nov. 2011.

[8] H. Guo, W. C. Wong, F. B. A. Thani, and Y. Wu, "An Optimized Routing Protocol for Vehicular Ad Hoc Networks," in *TENCON 2010 - 2010 IEEE Region 10 Conference*, Fukuoka, Japan, Nov. 2010.

[9] B. Li, Y. Liu, and G. Chu, "Improved AODV Routing Protocol for Vehicular Ad hoc Network," in 3^{rd} *International Conference on Advanced Computer Theor and Engineering(CTE)*, Chengdu, China, Aug. 2010.

[10] O. Abedi, R. Barangi, and M. A. Azgomi, "Improving route stability and overhead of the AODV routing protocol and makeing it usable for VANETs," in 29^{th} *IEEE International Conference on Distributed Computing Systems Workshops*, Montreal, QC, Canada, Jun. 2009.

[11] J. Nzouonta, N. Rajgure, G. Wang, and C. Borcea, "Vanet Routing on City Roads Using Real-Time Vehicular Taffic Information," *IEEE Transactions On Vehicular Technology*, vol. 58, no. 7, pp. 3609–3626, Sep. 2009.

[12] S. Sultana, S. Begum, N. Tara, and A. R. Chowdhury, "Enhanced-DSR: A New Approach to Improve Performance of DSR," *International Journal of Computer Science and Information Technology*, vol. 2, no. 2, pp. 113–123, Apr. 2010.

[13] N. Bhalaji, A. R. Sivaramkrishnan, S. Banerjee, V. Sundar, and A. Shanmugam, "Trust Enhanced Dynamic Source Routing Protocol for Adhoc Networks," *World Academy of Science, Engineering and Technology*, vol. 49, pp. 1074–1079, 2009.

[14] K. Zahedi, Y. Zahedi, and A. S. Ismail, "Enhancing the Performance of DSR Routing Protocol Using Link Breakage Prediction in Vehicular Ad Hoc Network," *International Journal of Computer Networks and Communications Security*, vol. 1, no. 1, pp. 7–14, Jun. 2013.

[15] A. Kout, S. Labed, S. Chikhi, and E. B. Bourennane, "AODVCS, a new bio-inspired routing protocol based on cuckoo search algorithm for mobile ad hoc networks," *Wireless Networks - Springer*, vol. 24, no. 7, pp. 2509–2519, Oct. 2018.

[16] X.-S. Yang and S. Deb, "Cuckoo Search via Lévy flights," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, India, Dec. 2009.

[17] M. B. B. Anbu Malar *et al.*, "Firefly Algorithm For Optical Path Detection In Ad Hoc On-Demand Distance Vector (AODV)," *International Journal of Advance Research in Science and Engineering*, vol. 6, no. 11, pp. 2039–2047, Nov. 2017.

[18] X.-S. Yang, "Firefly Algorithm, Lévy Flights and Global Optimization," in *Research and Development in Intelligent Systems XXVI*, Oct. 2009, pp. 209–218.

[19] E. C. G. Wille, H. I. D. Monego, B. V. Coutinho, and G. G. Basilio, "Routing Protocols for VANETs: An Approach based on Genetic Algorithms," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 2, pp. 542–558, Feb. 2016.

[20] C. Garg and B. Wadhwa, "G-AODV: A Novel Approach to Improve AODV by Using Genetic Algorithm in VANET," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 6, Jun. 2016.

[21] A. Boursier, S. Dahlen, J. Marie-Francoise, T. S. Marin, and S. Nethi, “Multipath DSR protocol for ad hoc network,” Aalborg University, Institute of Electronic Systems, Tech. Rep., Dec. 2004.
[22] M. K. Marina and S. R. Das, “Ad hoc on-demand multipath distance vector routing,” *Wireless Communications and Mobile Computing*, vol. 6, pp. 969–988, 2006.