IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 September 2023, accepted 3 October 2023, date of publication 9 October 2023, date of current version 25 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3322691

RESEARCH ARTICLE

Bio-Inspired Multi-Objective Algorithms Applied on the Optimization of the AODV's Routing Recovery Mechanism

CLODOMIR SANTANA[1,2], (Member, IEEE), MARIANA MACEDO[3], EMILLY ALVES[4], MARCIO T. GUERREIRO[5], HUGO VALADARES SIQUEIRA[5], ANURADHA GOKHALE[6], (Senior Member, IEEE), AND CARMELO J. A. BASTOS-FILHO[4], (Senior Member, IEEE)

[1]Department of Computer Science, University of Exeter, EX4 4PY Exeter, England, U.K.
[2]Tadeusz Manteuffel Institute of History, Polish Academy of Sciences, 01-224 Warsaw, Poland
[3]Center of Collective Learning, ANITI, University of Toulouse, 31015 Occitania, France
[4]Department of Computer Engineering, University of Pernambuco, Recife 50100-010, Brazil
[5]Graduate Program in Industrial Engineering, Federal University of Technology-Paraná, Ponta Grossa 84017-220, Brazil
[6]College of Applied Science and Technology, Illinois State University, Normal, IL 61761, USA

Corresponding author: Hugo Valadares Siqueira (hugosiqueira@utfpr.edu.br)

This work was supported in part by the Polytechnic School of the University of Pernambuco (POLI-UPE) and Brazilian Agencies Coordination for the Improvement of Higher Education Personnel (CAPES)—Financing Code 001, in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant 315298/2020-0, and in part by the Araucaria Foundation under Grant 51497. The work of Mariana Macedo was supported by the Artificial and Natural Intelligence Institute, University of Toulouse, under Grant ANR-19-PI3A-0004.

ABSTRACT Advances in electronic systems, wireless communication protocols, and intelligent devices allowed the development of networks of mobile devices such as cars, drones, and robots. The field of mobile ad hoc networks (MANETs) comprises networks where the mobility of the devices is one of the fundamental elements that characterise these networks. However, the node's mobility leads to constant changes in the network's topology, representing a challenge to routing protocols designed for MANETs. Although there is effort from researchers to tackle the intricacies of routing protocols in MANETs, there is still room for improvement as new applications with challenging specifications continue to arise. This research enriches the existing theoretical perspective by presenting an innovative method for optimising the routing performance of the ad hoc on-demand distance vector (AODV) protocol. Grounded on multi-objective metaheuristics, we aim to improve AODV's routing recovery performance concerning routing delay, energy consumption, packet loss ratio, and route load metrics. To gauge the quality of our contribution, we compare its performance to the standard AODV, a mono-objective optimised AODV, and four other well-known routing protocols with different routing approaches. The results indicate that the proposed solution was superior to the original AODV with average improvements of 56.0%, 59.3%, 48.1% and 0.7% on route load, routing delay, packet loss ratio and energy consumption, respectively. It also presented competitive results compared to other routing protocols.

INDEX TERMS AODV, mobile ad-hoc networks, multi-objective optimizations, route recovery.

I. INTRODUCTION

Mobile ad hoc networks (MANETs) are self-configuring wireless networks – that do not require a fixed or previously configured infrastructure – that allow devices (or nodes) to form connections in a dynamic manner. These networks have decentralised control and are composed of independent mobile nodes communicating in a multi-hop scheme [1]. Among many types of MANETs [2], [3], [4], [5], we can cite the flying ad hoc networks (FANETs) [6], [7]. FANETs

The associate editor coordinating the review of this manuscript and approving it for publication was Nurul I. Sarkar.

has attracted attention due to increased applications related to swarms of unnamed air vehicles (UAVs). Regardless of the type, all MANETs face challenges related to mobility, quality of service (QoS), energy, and routing [4], [5].

The mobility of MANET's nodes represents an extra challenge to [8] concerning the routing protocols. As the network topology changes continuously due to the mobility of the nodes, these routing protocols need mechanisms to minimise the impact caused by the mobility of the nodes on the routing process [9]. Furthermore, depending on the application, the nodes' movement does not consider the possible communication issues (e.g., breaking an active route), which adds more complexity to the routing process.

In an attempt to overcome the problems of proactive protocols, the reactive or on-demand protocols only maintain information from active routes. The route discovery works on demand [10]. When a node has to send data to a specific destination, it broadcasts request packets if it does not have an active route to that node [11].

Since no protocol has the best performance in all scenarios, the routing protocol selection depends on the characteristics of the application. Furthermore, as the node's energy in the network comes from limited energy sources such as commercial batteries, there is also an energy limitation [12], [13]. Since wireless communication is the primary consumer of MANETs (i.e., energy used to send and receive data/route packets) [14], an efficient routing protocol can reduce energy consumption in the network.

Among the routing protocols designed for MANETs, the ad hoc on-demand distance vector (AODV) is a reactive protocol commonly used in MANETs [15], [16], [17].

The ad hoc on-demand distance vector (AODV) is a simple flooding routing protocol. Due to these characteristics, it significantly produces excessive redundant traffic (i.e., broadcast storms), which overloads network resources such as bandwidth and battery, especially in high-density network environments, impacting network functionality and increasing packet loss, end-to-end delay, latency, and low throughput. [5].

Besides the advantages and drawbacks of reactive protocols, the AODV has positive and negative aspects. For example, Pereira et al. [18] analysed the AODV local and source repair mechanisms separately and observed that the source repair exhibits better performance in most scenarios when compared to the local repair. Moreover, a study published in 2012 showed that AODV's approach to selecting the route repair mechanism could be improved by adding the concept of node connectivity (i.e., the number of neighbours from a node) [19].

Although the on-demand behaviour indirectly helps AODV save energy by executing routing discovery and route repair operations only when needed, this protocol is not natively energy-aware nor has mechanisms designed to prevent excessive energy consumption. In fact, in 2009, a modified energy-aware version of AODV proposed to select routes that require less energy to deliver packets [20].

Another drawback commonly associated with the AODV is that this protocol needs to adapt its behaviour to meet the application's specifications [19]. Thus, AODV cannot be tuned to reduce energy consumption or maximise the packet delivery ratio. However, it is worth mentioning that this problem is not exclusive to AODV and is present in several other routing protocols for MANETs.

As adopted in previous works [19], [21], the introduction of node connectivity in the route repair process, coupled with a few more parameters, enhanced the performance of the AODV. These works employed a mono-objective algorithm to optimise the routing performance of the AODV regarding a single metric. This approach allows adapting the version of AODV by selecting different QoS metrics. However, as these works used mono-objective algorithms, they only optimise the protocols regarding a single metric (i.e., they cannot simultaneously minimise the routing delay and energy consumption).

This study contributes to enhancing the prevailing theoretical framework by introducing a novel technique to improve the routing efficiency of the AODV protocol. The primary goal is boosting AODV's routing recovery performance by tuning four parameters (i.e., SW1, SW2, LW1, and LW2) responsible for controlling which route recovery approach (e.g., local or source recovery) will be employed in a particular route breakage scenario. Our research further contributes to the existing body of knowledge by exploring diverse approaches for integrating the notion of node connectivity into the Ad Hoc On-Demand Distance Vector (AOVD) protocol. In contrast to earlier methods that extracted connectivity information from the simulation platform and treated it separately from the routing process, we investigate incorporating connectivity directly into the routing process. This investigation is of great significance as it unveils that various strategies for integrating connectivity into AODV can potentially deteriorate its performance.

To assess the routing performance of the proposed techniques against the original AODV and four other routing protocols, we utilise multiple QoS metrics [22], [23] such as normalised route delay (Delay), packet loss ratio (PLR), normalised route load (NRL), and energy consumption (EC). In this context, another contribution of our study revolves around evaluating Quality of Service (QoS) metrics that are apt for multi-objective optimisation. Our findings demonstrate a discernible correlation among specific metrics, implying that optimising one metric could indirectly lead to the optimisation of others. Among the chosen metrics, it's worth noting that only energy consumption did not exhibit a strong correlation with the rest.

Tackling the multi-objective optimisation of the four parameters, we select the non-dominated sorting genetic algorithm (NSGA-II), speed-constrained multi-objective particle swarm optimisation (SMPSO), and the strength Pareto evolutionary algorithm (SPEA2). We compare the proposed solution against alternative routing protocols and prior mono-objective methodologies. This comparative analysis enables us to discern variations in performance among these techniques, particularly in the context of multi-objective versus mono-objective strategies. Lastly, the optimal weight set derived from the optimisation process corroborates findings from earlier studies, underscoring that, across most scenarios, the source repair mechanism of AODV outperforms local repair mechanisms.

The remainder of this paper is organised as follows: Section II presents a brief classification on the main routing protocols; Section III presents a list of related work, Section IV explains the fundamental theoretical aspects behind this work, and Section VI describes the experimental setup and discusses the computational results. Next, Section VII discusses the main advantages and limitations of the proposed approach. Lastly, the conclusions are presented in Section VIII.

II. ROUTING PROTOCOLS CLASSIFICATION

We can divide routing protocols for mobile ad hoc networks into three classes based on their modes of operation: proactive, reactive, and hybrid protocols [24]. Proactive protocols periodically monitor the network to detect changes (e.g., new routes and route breaks) and use tables to store routing information for all or a group of nodes in the network. This strategy's main advantage is reducing the time needed to create or repair routes. Furthermore, since the nodes have updated routing information, they can promptly reconstruct/create routing paths. However, the number of tables and the amount of stored data can increase the memory consumption in the nodes. Besides, periodic network monitoring may flood it with routing packets, reduce the bandwidth available for the traffic of data packets, and increase packet loss risk.

In an attempt to overcome the problems of proactive protocols, the reactive or on-demand protocols only maintain information from active routes. The route discovery works on demand [10]. When a node has to send data to a specific destination, it broadcasts request packets if it does not have an active route to that node [11].

If a node receives the route request and has a busy way to the requested destination, it sends a reply packet with the solicited path to the source node. Otherwise, it forwards the route request packet to its neighbours. This process continues until a route is found, the destination is reached, or a predefined timeout is exceeded.

At MANETs reactive protocols, the nodes are flexible, which leads to frequent route failures and route rediscovery necessity, falling into a trade-off where broadcasting increases the reachability of the route request messages to the destinations in sparse networks. Still, on the other hand, rebroadcasting causes excessive redundant packets across high-density networks that significantly decrease network performance [5].

Compared to proactive protocols, reactive protocols store routing information with less memory. However, the time required to create routes is high due to the need to discover the desired way - instead of just searching in the routing tables.

Lastly, hybrid protocols combine the main characteristics of reactive and proactive protocols. They often operate by dividing the network into groups or zones. The routing occurs proactively inside each group while they behave reactively outside the groups [25].

This strategy aims to cluster together nodes that are close to each other or communicate more often. Because the frequency of communication between nodes from different groups is low, a reactive routing approach can be used without significantly impacting the protocol's overall performance. One of the hybrid protocols' most significant challenges is dividing the nodes into groups [5].

As mentioned, for the routing protocols designed for MANETs, thead hoc on-demand distance vector (AODV) is a reactive protocol commonly used in MANETs and focus of investigation.

III. IMPROVEMENTS ON THE AODV's ROUTING RECOVERY MECHANISM

This section presents a set of related works which aims to improve the route recovery mechanism of the AODV regarding a collection of quality of service (QoS) metrics. Some results tackled this issue by proposing new route recovery, while others focused on improving the original strategy. For example, the AODV-BR [26] and AODV-ABR [27] variants improve the AODV recovery strategy by supplying multiple backup routes to replace broken paths. However, this approach may need to be more efficient in a dense environment [28].

To overcome the drawback of the AODV-BR and AODV-ABR, Jeon et al. proposed the implicit backup routing-AODV (IBR-AODV) [28]. Their method employs local recovery of routes for reliability and reduces the number of control messages for efficiency. It implicitly conducts a route recovery process considering the mobility of a backup node. The results indicate the superiority of the IBR-AODV to the others regarding the number of link failures, data delivery ratio, message overhead, and end-to-end delay.

Similarly to the IBR-AODV, the bidirectional route repair method (BRRM-AODV) also claims to improve the route recovery speed [29]. This approach is bidirectional since when an old route disconnects, the source and destination start the route discovery simultaneously to shorten the disconnection duration. Moreover, this version features a density-based method for minimising the hop count in the repaired route and improving the successful probability of repairing the path. The simulation results indicate that the proposed method can reduce the route construction time by more than 20% and reduce the failure probability of route reconstruction by almost 50% compared with the AODV routing protocol. This method can also eliminate from 10% to

20% of the nodes participating in relaying protocol messages during the route discovery procedure.

Another extension of the AODV protocol was proposed by Castellanos et al. and claimed to provide a better mechanism to detect the link failures in a route and reestablish the connections considering the conditions of QoS that have been established during the route discovery phase [30]. Adaptive QoS-Aware AODV (AQA-AODV) claims improvements in packet delay, number of link failures, and connection setup latency compared with protocols like AODV.

Another approach for optimising the performance of routing protocols for MANETs is using bio-inspired optimises. Pereira et al. [19] introduced node connectivity to the AODV and employed the particle swarm optimisation (PSO) algorithm to select parameters. The idea was to use the PSO to find the best AODV route repair mechanism values. The results achieved by their proposal were superior to the standard AODV in most of the scenarios considered. More details on this approach are presented in Section IV-A.

In 2017, Santana et al. [21] extended the work of Pereira et al. [19] by comparing the performance of the PSO to another swarm-based optimiser: the artificial bee colony algorithm (ABC). They concluded that the optimisation of the AODV with the PSO and ABC achieved superior results than the standard AODV. However, there were no statistical differences between the results achieved by the PSO and ABC.

Unlike Santana et al., Maleki et al. uses optimizers to select whether a local or a source repair will be conducted when a route is broken [31]. In this version, GA-AODV, a genetic algorithm (GA), is the optimiser employed to find the best decision strategy based on the routing overhead, average end-to-end delay, and packet delivery ratio metrics. A similar approach was presented in 2020 by KN et al. [32]. In their process, particle swarm optimisation was applied to optimise the route recovery strategy of the AODV in a wireless sensor network to improve the packet delivery ratio and decrease routing overhead.

Despite all the advances in this field, there is still room for improvement. Furthermore, based on the no-free-lunch theorem for optimisation, we know that no solution based on optimizers can have the best performance in all scenarios. Hence, new and improved approaches can be proposed to meet the demands of specific applications.

IV. THEORETICAL BACKGROUND

A. AODV ROUTING PROTOCOL

The ad hoc on-demand distance vector (AODV) is a well-known reactive routing protocol with multi-hop and dynamic communication between mobile nodes [33], [34]. As a reactive protocol, the AODV works on demand to establish, recover and update routes. Moreover, to reduce the traffic of unnecessary routing packets in the network, the AODV only maintains active ways that avoid repair routes that are not in use or may never be used [35].

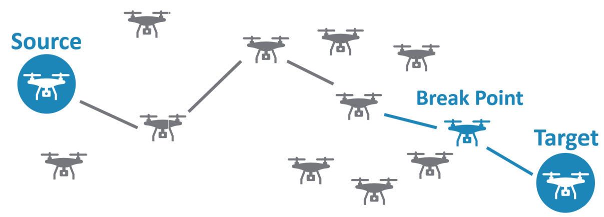

FIGURE 1. An example of MANET is where the UAVs represent nodes, and an active route connecting a source to a target node comprises the nodes connected by the grey and green lines. Note that the node described as "breakpoint" indicates a node that will cause the route to break.

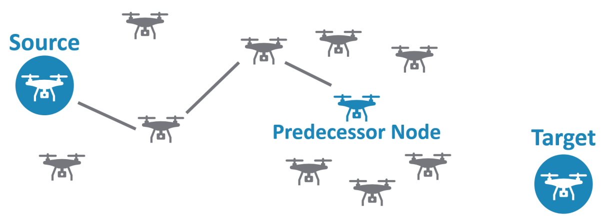

FIGURE 2. Same network of Figure IV-A but with the breakpoint node and its connections removed. Also, the predecessor node to the breakpoint is highlighted.

When a node needs to send information and does not have an active route to a destination node, it starts a route discovery process. The route discovery process begins with the initial node (source) broadcasting a route request packet (RREQ) to all nodes under its communication range (i.e., neighbours). If any neighbour that received the RREQ packet has a valid route, it replies to the RREQ with the desired path [18]. Otherwise, the route request packet is broadcasted by the neighbours that do not have the way expected. This process continues until a route is found or a time limit is exceeded.

Besides creating a new route, route repair is another essential procedure to maintain the information flow between the nodes in the network. The AODV uses two mechanisms to repair broken paths called source and local repairs. Consider the network configuration depicted in Figure IV-A, in which the UAVs represent the nodes, and the grey and green lines indicate an active route that connects the source to the target node. Also, consider that the node's movement described as a "breakpoint" will break the way.

When the breakage is detected by the predecessor node (Figure 2), it decides if it will attempt a local repair or inform the source node that the route is no longer valid and a source repair should be performed.

The predecessor node decides if it will try the local repair based on the number of hops between the source and the predecessor node (*packetForward*) and the number of hops between the predecessor node and the destination node (*predecessorHopCount*). A local repair is made if the predecessor-target path is shorter than the source-predecessor path (i.e., *packetForward* > *predecessorHopCount*). Otherwise, a route error packet (RERR) is sent to the source node to start the source repair. In the example of Figure 2, we have

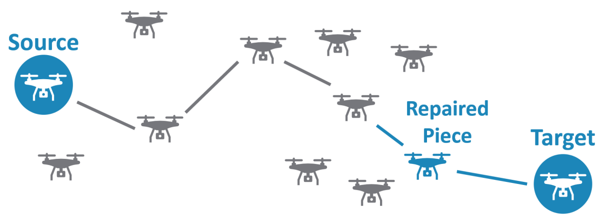

FIGURE 3. Example of a possible outcome to the repair process on the scenario described by Figure 2. Note that the new part of the repaired route is highlighted in green.

packetForward = 3 and *predecessorHopCount* = 2, which means that a local repair would be attempted, and a possible outcome is presented in Figure 3.

When a local repair is selected, the predecessor node uses the route discovery process to find a new route to the target. During this process, data packets are buffered in the predecessor node, and if no valid path is found, all the buffered data is dropped, and it propagates a route error packet (RERR) to inform the source node that the route is no longer valid. Note that a failure in the local repair process generates a significant data loss and increases the routing delay since the time to create a new route will equal the time to perform the local and source repair. Moreover, this also can impact the energy consumption of the nodes. Researchers have proposed modifications to the decision process to avoid the performance issues caused by selecting the route repair mechanism. One of the approaches used to minimise these issues was presented by Pereira et al. [19]. The idea behind this method is explained in the next section.

1) IMPROVING THE AODV ROUTE REPAIR SCHEME

A study from 2009 showed that the different route repair options on the AODV could produce different results depending on the scenario analysed [18]. In other words, the selection between local and source repair can impact the overall performance of the AODV. In this sense, Pereira et al. [19] proposed an approach to improve the AODV by modifying the route repair decision process. This approach has the assumption that more connected nodes are more likely to find feasible new routes than less connected nodes. Besides *packetForward* and *predecessorHopCount*, the connectivity can be defined as the number of neighbours or nodes under the communication range (Figure 4), and the connectivity of the source and target nodes are also considered to choose the route repair mechanism.

Furthermore, they proposed using weights to adjust the importance of terms considered to select the route repair. These weights were introduced to represent the impact of the connectivity, *packetForward*, and *predecessorHopCount* vary according to the characteristics of the networks (e.g., number and velocity of nodes) and the environment's features (e.g., size and presence of obstacles).

Algorithm 1 summarises the modified route repair decision process. The source node weights ($SW1$ and $SW2$) controls,

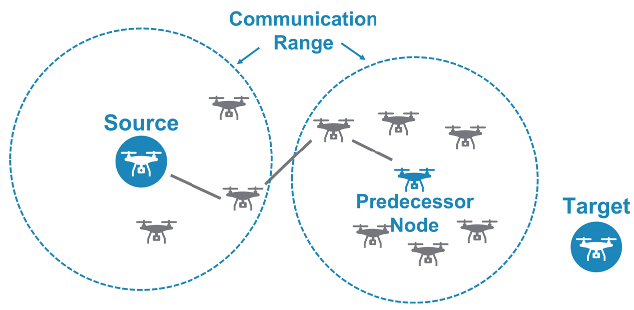

FIGURE 4. Illustration of a route breakage scenario where we can see that the set of neighbours of a node is composed of all the nodes under its communication range (dashed green circles). In this example, the connectivity of the source node is three, while the connectivity of the predecessor is six.

respectively, the importance of the number of hops between the source and the predecessor node and the connectivity of the source (*sourceConnectivity*). In the same way, $LW1$ and $LW2$ are the weights of the number of hops between the predecessor and the destination node and the predecessor node's connectivity (*predecessorConnectivity*).

Pereira et al. [19] used the PSO to find their previous work's best local and source weight values. However, this approach is limited to optimising the AODV concerning a single metric. Since the routing problem in MANETs is naturally multi-objective (i.e., reducing energy consumption while keeping the routing delay low), we propose to use multi-objective algorithms to optimise the routing performance of the AODV considering multiple QoS metrics.

Algorithm 1 Modified Route Repair Decision Scheme

source = (SW1 · sourceHopCount) + (SW2 · sourceConnectivity);
local = (LW1 · predecessorHopCount) + (LW2 · predecessorConnectivity);
if (source $\leq$ local); **then**
    Local Repair;
end if
if (source $>$ local); **then**
    Source Repair;
end if

B. MULTI-OBJECTIVE OPTIMISATION AND PARETO DOMINANCE

In mono-objective optimisation problems, the goal, in general, is to find the maximal or minimal value of a predetermined cost function [36], [37]. In contrast, multi-objective optimisation (MOO) aims to optimise a set of conflicting objective functions simultaneously [38], [39]. It means that a candidate solution has to satisfy the posed constraints and give a reasonable value to all objective functions simultaneously according to a predefined rule (in our case, the dominance concept). The optimisation methods usually determine a group of solutions named Pareto optimal,

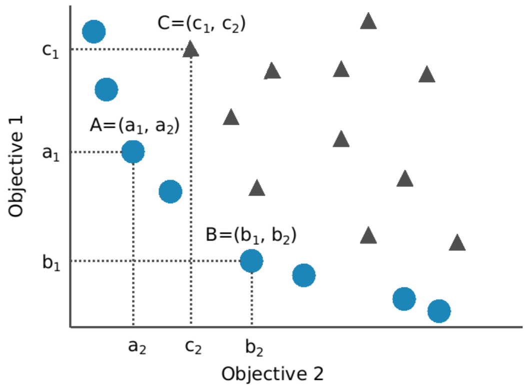

FIGURE 5. Example of Pareto Front for the minimisation a problem with two conflicting objective functions. The blue circles and the triangles are, respectively, the non-dominated and dominated solutions. The set composed of all circles if the the Pareto front.

and inside this set, we store non-dominated solutions. A non-dominated solution can be viewed as a solution that can only strictly improve one of its objectives without worsening at least one of the remaining ones [40], [41]. A multi-objective problem can be defined as follows [40], [42]:

Min $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})]$,

subject to $(\mathbf{x}) = (x_1, x_2, \ldots x_n) \in \Omega \subset \Re^n$,

where $\mathbf{x} \in \Re^n$ is the vector containing the n decisions variables, Ω is the set of the feasible decision vectors which a group of constraints determines, $\mathbf{F}$ is the vector containing all the mono-objective functions $f_m(\mathbf{x})$ and $m = 2|3$ is the number of objective functions which have to be optimised at the same time.

Mathematically, a solution $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \Re^n$ are non-dominated if there is no solution vector $\mathbf{z} = (z_1, z_2, \ldots, z_n) \in \Re^n$ such that $f(\mathbf{z}) \leq f(\mathbf{x})$ and at least one $f(\mathbf{z}) < f(\mathbf{x})$. Similarly, we say that $\mathbf{x}$ dominates $\mathbf{z}$ ($\mathbf{x} \prec \mathbf{z}$) if and only if two conditions are satisfied (in minimisation problems):

i) $\forall i \in (1, 2, \ldots, n), x_i \leq z_i$;

ii) $\exists i \in (1, 2, \ldots, n), x_i < z_i$.

Solutions are incomparable if distinct solutions cannot dominate each other. In this case, all non-dominated solutions belong to the Pareto optimal set, nominated in the space of objectives as Pareto front [39]. Figure 5 depicts an example of a Pareto front set for a given problem where the goal is to minimise two conflicting objectives.

As illustrated in Figure 5, The circles represent the non-dominated elements, the triangles are the dominated ones, and the set composed of all the circles is the Pareto front for that specific problem. In this example, the C is dominated by A and B (A and B dominates C). However, since a_2 is better than b_2 but a_1 is not better than b_1, we say that A and B are incomparable.

The multi-objective optimisation algorithms were designed to find answers in the Pareto front. They employ the Pareto dominance as a criterion for fitness assignment in multi-objective problems [43].

C. SELECTED OPTIMISATION ALGORITHMS

This section describes the multi-objective optimisation algorithms (MOA) addressed in this work: NSGA-II, SPEA2, and SMPSO. Generally, a good MOA must find a set of non-dominated solutions regarding all solutions and approximate as best as possible to the genuine Pareto front (PF). It has two goals: i) the non-dominated solutions obtained should be as close as possible to the true PF, and ii) the solutions should be as diverse as possible. The NSGA-II, SPEA2, and SMPSO use different strategies to accomplish these goals [44].

1) NSGA-II

Multi-objective evolutionary algorithms (MOEA) are a prominent class of optimisation methods to deal with mono and multi-objective problems using evolutionary mechanisms [45]. Undoubtedly the NSGA-II (Non-dominated Sorting Genetic Algorithm) [46] and the SPEA2 (Strength Pareto Evolutionary Algorithm) [47] are the most well-known and used MOEAs in recent literature [40], [48].

The NSGA-II presents two important operations: assigning a non-domination rank and calculating the crowding distance [46]. In the first case, a non-dominated sorting process must be performed, in which non-dominated fronts (groups) F_k, $k = 1, \ldots, n_k$ are created. For $k = 1$, the solutions are all non-dominated among each other. In F_2 are the chromosomes dominated just by the individuals in F_1, and so on. In F_{n_k} are the fully dominated solutions. Therefore, each individual $\mathbf{x}_i$ in F_k is assigned a non-domination rank value $i_{rank} = k$ [43].

The next step is to calculate the crowding distance of the population, front by front. This action intends to estimate the density of solutions in the same front F_k. The lower the domination rank, the better the resolution. In this case, the algorithm drives to the actual Pareto front. It is essential to favour solutions with smaller crowding distances to ensure a good spread of solutions that may cover the entire PF [40], [46]. Diversity in population is mandatory [39], [42], [46].

Based on these premises, the NSGA-II is initiated randomly, generating a population ($\mathbf{X}_0$) of N individuals (chromosomes) $\mathbf{x}_i$. Then, they are assigned the non-domination rank, and the crowding distance is calculated [39].

First, we choose the individuals (parents) of the current population ($\mathbf{X}_t$) using the binary tournament with the reposition method. The selections are made based on higher fitness or higher crowding distance. Then, we repeat the process until all the N individuals are chosen.

The next step is to perform the crossover operation, and in this step, we use the crossover probability parameter (p_c). We sort $r \in [0, 1]$ for each pair of parents previously selected. If $r > p_c$, two new individuals are created. If $r < p_c$ the crossover does not occur, and the parents are retained in the offspring ($\mathbf{X'}_t$). Then, we apply the mutation procedure with probability p_m to all individuals of the offspring.

The algorithm then combines the parents ($\mathbf{X}_t$) and the offspring ($\mathbf{X'}_t$), generating a population of $2N$ individuals that is sorted after the calculation of the non-dominated fronts and the crowding distances. Lastly, we select for the next generation the best half of individuals. We present the Algorithm 2 to summarise the process.

Algorithm 2 NSGA-II Pseudocode

```
Initialise the parameters;
Generate the initial population randomly using a uniform distribution;
Evaluate the fitness of the whole population;
Calculate F_k and the crowding distance of each agent;
while a termination criterion is not met do
    Select the parent chromosomes from the population using a binary tournament;
    Apply the crossover to parents, forming the new population;
    Execute the mutation to each new chromosome;
    Evaluate the new individuals;
    Combine the current population with new individuals;
    Calculate all F_k groups and the crowding distance;
    Select the best 50% of the population to generate the new population;
end while
return Best Pareto found;
```

2) SMPSO

Since the introduction of the Particle Swarm Optimization Algorithm (PSO) by Kennedy and Eberhart in 1995 [49], it became the most famous swarm-based optimisation algorithm to deal with mono-objective problems [50], [51], [52], [53].

In this sense, several proposals for PSO-based algorithms to deal with multi-objective tasks are also in the literature. Undoubtedly, the most known recommendations are the multi-objective PSO (MOPSO), introduced by Moore and Chapman [54], and the speed-constrained multi-objective particle swarm optimization (SMPSO), from Nebro et al. [55]. In addition, many other proposals adding improvements to the methods mentioned earlier are available, as discussed in Reyes et al. [56]. In this work, we chose the standard version of the SMPSO because it is a state-of-art proposal yet.

In PSO-based algorithms, the potential solutions are called particles, and a population (set of particles) is named Swarm. Two basic equations need to be updated at each iteration. The first is the position $\mathbf{x}_p^t$ of each particle $p = 1, 2 \ldots, P$ at iteration t, calculated according to Equation 1.

$$\mathbf{x}_p^{t+1} = \mathbf{x}_p^t + \mathbf{v}_p^{t+1}, \tag{1}$$

where $\mathbf{v}_p^{t+1}$ is the particle's velocity updated according to Equation 2

$$\begin{aligned}\mathbf{v}_p^{t+1} = \chi[\omega \mathbf{v}_{pd}^t + c_1 \mathbf{r_1} \otimes (\mathbf{pbest}_{pd}^t - \mathbf{x}_{pd}^t) \\ + c_2 \mathbf{r_2} \otimes (\mathbf{gbest}_d^t - \mathbf{x}_{pd}^t)]\end{aligned} \tag{2}$$

where ω is the inertia weight, $\mathbf{r_1}$ and $\mathbf{r_2}$ are random vector generated uniformly in the interval [0,1] for each variable d (dimension), c_1 and c_2 are the cognitive and social rates, respectively; $\mathbf{pbest}_{pd}^t$ is the best position found along the iterations by particle p (the best individual experience or the situation that led to the best performance index) and $\mathbf{gbest}_{pd}^t$ is the best position found by a predefined neighbour (the best collective experience). A main modification in the SMPSO is using a constriction coefficient χ to limit the speed of the particles, proposed by Clerk and Kennedy [57]. Classical approaches, as the MOPSO adopts upper and lower limits to the actual velocity. The χ coefficient is calculated as follows:

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}}, \tag{3}$$

where

$$\varphi = \begin{cases} c_1 + c_2 & \text{if } c_1 + c_2 > 4 \\ 1 & \text{if } c_1 + c_2 \leq 4. \end{cases} \tag{4}$$

The new velocity update calculation is done using Equation 5

$$\begin{aligned}\mathbf{v}_p^{t+1} = \chi[\omega \mathbf{v}_{pd}^t + c_1 \mathbf{r_1} \otimes (\mathbf{pbest}_{pd}^t - \mathbf{x}_{pd}^t) \\ + c_2 \mathbf{r_2} \otimes (\mathbf{gbest}_d^t - \mathbf{x}_{pd}^t)].\end{aligned} \tag{5}$$

Besides, the accumulated velocity of each variable d for each particle is constricted according Equation 4:

$$v_{pd}^t = \begin{cases} \delta_d & \text{if} v_{pd}^t > \delta_d \\ -\delta_d & \text{if} v_{pd}^t \leq -\delta_d \\ v_{pd}^t & otherwise, \end{cases} \tag{6}$$

where $\delta_d = \frac{(v_d^{max} - v_d^{min})}{2}$, v_d^{max} and v_d^{min} are the upper and lower limits of the velocity in terms of the variable d.

In summary, the velocities of the swarm are calculated by Equation2 and then multiplied by the constriction factor of Equation 3. Finally, the result is constrained using the expression defined in Equation 6.

Another essential step in the SMPSO algorithm applies the turbulence operator based on a mutation [45]. We adopt the polynomial mutation operator described by Deb and Deb [41], applying it in 15% of the variables, considering all the swarm ($p \times d$ variables). The new particle formed $\mathbf{x}_{pd}^{t,mut}$, which substitutes the old one, is modified in the dimension d according to Equation 7.

$$\mathbf{x}_{pd}^{t,mut} = \begin{cases} \mathbf{x}_{pd}^t + \zeta(\mathbf{x}_{pd}^t - \mathbf{x}_d^L) & \text{if } r \leq 0.5 \\ \mathbf{x}_{pd}^t + \zeta(\mathbf{x}_{pd}^U - \mathbf{x}_d^U) & \text{if } r > 0.5, \end{cases} \tag{7}$$

where p and d are, respectively the numbers of individuals in the swarm and the number of variables, r is a number randomly generated according to the uniform distribution within the interval [0,1], $\mathbf{x}_d^L$ and $\mathbf{x}_d^U$ are the lower and upper bounds of variable d, respectively, and ζ is calculated as follow:

$$\zeta = \begin{cases} (2r)^{\frac{1}{(1+\eta m)}} - 1 & \text{if } r \leq 0.5 \\ 1 - (2(1-r))^{\frac{1}{(1+\eta m)}} & \text{if} r > 0.5 \end{cases} \tag{8}$$

where $\eta_m \in [20,100]$ is a user-defined index parameter.

The last step of the SMPSO is to define an external archive, or leaders archive, with size $S < P$ composed of the non-dominated solutions. If the library becomes full or, in other words, if there are more than S non-dominated solutions, the crowding distance is used to select the particles that remain. Algorithm 3 summarises the description of the SMPSO.

Algorithm 3 SMPSO Pseudocode

Initialise all particle's positions randomly using a uniform distribution;
Initialise the particle's velocity and set their best-known position as their initial position;
Evaluate the fitness of the swarm;
Initialise the leaders' archive with the non-dominated vectors;
Set as the position of the particle which has the best fitness;
while a termination criterion is not met **do**
  for each particle $i = 1, \ldots, N$ **do**
    Calculate the particle's velocity and then update its position;
    Apply the mutation/turbulence operator;
    Update particle's fitness;
    if $f(x_i) < pbest_i$ **then**
      Update the particle's best-known position;
    end if
    if $f(x_p) < gbest$ **then**
      Update the swarm's best-known position;
    end if
    Update the leaders' archive;
  end for
end while
return The leaders archive;

3) SPEA2

The second version of the strength pareto evolutionary algorithm (SPEA2) is a multi-objective evolutionary algorithm, such as the NSGA-II. Zitzler et al. [47] introduced SPEA2. The method has gained much attention in the last decade and is considered efficient in many applications [40], [58].

The SPEA2 presents three critical features which differ it from the NSGA-II [41], [47]:

i) it utilises a fitness assignment mechanism for each individual;

ii) simultaneous maintenance of two populations, the first composed of individuals who perform the search process and an external archive to store the non-dominated solutions found during the search process;

iii) the density of the neighbourhood of each drives the search.

Besides, the strength of Pareto is essential once it presents how close the solutions are to the first rank, as defined in NSGA-II.

The goal of the SPEA2 is to obtain orderly distributed Pareto solutions by managing the external archive. Also, it presents a few configuration parameters and relatively fast convergence.

The steps to implement the SPEA2 are described as follows. Again, consider $\mathbf{X}_t$ the population at iteration t. Here, define $\mathbf{A}_t$ as the external archive containing up to N_a chromosomes.

The initialisation randomly generates the initial population $\mathbf{X}_0$ and an empty archive $\mathbf{A}_0$. Then, we calculate the fitness of each individual.

Before discussing the steps in the algorithm's main loop, it is necessary to define some variables to allow the fitness assignment. The first is the strength Pareto S_i of each chromosome i, calculated according to Equation 9.

$$S_i = \sum_{i \succ j} S_j \tag{9}$$

where N_{di} is the number of solutions dominated by $\mathbf{x}_i$, $j = 1, \ldots, N_{di}$ is the index of the individual that $\mathbf{x}_i$ dominates considering $\mathbf{X}_t \cup \mathbf{A}_t$. The strength is the number of individuals that $\mathbf{x}_i$ dominates in the current iteration t.

Following, the raw fitness R_i of the individual i is

$$R_i = \sum_{i \prec l} S_l \tag{10}$$

in which N_{dom_i} is the number of solutions that dominates $\mathbf{x}_i$, $l = 1, \ldots, N_{dom_i}$ is the index of the individuals which dominates $\mathbf{x}_i$ in $\mathbf{X}_t \cup \mathbf{A}_t$. The raw fitness is the sum of the strength of the individuals which dominates $\mathbf{x}_i$ in the current iteration t. The higher the R_i, the worse is $\mathbf{x}_i$.

Then, the chromosome's density is estimated using the K-nearest neighbour method, using Equation 11.

$$D_i = \frac{1}{\sigma_i^k + 1} \tag{11}$$

where σ_i^k is the distance to the k-th nearest neighbour in the objective space. The insertion of this variable tends to lead the algorithm to explore sparsely populated regions.

Finally, the fitness of $\mathbf{x}_i$ is calculated by Equation 12.

$$F_i = R_i + D_i \tag{12}$$

Therefore, the objective function of the SPEA2 is to minimise fitness. Observing that R_i intends to approximate the PF and D_i brings diversity in the objective space.

The external archive $\mathbf{A}_t$ is updated by inserting non-dominated solutions. This process must consider that the number of keys is constant and equals N_a. If the current number of individuals in iteration t is less than N_a, the archive is completed using the best-dominated solutions regarding fitness. Otherwise, the exceed chromosomes are eliminated, considering those with a shorter distance to their k nearest neighbours.

The selection, crossover, and mutation operations are the same as NSGA-II, described in Section IV-C1. The compiled process of the SPEA2 is in Algorithm 4.

Algorithm 4 SPEA2 Pseudocode

Initialise the population size (N), external archive size N_a and stop criteria;
Generate the initial population of chromosomes randomly using a uniform distribution;
Produce an empty external archive;
Evaluate the fitness of each individual in the population;
while a stop criterion is not reached **do**
  Find the non-dominated items in the external archive;
  if *Number of non-dominated elements* $> N_a$ **then**
    Calculate the crowding distance of the non-dominated set;
    Keep the best N_a non-dominated elements in the external archive;
  else
    Fill the archive until it becomes full using best-dominated vectors;
  end if
  Select the parent chromosomes in the population using a binary tournament;
  Execute the crossover operator to parents, forming the new population;
  Perform the mutation to each chromosome in the generated individuals;
  Evaluate all the new individuals;
  Combine the current population with the new chromosomes;
end while
return The set of non-dominated vectors in the external archive

V. PROPOSED SOLUTION

Based on the methodology proposed by Pereira et al. [19], we summarise our solution's architecture in the diagram depicted in Figure 6. The first step of the proposed solution consists of using a multi-objective algorithm to generate possible values for the parameters that represent the weights of the source repair (i.e., SW1 and SW2) and local repair (i.e., LW1 and LW2). Next, to access the quality of the values generated, we employed the NS2 platform to simulate the AODV protocol within the connectivity framework, incorporating the weight values derived from the multi-objective algorithm. Following this, a script was employed to parse the output files from each simulation, extracting the pertinent metrics such as Delay, packet loss ratio (PLR), network route load (NRL), and energy consumption (EC). These metrics were then fed back into the multi-objective algorithm, measuring the quality (fitness) of the generated solution. This iterative process continues until the stop criteria of the metaheuristic are met.

A real-value strategy was used for all optimizers to encode the solutions. The mutation and crossover operators used actual codification for the evolutionary-based algorithms. Each population element has a candidate solution represented by a four-dimensional array. In this array, each dimension corresponds to one of the parameters to be optimised (e.g., SW1, SW2, LW1, and LW2). We fixed the search space for all dimensions as [-1, 1] based on previous works and experiments [19], [21].

No additional modification was required to the optimizers to ensure that different optimisation techniques could easily replace them in future experiments. All the problem-specific dependencies were modeled in the objective function class. In the jMetal framework [59], this class poses all the problem attributes such as the number of dimensions, search space range, and fitness function). When the optimizers generate a solution and need to evaluate its quality, they will call the objective function class, passing the solution array as an argument. Inside the function, a new NS2 simulation setup with be created using the values for SW1, SW2, LW1, and LW2 generated by the optimiser.

Next, the NS2 simulation will be executed using the specifications described in Section VI and the values generated by the optimiser. Once the simulation finishes, the output files are processed using a script that evaluates the QoS metrics' value (e.g., routing delay, energy consumption, packet loss, ratio, and route load). These values are sent back to the optimiser as the solution's fitness (i.e., quality) is produced. This process is repeated for all candidate solutions generated/updated during optimisation until the stop criteria are met.

Before performing the experiments with the multi-objective algorithms, we first needed to modify the AODV to account for the connectivity information in the new route repair decision mechanism. Instead of adding the connectivity as part of the routing process, previous works retrieved it from the simulation platform. This study is essential since different strategies to include connectivity into the AODV can worsen its performance, as illustrated in Table 1.

To include the connectivity information we considered the following possible approaches:

- **Create new packets for request/reply node connectivity (AODV-C1)**: We create two packets for requesting and replying with the connectivity information. When a route breakage is detected, the predecessor node will unicast packets to its neighbours requesting their connectivity information. It will wait for the replies and then proceed with the recovery decision. The advantage of this approach is to avoid flooding the network with packets since they are only used when needed. Nevertheless, its main drawback might be the increase in the repair time, as the connectivity information will be requested during the routing repair process. Additionally, we have the routing overhead of introducing two new types of packets circulating in the network and impacting the NRL metric.
- **Include the connectivity information in the data packets (AODV-C2)**: The idea here is to avoid increasing the number of routing packets in the network by

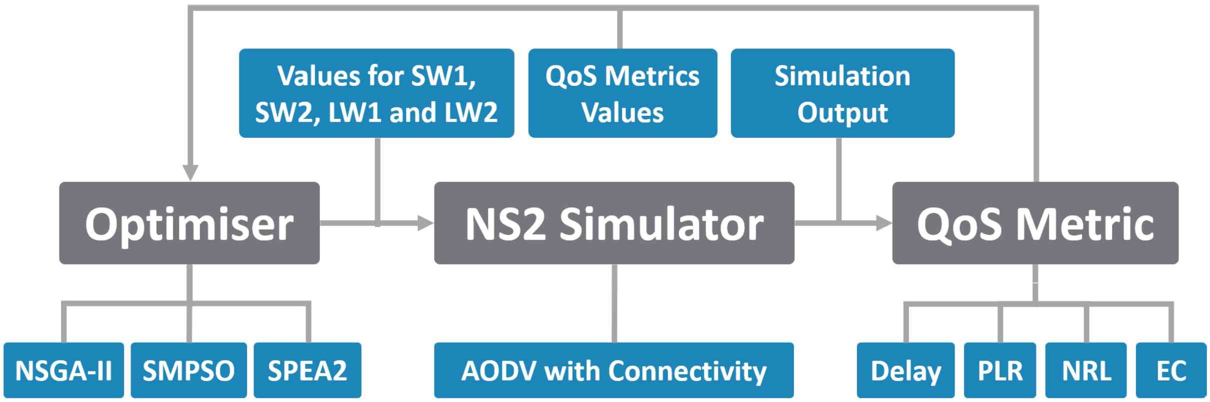

FIGURE 6. Representation of the proposed solution.

including the connectivity information in the structure of the regular data packets. This strategy's benefit is continuously updating the connectivity information of active nodes in the network. Besides, because the predecessor node will have all the necessary information to start the repair process, we expect the repair time to be the same as the AODV in the worst-case scenario. Nonetheless, the downside is that the nodes will need to store their neighbours' information, which may increase the memory usage in the nodes.

- **Modify the HELLO and RREQ packets to include the connectivity information (AODV-C3)**: This last approach incorporates using existing packages to include connectivity information. However, since the flow of HELLO and route request packets occur less often than the data packets, we expect to reduce the impact of this modification on the AODV performance. We modified the structure of these packets to add a field to send the nodes' connectivity. When a node receives these packets, it retrieves and stores/updates its neighbour connectivity information. The principal disadvantage is that the information will be updated less regularly than in the AODV-C2 approach.

We conducted experiments to assess the performance of the AODV-C1, AODV-C2, and AODV-C3, and the results are presented in Table 1. As we can see in Table 1, the AODV-C3 approach was the one that achieved the best results among the tested option. The better performance of the AODV-C3 approach might be because it does not introduce any extra route load with new routing packets as the AODV-C1. Also, because the Hello/RREQ packets circulate less often than the data packets, we believe that the impact of the AODV-C3 is also minimised compared to the AODV-C2. Hence, it was selected as the version used in the experiments with the multi-objective algorithms.

It is worth mentioning that, in another scenario where no modifications were introduced to the route recovery process, all of the proposed changes would harm the performance of the AODV. For example, introducing new packets will increase the routing overhead, and including further information on existing packets results in worse delay time and NRL. However, the results presented in Table 1 compare the standard AODV to a PSO-optimised AODV, which already modifies the route recovery strategy. For this reason, we can see that some of the proposed approaches present superior results to the standard AODV.

TABLE 1. Performance comparison considering the mean values of the mono-objective algorithms and the standard AODV.

Algorithm	Simulation Time (sec)	Delay	NRL	PLR
AODV	$\approx$ 130.000	2.968	6.741	0.496
AODV-C1	$\approx$ 130.000	2.132	31.590	0.842
AODV-C2	$\approx$ 150.000	3.898	15.200	0.695
AODV-C3	**$\approx$ 130.000**	**1.425**	**6.31**	**0.417**

VI. EXPERIMENTS AND RESULTS

In this section, we detail the experiments and results. Subsection VI-A explains the decisions regarding the characteristics of the simulated environment, its limitations, and the simulation configurations. Then, Subsection VI-B describes the metrics utilised to evaluate the algorithms. Finally, in Subsection VI-C, we analyse the results.

A. SIMULATION SETUP

The computational simulations were run in an Intel Xeon 3.1 GHz computer with 16 GB RAM and 1TB, running a Ubuntu 15.10 64-bit operating system. The algorithms were implemented in Java programming language using the jMetal Framework [59]. We noticed that it was necessary to develop a script in Bash to communicate the Java code with the network simulation platform.

The simulation platform was network simulator 2 (NS2) [60] version 2.35. The NS2 is a robust and well-known tool that implements several routing protocols, including AODV. Using this platform, we implemented the modifications on the AODV to include the new parameters for route recovery

TABLE 2. Configuration of the NSGA-II, SPEA2 and SMPSO. Where NV = Number of Variables, *pm* = Mutation probability and *pc* = Crossover Probability.

Parameter	NSGA-II	SMPSO	SPEA2
Population Size	100	100	100
Archive Size	–	100	100
Mutation	Polynomial $pm = 1.0/NV$	Polynomial $pm = 1.0/NV$	Polynomial $pm = 1.0/NV$
Selection	Binary Tournaments	–	Binary Tournaments
Crossover	Simulated Binary $pc = 0.9$	–	Simulated Binary $pc = 0.9$

decisions (SW1, SW2, LW1, and LW2), modifying the structure of the packets and the route repair mechanism.

Also, using the NS2, we modeled a network with 50 nodes with a maximum speed of 20 m/s. The environment dimensions were set to 1500 by 300 meters, and the duration of each simulation was 900 seconds with a pause time of 30 seconds (Sleep mode). The medium access (MAC) and the physical (PHY) layers follow the IEEE 802.11 at a bit rate of 2 Mbits/s, a transmission range of 250m, and the propagation model used was a two-ray ground.

The user datagram protocol (UDP) was used with a CDR traffic pattern for the traffic model. There are 30 traffic sources transmitting packets of 512 bytes at a rate of 4 packets per second. Both data and routing packets are buffered in a queue that holds at most 50 packages until the MAC layer can transmit them.

The energy model defines the initial energy of each node as 1000 Joules. Besides, the energy consumed during sleep mode is 1mW per second, the energy consumed to transmit packets was defined as 1.65W, and the energy used to receive packages equals 1.1W.

Table 2 shows the values for the parameters used during the experiments regarding the metaheuristics configuration. All the algorithms were executed 30 times, the stop criteria adopted was 20.000 fitness evaluations, and the search space for the four parameters (SW1, SW2, LW1, and LW2) was [-1, 1].

The values adopted in the simulations for the network simulator and the algorithms were based on previous works by [21] and [55] and validated experiments. With the earlier experiments, we verified that those values were sufficient to achieve satisfactory results. We highlight that increasing or decreasing those values does not necessarily imply better results. The multi-objective algorithms' fitness function (fit) is defined in Equation 13.

$$\text{Minimise } fit(x) = (NRL(x), EC(x)) \tag{13}$$

where NRL and EC are, respectively, the normalised route load, and the energy consumption and they are calculated using Equation 16 and Equation 17 respectively.

The mono-objective algorithms (ABC and PSO) were simulated using the same architecture as the multi-objective (Figure 6). The only difference is that the mono-objective algorithms used only the NRL as their fitness function in a minimisation process.

Besides the AODV, we selected the following protocols to compare the performance of the proposed approach with other proactive and reactive routing protocols in the literature: ad hoc on-demand multiple path distance vector (AOMDV) [61], destination sequenced distance vector (DSDV) [62], dynamic source routing (DSR) [63], and optimised link state routing protocol (OLSR) [64]. The AOMDV was selected as a more recent version of the AODV featuring a multiple path discovery capability, allowing it to use the backup route instead of the route recovery when a route is no longer valid. The DSR was selected for being an alternative reactive protocol to the AODV. It has a different route recovery strategy that only allows local repairs. On the other hand, the DSDV and OLSR were selected to represent distinct proactive routing protocols for MANETs. These four routing protocols were also simulated on the NS2 platform in the same scenario, energy model, and node network characteristics as the AODV but without optimisation.

B. PERFORMANCE METRICS

The performance of the routing protocols was assessed using the following metrics:

1) **Normalised route delay (RD or Delay)**: is defined as the average time it takes for data packets to arrive at the destination node divided by the number of active connections in the network. The Delay is calculated using Equation 14.

$$Delay = \frac{\sum_{i=1}^{DP} ArriveTime - SendTime}{\sum_{i=1}^{N} NumberOfConnections}, \tag{14}$$

where $ArriveTime$, and $SendTime$ are, respectively, the time when the packet arrived in the destination node, and the time when the packed left the source node, $NumberOfConnections$ is the number of active connections in the network, N is the number of nodes, and DP is the number of data packets received.

2) **Packet loss ratio (PLR)**: measures the fraction of the data packets that were not delivered to the destination as presented and can be calculated by applying Equation 15.

$$PLR = \frac{ReceivedDataPackets}{DataPackets}, \tag{15}$$

where $ReceivedDataPackets$ and $DataPackets$ are the numbers of data packets received and sent in the network.

3) **Normalised route load (NRL)**: is the number of routing packets ($RoutingPackets$) sent divided by the number of data packets ($DataPackets$) sent during the simulation (Equation 16):

$$NRL = \frac{RoutingPackets}{DataPackets}. \tag{16}$$

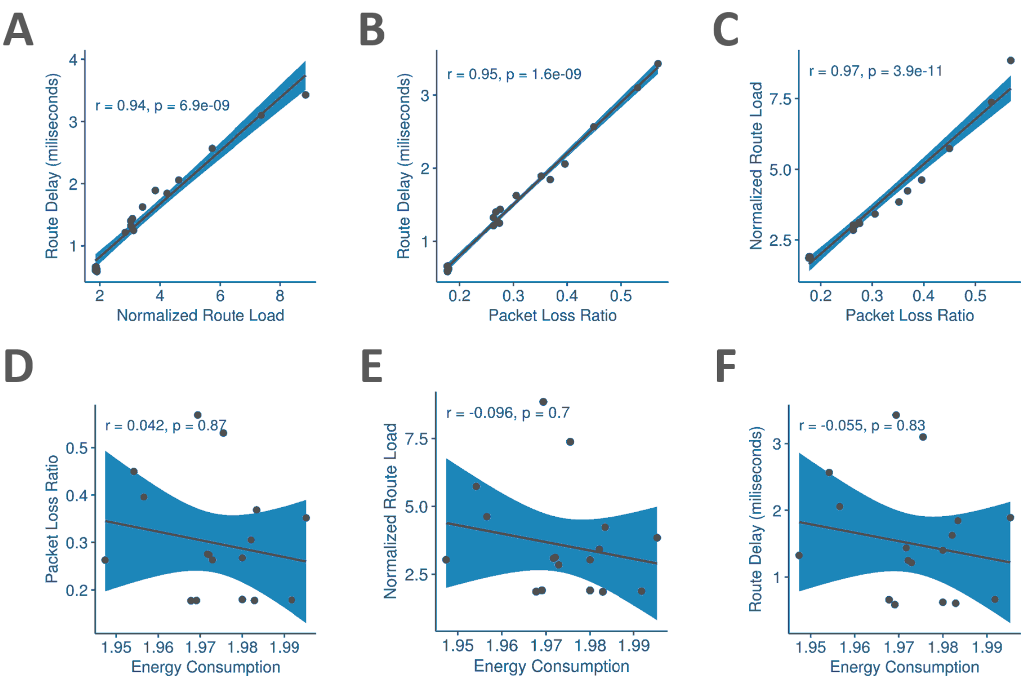

FIGURE 7. **Spearman correlation for the metrics: Normalised Route Load, Routing Delay, Paxcket Loss Ratio and Energy Consumption. Note that only energy consumption has a non-positive correlation with the other metrics.**

4) **Energy consumption (EC)**: is the average energy spent by the node from the beginning to the end of the simulation (Equation 17).

$$EC = \frac{\sum_{i=1}^{N} (InitalEnergy_i - FinalEnergy_i)}{N}, \quad (17)$$

where N is the number of nodes in the network, $InitalEnergy_i$ is the initial energy on the node i, and $FinalEnergy_i$ is the energy left on the node i at the end of the simulation.

Our goal is to minimise all the values for these four metrics. Furthermore, when dealing with multi-objective algorithms, it is necessary to work with conflicting objectives. As shown in Figure 7, the correlation analysis of the selected metrics reveals that NRL, RD, and PLR are positively correlated, and there is no conflicting relationship between them. On the other hand, the correlation between the energy consumed and the other metrics is not high, and, in some cases, we can see a slight negative correlation. For these reasons, Energy Consumption was selected as one of the objectives to be optimised. The Route Delay, NRL, and PLR metrics could be used for the second objective instead of the NRL without changing the final results. Since the goal is to minimise the selected metrics to achieve better results, the algorithms were implemented considering the minimisation of the objectives.

C. RESULTS

We found that NSGA-II, SMPSO, and SPEA2 presented a similar Pareto front while minimising energy consumption and normalised route load. Figure 8 A depicts the Pareto front achieved by the three metaheuristics. It is possible to observe that all algorithms found few solutions: four solutions found by NSGA-II and three obtained by SMPSO and SPEA2. Although the NSGA-II was able to get some superior answers regarding the SMPSO and SPEA2, the Pareto front of these two algorithms dominates the one found by NSGA-II.

Concerning the absolute variation of the metrics values, the PLR values variation is higher than the EC variation. This variation may occur because reducing the energy consumption is more complex than reducing the packet loss ratio. Given that the NRL measures the balance between the packets and data packets sent during the simulation and that the number of transmitted data packets is fixed, only the amount of routing packets can be reduced to minimise the NLR.

The algorithms must work on the routing repair process to reduce the routing packet traffic since it is the only one that interferes with routing packets and can be modified in the simulations. As shown in Figure 8 B, all metaheuristics tend to give higher weight to SW1 and SW2, which are responsible for the source route repair. As a result, the source repair tends to be executed more often than the local repair. This behaviour confirms the results of Pereira et al. [65], in which the source repair achieved better results than local repair in a network with a similar configuration to the one used in this work.

Table 3 shows that the NSGA-II, SMPSO, and SPEA2 found the same set of best values for the parameters. Furthermore, the achieved values also gave more weight to

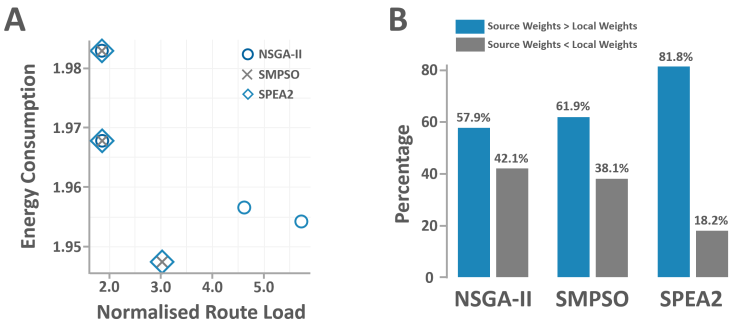

FIGURE 8. Characteristics of the solutions found by the multi-objective algorithms. Panel A shows Pareto Front achieved by the NSGA-II, SMPSO and SPEA2, while panel B displays the number of times (%) when the values found by the algorithms for the source repair (SW1 and SW2) weight is higher than the local repair (LW1 and LW2).

TABLE 3. Optimum values for A, B, C and D found by each algorithm, considering the best value as the one which presents the best balance between the normalised route load and the energy consumption.

Algorithm	SW1	SW2	LW1	LW2
NSGA-II	0.3782	-0.5895	-0.1362	-0.1362
SMPSO	0.3782	-0.5895	-0.1362	-0.1362
SPEA2	0.3782	-0.5895	-0.1362	-0.1362

the terms responsible for the source repair, validating that source repair presents better performance than local repair for the network scenario described in this paper.

In the NRL minimisation, the algorithm only reduces traffic or routing packets. However, reducing the traffic of data and routing packets is necessary to minimise the EC. This behaviour occurs because the network traffic is the primary energy consumption responsibility. Hence, the number of packages sent during the simulation does not change. Therefore, only the traffic and the re-transmission/reception of routing packets can be modified to minimise the EC.

A feasible strategy to minimise the EC is to reduce the network traffic by reducing the average network path length. In the context of the problem tackled in this work, since we are only dealing with route recovery, this reduction can be achieved by prioritising the reconstruction of shorter routes. The idea behind the strategy is that by reducing the length of the reconstructed paths, the overall number of packets received and re-transmitted will decrease. A direct consequence of this approach is reducing the Route Delay observed in part of the results. For this reason, the algorithms adopted a strategy that uses the average path length minimisation strategy in association with another method to minimise energy consumption.

However, the strategy to force the packet loss is also penalised once it harms the NRL, which is another objective function to be minimised. Loose/drop packets may indicate route break or failure in the route repair process. In both cases, the NRL will increase. Hence, a new repair operation will be needed, and these operations require the use of route packets.

TABLE 4. Mean Values and the Standard Deviation of the results achieved by the standard AODV, other routing protocols, and the association of AODV with multi and mono objective bio-inspired algorithms.

Algorithm	NRL	Delay	PLR	EC
AODV	6.7417 (0.0000)	2.9684 (0.0000)	0.4962 (0.0000)	1.9801 (0.0000)
AOMDV	1.5700 (0.0000)	0.8711 (0.0000)	0.3309 (0.0000)	2.0000 (0.0000)
DSDV	3.2134 (0.0000)	**0.0260 (0.0000)**	0.3848 (0.0000)	2.0000 (0.0000)
DSR	33.0000 (0.0000)	1.8187 (0.0000)	0.3333 (0.0000)	2.0000 (0.0000)
OLSR	**0.9563 (0.0000)**	0.7338 (0.0000)	0.4279 (0.0000)	2.0000 (0.0000)
AODV + NSGA-II	3.5185 (1.9696)	1.4720 (0.9911)	0.3001 (0.0143)	1.9653 (0.0131)
AODV + SMPSO	3.1211 (1.8281)	1.2889 (0.9915)	0.2667 (0.0128)	1.9630 (0.0157)
AODV + SPEA2	2.2500 (0.6777)	0.8633 (0.3999)	0.2058 (0.0493)	1.9660 (0.0178)
AODV + ABC	2.0227 (0.4920)	0.7044 (0.3235)	**0.1950 (0.0403)**	1.9950 (0.0043)
AODV + PSO	2.1868 (0.2812)	0.6083 (0.2100)	0.1881 (0.0319)	**1.9710 (0.0140)**

Consequently, the best solution is the one that has the best balance between energy consumption and the NRL. We assume that the selected solution can minimise the NRL without compromising the node's energy in the network. As a result, the three algorithms found the best solution, and the values are $NRL = 1.8612$, $PLR = 17.7074$, $RD = 0.6591$, and $EC = 1.9678$. Table 4 presents the mean and the Standard Deviation of the results achieved by the Algorithms.

As seen in Table 4, the results obtained by the algorithms in association with AODV were better than the standard AODV in all analysed metrics. Furthermore, applying the Wilcoxon test with a significance level of 5% confirmed that the results of the three algorithms are indeed different from the AODV. Moreover, compared to reactive and proactive routing protocols concerning energy consumption and packet loss ratio, the proposed methodology overcame other routing protocols used in this study. However, our proposal overcame the DSR protocol regarding the NRL and Delay.

The OLSR produced superior results than the other protocols regarding NRL; this might be because this protocol has mechanisms to prevent redundant packets in the network. However, the drawback of this behaviour is to increase the risk of losing information, and as can be seen in Table 4, they had the second-worst performance regarding the PLR metric.

Concerning Delay, the DSDV protocol had the best result. This result can be because the proactive routing strategy helps maintain the updated routing tables, reducing the time required to send the packets. However, the proactive approach in this protocol led to an increase in the NRL and the EC.

Nonetheless, there is no statistical difference between them. Therefore, the similarity between the results can indicate that the algorithms may get trapped in local minimal, or the three multi-objective algorithms could find the best solution regarding NRL and EC.

When comparing the results of the multi-objective proposals with mono-objective algorithms (PSO and ABC) as reported in previous studies [21], regarding better results of NRL, the MOAs could still find a slightly better solution (Table 4). However, there is no statistical distinction between them when we apply the Wilcoxon test with 95% confidence interval (population of 30 elements of each algorithm).

In Table 4, it is possible to observe that the Energy Consumption of the solutions found by the multi-objective algorithms was slightly higher than the mono-objective values. It may be related to the fact that when the Packet Loss is reduced, the number of packets re-transmitted and received increases. Since the packet traffic is the primary component for the energy consumption in the network, it causes the elevation of EC. It is worth mentioning that the energy consumption increase was mainly due to the traffic of data packets. Hence, the number of routing packets (NRL) decreased.

Lastly, concerning the convergence of the algorithms, we observed in previous experiments that the definition of the stop criteria as 20.000 fitness evaluations was enough to allow all t three multi-objective algorithms to converge. The convergence happened even before the 20.000 fitness evaluations mark in most of the executions. The fast convergence can suggest that this problem is multimodal, and the algorithms get trapped in sub-optimum solutions. This hypothesis is further supported by the discrepancies between the solutions found by the NSGA-II and the other two algorithms. Because the NSGA-II were equally good or dominated by the other optimizers' answers, it could not improve them even after several iterations.

VII. DISCUSSION

The proposed methodology offers two fundamental benefits compared to the conventional AODV and analogous techniques documented in the literature. Firstly, it demonstrates enhanced flexibility while simultaneously maintaining compatibility with AODV-compatible devices. Diverse routing behaviours might be required depending on the specific attributes of the problem at hand. For example, different routing approaches might be needed depending on the network size, battery capacity, node velocity, and environmental dimensions. Given that our solution's optimisation phase considers these intricacies (i.e., the simulated routing scenario should capture these characteristics), it permits a more bespoke routing approach than conventional rigid methods.

The second benefit of our proposal pertains to implementation procedures and computational costs. The multi-objective optimisations can be conducted offline in a dedicated computer, removing the requirement of more robust hardware on the nodes. The only requirement is that the simulated environment mirrors real-world conditions. After achieving the optimal parameter values with the simulations, a customised version of the AODV featuring these determined values can be deployed as the routing protocol within MANETs' nodes. This attribute extends the compatibility of our solution to virtually any device capable of employing the standard AODV. Our solution retains retro-compatibility and does not impose substantial supplementary computational burdens, mitigating the potential rise in energy consumption among the nodes.

Our proposed solution also presents some trade-offs connected mainly to the modelling and simulation process and the nature of multi-objective optimisation. As mentioned, our approach optimises the route recovery of the AODV to suit the characteristics of the environment best. In this case, the simulated scenario must capture the main features of the actual conditions to ensure an optimal performance. Hence, better modelling can positively affect the implementation of the proposed solution.

Another drawback is related to the simulation process, which can be done offline on a computer with more robust hardware but requires hours to reach the optimal value. Because the simulation setup involves different programs (e.g., Python scripts, Java implementation of the multi-objective algorithms, and NS2 simulator), the interconnectivity between these components and the information exchange can be time-consuming.

Multi-objective optimisation is a more complex process than the mono-objective one, but the improvements achieved in one of the metrics can be slightly inferior compared to the mono-objective. For example, we can see in Table 4 that the mono-objective algorithms overall were able to find better NRL results than the multi-objective one. However, looking at the overall results, the multi-objective solutions reached better results than the mono-objective ones – a better trade-off between the reduction in the NRL and energy consumption simultaneously.

VIII. CONCLUSION

Mobile ad hoc networks are a field with several real-world applications varying from intelligent devices to swarm robotics. Regardless of the application, the section of the appropriate routing protocol is a critical step to achieving adequate performance. This study introduces a multi-objective optimisation approach for AODV, a widely employed routing protocol in MANETs. Through the optimisation of parameters governing the weight factors associated with source repair (SW1 and SW2) and local repair (LW1 and LW2) within the route repair decision process, we successfully enhance the protocol's overall performance across all designated metrics.

Another contribution of our work concerns the assessment of QoS metrics suitable for multi-objective optimisation. We show that there is a correlation between some of these metrics. Hence, the optimisation of one would result in the indirect optimisation of the others. From the metrics selected, only the energy consumption was not strongly correlated with the others. Moreover, comparing the proposed solution with other routing protocols and previous mono-objective approaches under the same environmental setup is also a novelty. This comparison allowed us to see these methods' performance differences, particularly between the multi-objective and mono-objective methods. Lastly, the set of best weights found by the optimisers supports the results of previous works, indicating that in most scenarios, the source repair mechanism of the AODV is more effective than the local repairs.

As a future direction of this research, we intend to compare the proposed approach with other reactive routing protocols, such as DSR and TORA, or even proactive and hybrid routing protocols for future work. Furthermore, we can further investigate how the velocity of the nodes, number of nodes, traffic intensity, and size of the environment impact the performance of the algorithms.

ACKNOWLEDGMENT
The authors thank the Federal University of Technology - Parana for the funding of this study.

CONFLICT OF INTEREST
The authors declare that they have no conflict of interest.

REFERENCES
[1] M. Conti and S. Giordano, "Mobile ad hoc networking: Milestones, challenges, and new research directions," *IEEE Commun. Mag.*, vol. 52, no. 1, pp. 85–96, Jan. 2014.
[2] C.-M. Huang and C.-F. Lai, "The delay-constrained and network-situation-aware V2V2I VANET data offloading based on the multi-access edge computing (MEC) architecture," *IEEE Open J. Veh. Technol.*, vol. 1, pp. 331–347, 2020.
[3] G. D. Singh, M. Prateek, S. Kumar, M. Verma, D. Singh, and H.-N. Lee, "Hybrid genetic firefly algorithm-based routing protocol for VANETs," *IEEE Access*, vol. 10, pp. 9142–9151, 2022.
[4] B. U. I. Khan, R. F. Olanrewaju, F. Anwar, A. R. Najeeb, and M. Yaacob, "A survey on MANETs: Architecture, evolution, applications, security issues and solutions," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 12, no. 2, pp. 832–842, 2018.
[5] F. Safari, H. Kunze, J. Ernst, and D. Gillis, "A novel cross-layer adaptive fuzzy-based ad hoc on-demand distance vector routing protocol for MANETs," *IEEE Access*, vol. 11, pp. 50805–50822, 2023.
[6] V. Bhardwaj and N. Kaur, "An efficient routing protocol for FANET based on hybrid optimization algorithm," in *Proc. Int. Conf. Intell. Eng. Manage. (ICIEM)*, Jun. 2020, pp. 252–255.
[7] C. Wang, J. Zhao, X. Wu, W. Ma, H. Tian, Q. Yu, H. Shi, Y. Yao, and X. Hao, "Elastic routing mechanism for flying ad hoc network," *IEEE Access*, vol. 10, pp. 98712–98723, 2022.
[8] S. Mohapatra and P. Kanungo, "Performance analysis of AODV, DSR, OLSR and DSDV routing protocols using NS2 simulator," *Proc. Eng.*, vol. 30, pp. 69–76, Jan. 2012.
[9] N. Shah, H. El-Ocla, and P. Shah, "Adaptive routing protocol in mobile ad-hoc networks using genetic algorithm," *IEEE Access*, vol. 10, pp. 132949–132964, 2022.
[10] A. S. Mustafa, M. M. Al-Heeti, M. M. Hamdi, and A. M. Shantaf, "Performance analyzing the effect of network size on routing protocols in MANETs," in *Proc. Int. Congr. Hum.-Comput. Interact., Optim. Robot. Appl. (HORA)*, Jun. 2020, pp. 1–5.
[11] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Netw.*, vol. 2, no. 1, pp. 1–22, Jan. 2004.
[12] F. Muchtar, A. H. Abdullah, S. Hassan, and F. Masud, "Energy conservation strategies in host centric networking based MANET: A review," *J. Netw. Comput. Appl.*, vol. 111, pp. 77–98, Jun. 2018.
[13] U. Srilakshmi, S. A. Alghamdi, V. A. Vuyyuru, N. Veeraiah, and Y. Alotaibi, "A secure optimization routing algorithm for mobile ad hoc networks," *IEEE Access*, vol. 10, pp. 14260–14269, 2022.
[14] S. Gautam and R. Kumar, "A review of energy-aware routing protocols in MANETs," *Int. J. Mod. Eng. Res.*, vol. 2, no. 3, pp. 1129–1133, 2012.
[15] M. Goyal, D. Goyal, and S. Kumar, "Dragon-AODV: Efficient ad hoc on-demand distance vector routing protocol using dragon fly algorithm," in *Soft Computing: Theories and Applications*. Singapore: Springer, 2020, pp. 181–191.
[16] K. Vyas and A. Chaturvedi, "Comparative analysis of routing protocols in MANETS," in *Proc. Int. Conf. Signal Propag. Comput. Technol. (ICSPCT)*, Jul. 2014, pp. 692–697.
[17] D.-W. Kum, J.-S. Park, Y.-Z. Cho, and B.-Y. Cheon, "Performance evaluation of AODV and DYMO routing protocols in MANET," in *Proc. 7th IEEE Consum. Commun. Netw. Conf.*, Jan. 2010, pp. 1–2.
[18] N. C. V. N. Pereira and R. M. de Moraes, "A comparative analysis of AODV route recovery mechanisms in wireless ad hoc networks," in *Proc. IEEE Latin-Amer. Conf. Commun.*, Sep. 2009, pp. 1–6.
[19] N. C. V. N. Pereira, C. J. A. Bastos-Filho, and R. M. Moraes, "Improving AODV route recovery mechanisms with connectivity and particle swarm optimization," *J. Commun. Inf. Syst.*, vol. 27, no. 1, pp. 15–21, Apr. 2012.
[20] Z. Zhaoxiao, P. Tingrui, and Z. Wenli, "Modified energy-aware AODV routing for ad hoc networks," in *Proc. WRI Global Congr. Intell. Syst.*, vol. 3, May 2009, pp. 338–342.
[21] C. J. Santana Jr., E. P. Alves, E. Figueiredo, M. Macedo, P. Santos, C. Bastos-Filho, H. Siqueira, and A. Gokhale, "Improving AODV routing protocol for mobile ad-hoc networks using swarm-based algorithms," in *Proc. 8th Brazilian Congr. Comput. Inteligence*, R. J. Niterói, Ed., 2018, p. 1.
[22] Y. de Souza Tadano, H. V. Siqueira, and T. A. Alves, "Unorganized machines to predict hospital admissions for respiratory diseases," in *Proc. IEEE Latin Amer. Conf. Comput. Intell. (LA-CCI)*, Nov. 2016, pp. 1–6.
[23] H. Siqueira and I. Luna, "Performance comparison of feedforward neural networks applied to streamflow series forecasting," *Math. Eng., Sci. Aerosp.*, vol. 10, no. 1, pp. 41–53, 2019.
[24] R. R. Ema, A. Anik, N. Nahar, M. A. Rahman, K. P. Eti, and T. Islam, "Simulation based performance analysis of proactive, reactive and hybrid routing protocols in wireless sensor network," in *Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2020, pp. 1–6.
[25] K. Pandey and A. Swaroop, "A comprehensive performance analysis of proactive, reactive and hybrid MANETs routing protocols," *Int. J. Comput. Sci. Issues*, vol. 8, no. 6, pp. 432–441, 2011.
[26] S.-J. Lee and M. Gerla, "AODV-BR: Backup routing in ad hoc networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, vol. 3, Sep. 2000, pp. 1311–1316.
[27] W. K. Lai, S.-Y. Hsiao, and Y.-C. Lin, "Adaptive backup routing for ad-hoc networks," *Comput. Commun.*, vol. 30, no. 2, pp. 453–464, Jan. 2007.
[28] J. Jeon, K. Lee, and C. Kim, "Fast route recovery scheme for mobile ad hoc networks," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2011, pp. 419–423.
[29] S.-C. Huang and H.-Y. Chang, "A bidirectional route repair method for wireless mobile ad hoc networks," *Comput. J.*, vol. 58, no. 2, pp. 338–353, Feb. 2015.
[30] W. Castellanos, P. Arce, P. Acelas, and J. C. Guerri, "Route recovery algorithm for QOS-aware routing in MANETs," in *Proc. Int. Conf. Mobile Lightweight Wireless Syst.* Berlin, Germany: Springer, 2011, pp. 81–93.
[31] A. Maleki and M. Sadeghzadeh, "Improved AODV route recovery in mobile ad-hoc networks using a genetic algorithm," *Manage. Sci. Lett.*, vol. 4, no. 9, pp. 2171–2176, 2014.

[32] K. N. K. Sunil and S. Shankar, "Improving the AODV route recovery mechanism using PSO in WSN," *Int. J. Sensors, Wireless Commun. Control*, vol. 10, no. 5, pp. 707–715, Feb. 2021.
[33] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Pers. Commun.*, vol. 6, no. 2, pp. 46–55, Apr. 1999.
[34] D.-G. Zhang, L. Chen, J. Zhang, J. Chen, T. Zhang, Y.-M. Tang, and J.-N. Qiu, "A multi-path routing protocol based on link lifetime and energy consumption prediction for mobile edge computing," *IEEE Access*, vol. 8, pp. 69058–69071, 2020.
[35] C. Perkins, E. Belding-Royer, and S. Das, *Ad Hoc On-Demand Distance Vector (AODV) Routing*, document RFC 3561, Jul. 2003.
[36] E. D. P. Puchta, H. V. Siqueira, and M. D. S. Kaster, "Optimization tools based on metaheuristics for performance enhancement in a Gaussian adaptive PID controller," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1185–1194, Mar. 2020.
[37] H. Siqueira, J. T. Belotti, L. Boccato, I. Luna, R. Attux, and C. Lyra, "Recursive linear models optimized by bioinspired metaheuristics to streamflow time series prediction," *Int. Trans. Oper. Res.*, vol. 30, no. 2, pp. 742–773, Mar. 2023.
[38] J. Branke, K. Deb, and K. Miettinen, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, vol. 5252. Berlin, Germany: Springer, 2008.
[39] Y. Collette and P. Siarry, *Multiobjective Optimization: Principles and Case Studies*. Berlin, Germany: Springer-Verlag, 2013.
[40] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5. New York, NY, USA: Springer, 2007.
[41] K. Deb and D. Deb, "Analysing mutation schemes for real-parameter genetic algorithms," *Int. J. Artif. Intell. Soft Comput.*, vol. 4, no. 1, pp. 1–28, 2014.
[42] K. Deb, "Multi-objective optimization," in *Search Methodologies*. Boston, MA, USA: Springer, 2014, pp. 403–449.
[43] C. A. C. Coello, "Evolutionary multiobjective optimization: Current and future challenges," in *Advances in Soft Computing*. London, U.K.: Springer, 2003, pp. 243–256.
[44] R. A. Coelho, F. D. R. N. Guimarães, and A. A. A. Esmin, "Applying swarm ensemble clustering technique for fault prediction using software metrics," in *Proc. 13th Int. Conf. Mach. Learn. Appl.*, Dec. 2014, pp. 356–361.
[45] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 1. Hoboken, NJ, USA: Wiley, 2001.
[46] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[47] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA 2: Improving the strength Pareto evolutionary algorithm," Comput. Eng. Netw. Lab. (TIK), ETH Zurich, Zurich, Switzerland, TIK Rep. 103, 2001, vol. 236.
[48] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "A survey of multiobjective evolutionary clustering," *ACM Comput. Surv.*, vol. 47, no. 4, pp. 1–46, Jul. 2015.
[49] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, 1995, pp. 1942–1948.
[50] S. J. Nanda and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," *Swarm Evol. Comput.*, vol. 16, pp. 1–18, Jun. 2014.
[51] P. Santos, M. Macedo, E. Figueiredo, C. J. Santana, F. Soares, H. Siqueira, A. Maciel, A. Gokhale, and C. J. A. Bastos-Filho, "Application of PSO-based clustering algorithms on educational databases," in *Proc. IEEE Latin Amer. Conf. Comput. Intell. (LA-CCI)*, Nov. 2017, pp. 1–6.
[52] E. D. P. Puchta, R. Lucas, F. R. V. Ferreira, H. V. Siqueira, and M. S. Kaster, "Gaussian adaptive PID control optimized via genetic algorithm applied to a step-down DC–DC converter," in *Proc. 12th IEEE Int. Conf. Ind. Appl. (INDUSCON)*, Nov. 2016, pp. 1–6.
[53] H. Siqueira, M. Macedo, Y. D. S. Tadano, T. A. Alves, S. L. Stevan, D. S. Oliveira, M. H. N. Marinho, P. S. G. D. M. Neto, J. F. L. D. Oliveira, I. Luna, M. D. A. L. Filho, L. A. Sarubbo, and A. Converti, "Selection of temporal lags for predicting riverflow series from hydroelectric plants using variable selection methods," *Energies*, vol. 13, no. 16, p. 4236, Aug. 2020.
[54] J. Moore and R. Chapman, "Application of particle swarm to multiobjective optimization," Dept. Comput. Sci. Softw. Eng., Auburn Univ., Auburn, AL, USA, 1999, vol. 32.
[55] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. A. C. Coello, F. Luna, and E. Alba, "SMPSO: A new PSO-based metaheuristic for multi-objective optimization," in *Proc. IEEE Symp. Comput. Intell. Milti-Criteria Decision-Making*, Mar. 2009, pp. 66–73.
[56] C. A. Coello Coello and M. Reyes-Sierra, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," *Int. J. Comput. Intell. Res.*, vol. 2, no. 3, pp. 287–308, 2006.
[57] M. Clerc and J. Kennedy, "The particle swarm—Explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.
[58] R. Gharari, N. Poursalehi, M. Abbasi, and M. Aghaie, "Implementation of strength Pareto evolutionary algorithm II in the multiobjective burnable poison placement optimization of KWU pressurized water reactor," *Nucl. Eng. Technol.*, vol. 48, no. 5, pp. 1126–1139, Oct. 2016.
[59] J. J. Durillo and A. J. Nebro, "JMetal: A Java framework for multi-objective optimization," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, Oct. 2011.
[60] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*, 1st ed. New York, NY, USA: Springer, 2010.
[61] Y. Yuan, H. Chen, and M. Jia, "An optimized ad-hoc on-demand multipath distance vector(AOMDV) routing protocol," in *Proc. Asia–Pacific Conf. Commun.*, 2005, pp. 569–573.
[62] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, pp. 234–244, Oct. 1994.
[63] D. B. Johnson, D. A. Maltz, and J. Broach, "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks," *Ad Hoc Netw.*, vol. 5, pp. 139–172, Jan. 2001.
[64] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proc. IEEE Int. Multi Topic Conf., Technol. 21st Century (INMIC)*, Dec. 2001, pp. 62–68.
[65] N. C. V. N. Pereira and R. M. D. Moraes, "Comparative analysis of AODV route recovery mechanisms in wireless ad hoc networks," *IEEE Latin Amer. Trans.*, vol. 8, no. 4, pp. 385–393, Aug. 2010.

CLODOMIR SANTANA (Member, IEEE) received the bachelor's degree in computer engineering from the Polytechnic School, University of Pernambuco, the master's degree in systems engineering from the University of Pernambuco, in 2019, and the Ph.D. degree in computer science from the University of Exeter, U.K. Currently he is with Tadeusz Manteuffel Institute of History, Polish Academy of Sciences. His master's dissertation received an honourable mention from the University of Pernambuco. During the bachelor's degree, he was awarded a scholarship from the Brazilian Coordination of Superior Level Staff Improvement (CAPES) to be a visiting student with the Faculty of Engineering and Applied Science, Memorial University, Canada, for 16 months. His research interests include bio-inspired metaheuristics, robot's swarm, clustering techniques, multi-objective optimization, complex networks, human dynamics, and neural networks.

MARIANA MACEDO received the B.Sc. and M.Sc. degrees in computer engineering from the University of Pernambuco, in 2016 and 2017, respectively, and the Ph.D. degree in computer science from the University of Exeter. Her research interests include swarm intelligence, evolutionary computation, multi and many-objective, binary optimization, complex networks, network science, data mining, and machine learning.

EMILLY ALVES is currently pursuing the bachelor's degree in electrical engineering, with an emphasis to telecommunications with the Polytechnic School of Pernambuco, University of Pernambuco. Her research interests include neural networks, image segmentation, and digital signal processing.

MARCIO T. GUERREIRO received the B.Sc. degree in mechanical engineering from the Universidade Estadual de São—UNESP, Brazil, in 2001, and the M.Sc. degree in computer science from the Federal University of Technology-Paraná (UTFPR), Brazil, in 2021, where he is currently pursuing the Ph.D. degree in industrial engineering. His research interests include artificial intelligence and data science and their applications in engineering and operational problems. Beyond that, he has been trained in agile methods, such as XP, Kanban, and Scrum. Since 2019, he has been a Business Specialist with the Information Technology Department, DAF Caminhões, Brazil. Enthusiasm concerning agile methods is based on project implementation in the real world, delivering high-quality software products to customers and end-users adapting to the environment, customer, and discovery needs during the development phase, increasing the final value added to the business.

HUGO VALADARES SIQUEIRA received the B.Sc. degree in electrical engineering from the State University of São Paulo, in 2006, the master's, Ph.D., and Postdoctoral degrees from the University of Campinas, in 2009, 2013, and 2014, respectively, and the joint Postdoctoral degree from Illinois State University, USA, and the University of Pernambuco, in 2017. He worked on electrical engineering and computer science, acting on the following subjects: neural networks, evolutionary algorithms, immune algorithms, swarm intelligence, time series forecasting, pollutant impact on human health, and clustering tasks. He is currently an Adjunct Professor with the Federal University of Technology-Paraná (UTFPR). He is also an Advisor of Graduate Programs in Electrical Engineering (PPGEE), Computer Science (PPGCC), and Industrial Engineering (PPGEP). He is a Coordinator with the Computational Intelligence and Advanced Control Laboratory (LICON), UTFPR-PG.

ANURADHA GOKHALE (Senior Member, IEEE) received the master's degree in physics-electronics from the College of William & Mary and the Doctorate degree from Iowa State University. She is currently a Distinguished Professor and a Coordinator of the Computer Systems Technology Program, Illinois State University (ISU). She has completed 30 years as a faculty member and has received several college and university research, teaching, and service awards. She was named as the Fulbright Distinguished Chair of STEM+C with the University of Pernambuco, Brazil (2016–2017). She was a Faculty Fellow in Israel and a Fulbright Specialist in cybersecurity with Gujarat Technological University, India, in Summer 2017; and a Visiting Professor with the College of Business, Shandong University, Jinan, China, in Spring 2017, where she focused on data analytics and e-commerce. Her achievements encompass extensively cited refereed publications; groundbreaking externally funded research supported by a continuous stream for 20 years of grants from state and federal agencies, including the National Science Foundation and elevation of the ISU student experience through excellence in teaching, mentorship, and the creation of opportunities for students to get involved in research. The current NSF funded project is in Computing Education for the 21st Century.

CARMELO J. A. BASTOS-FILHO (Senior Member, IEEE) was born in Recife, Brazil, in 1978. He received the B.Sc. degree in electronics engineering and the M.Sc. and Ph.D. degrees in electrical engineering from the Federal University of Pernambuco (UFPE), in 2000, 2003, and 2005, respectively. In 2006, he received the best Brazilian thesis award in electrical engineering. His research interests include the development of protocols and algorithms to manage and to design optical communication networks, development of novel swarm intelligence algorithms and the deployment of swarm intelligence for complex optimizations and clustering problems, development and application of multi-objective optimizations and many-objective optimizations for real-world problems, development and application of soft deep learning techniques, development of solutions using network sciences for big data, and applications in robotics and applications in biomedical problems. He is currently an Associate Professor with the Polytechnic School, University of Pernambuco. He has published roughly 300 full papers in journals and conferences and advised over 50 Ph.D. and M.Sc. candidates. He was the Scientist-in-Chief of the Technological Park for Electronics and Industry 4.0 of Pernambuco, from 2016 to 2020. He is also the Director of Innovation Parks and Graduate Studies of Pernambuco. He is also the Coordinator of the National Initiative for Promoting ICT in Pernambuco jointly with the Ministry for Regional Development. He is a Research fellow level 1D of the National Research Council of Brazil (CNPq). For more information visit the link (http://scholar.google.com/citations?user=t3A96agAAAAJ&hl=en).