Received August 23, 2019, accepted September 14, 2019, date of publication September 24, 2019,
date of current version November 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943546

An Efficient Clustering Strategy Avoiding Buffer Overflow in IoT Sensors: A Bio-Inspired Based Approach

RANIDA HAMIDOUCHE[1,2], ZIBOUDA ALIOUAT[1], ADO ADAMOU ABBA ARI[2,3], AND MOURAD GUEROUI[2]

[1]LRSD Lab, Department of Computer Science, University Ferhat Abbes Setif 1, Setif 19000, Algeria
[2]LI-PaRAD Lab, Université Paris Saclay, University of Versailles Saint-Quentin-en-Yvelines, 78000 Versailles, France
[3]LaRI Lab, University of Maroua, Maroua, Cameroon

Corresponding author: Ranida Hamidouche (ranida.hamidouche@univ-setif.dz)

This research work was supported by PHC-Tassili, Grant Number 18MDU114.

ABSTRACT The Internet of Things (IoT) invention has taken the growth of sensors technology to a completely high step. New challenges in terms of data delivery have emerged due to strict QoS conditions. Among the solutions proposed in the literature is the subdivision of the large-scale network into several clusters. Except that most of these solutions are conventional. However, prior research generally confirms that bio-inspired paradigms are more flexible and effective compared to traditional methods. When it comes to a heterogeneous network, additional constraints appear. Nodes have different buffer sizes. Then, data captured must be sent before their buffers are full, otherwise, some data will be lost. This is not suitable for a real-time application where time and information are crucial elements. In this study, a comprehensive overview of the use of sensors in IoT contexts is performed. Two algorithms as Grey Wolf Optimizer (GWO) and Whale Optimization Algorithm (WOA), combined with the Imperialist Competitive Algorithm (ICA) based Cluster Head (CH) selection with a novel approach for heterogeneous networks are proposed. These algorithms can support data exchange over a heterogeneous Wireless Sensor Network (WSN) infrastructure with taking into consideration the buffer overflow problem. Simulation results are presented and discussed in different network designs. The research demonstrated that knowing well how to manage buffers using bio-inspired techniques, leads to a significant reduction in data loss.

INDEX TERMS Bio-inspired optimization, heterogeneous WSN, CH selection, IoT, model checking.

I. INTRODUCTION

Nowadays, the world population is increasing in urban centers and the needs in terms of Information Communication Technologies (ICT) services, as well as networking infrastructures, are rapidly evolving [1]. To achieve these goals, thousands of independent smart devices including smartphones, sensors, actuators, RFID, measurement devices and computers, which are part of our daily life, are put together to provide a smart interaction in a self-orchestration and self-organization way to form the future Internet. Moreover, all these heterogeneous devices are capable of data generation as well as data delivery, in which a number of techniques and schemes are involved in data gathering, transmission, and storage [2]. Such kind of interaction of smart objects is known as the IoT.

The IoT paradigm can be seen as dynamic networking in which smart devices and a number of virtual entities, say, things, which are autonomous, self-configurable and capable of auto-organization, interact themselves and with the environment to achieve common goals [3]. Furthermore, IoT becomes a worldwide Machine to Machine (M2M) communications tool in which interconnected objects smoothly communicate to exchange a huge amount of data generated by sensing devices and react with events from the environment by triggering a number of actions intended to realize the desired goal [4]. Moreover, heterogeneous and widespread sensing and actuating networks such as WSNs that have

The associate editor coordinating the review of this manuscript and approving it for publication was Giacomo Verticale.

outstanding potential in a wide range of applications and industrial systems, act as the common sensing platforms for IoT [5]–[7].

The evolution of microelectronics in the last decades allowed the development of low cost and low power sensors, which can be deployed on a large-scale environment to collectively gather information from the deployed field and wirelessly deliver the gathered data to a remote computer for computing and storage purposes [5], [8]–[10]. This cycle is summarized in figure 1.

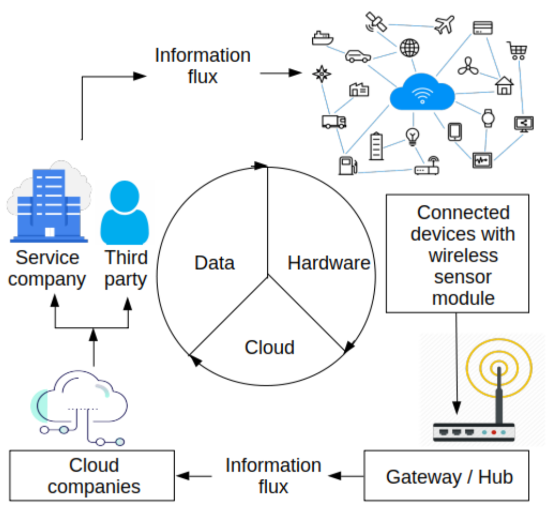


FIGURE 1. IoT global market structure.

These networks play an important role in a number of applications including precision agriculture, military surveillance, forest fire, and e-health. Besides, WSNs have evolved from sensing and actuating networks to an essential system in the IoT, especially for data delivery. However, in spite of the fact that many data delivery schemes and algorithms have been among the first most addressed issue in WSNs, reliable delivery of data originated from sensors deployed on sometimes hostile environments remains an actual issue. That is why several requirements must be taken into consideration when designing the hardware and software of the sensors [2].

One of the methods to improve sensors capacities is using topology management schemes [11]. These schemes consist of three steps: topology discovery, sleep round management and clustering. We are much more interested in this last method. It can be centralized or distributed. Both methods are suitable for smaller sensor nodes [12]. However, centralized approaches are not well suited for a large-scale network. Moreover, nodes near the central authority will deplete their energy faster as a huge amount of data transfer takes place between nodes and Base Station (BS). The size of clusters near the BS should be small so that energy can be conserved [13]. Contrariwise, distributed approaches are considered more efficient as the amount of information transferred that takes place between BS and nodes are reduced and as nodes themselves take the decision regarding clustering approach [13]. Authors in [14], proposed an asynchronous distributed clustering to treat buffer overflow problem at CH level. Some other important issues than energy consumption should be taken into account when designing an optimal hierarchical WSN topology, such as connectivity, coverage [15] and data fusion [16]. Therefore, minimizing or maximizing these parameters leads the topology design to a discrete NP-hard problem. This type of problems could not be solved in polynomial time using conventional methods.

Evolutionary algorithms could be an efficient alternative capable to find optimal solutions for most NP-hard problems [17]. This methods have proven their effectiveness in solving complex optimization problems. Meta-heuristic optimization methods have become extremely popular over the past two decades because of their simplicity, flexibility, and local minima avoidance. The inspiration source for these techniques has simple concepts. It mostly inspired by animals' behavior [18], physical phenomena, and evolutionary concepts. This simplicity attracts the researchers to develop and propose new meta-heuristics.

This paper presents a new clustering method in heterogeneous WSN, to deal with the problem of the buffer overflow, with better throughput and high availability. It combines the strength of ICA with GWO and WOA. The proposed protocol takes into account a new characteristic: buffer size of the sensor. To form the groups, the protocol imitates the hierarchical process of the grey wolves and the hunting behavior of whales in real life, which introduces an autonomous and intelligent cluster formation. Our protocol can be a good solution to be applied on e-health. This kind of application uses a particular type of sensors named Wireless Body Area Sensor (WBAN). Unlike typical WSN, these sensors can be very limited in terms of power availability and processing strength and hence preserving the energy of the nodes is of great importance. Additionally, in order to minimize interference and to cope with health concerns such as avoiding tissue heating of skin on patients, an extremely low transmit power per node is required. Energy is not the only factor that must be considered in this type of application, there are other factors such as topology, temperature, posture, radio range of sensors, appropriate quality of service and data security. These two last points are very complementary. Ensuring patients information security and data user privacy over the wireless networks require the use of complex encryption algorithms. The Reliability Dilemma is particularly important in that case, to be highly reliable, high overheads in terms of data size, power consumption and scalability are needed. Our protocol manages to find a compromise between these three points at the same time. For that case, the protocol parameters must be set in order to be applied to the WBAN. The number of sensors expected to be placed in the network is up to 50 and at maximum 256 as defined in IEEE 802.15.6. In addition, due to the short communication range in WBAN, the radius of the node must be fixed to 3 meters. The heterogeneity of sensors in terms of available energy, computing power, and buffer size are taken into consideration in our solution.

The rest of this paper is organized as follows: An overview of ICA, GWO, and WOA is given in section 2. Some preliminaries about energy and network model are outlined in section 3. The hybridized solution for the buffer overflow problem on normal nodes is discussed in section 4. An illustrative case is presented in section 5. The protocol model and its verification are highlighted in section 6. The experimental results are presented in section 7 followed by the conclusion.

II. LITERATURE SURVEY

Many algorithms for optimization problems have been proposed in the past few years. Metaheuristic ones proved that they are the most effective for this field. In what follows, we will discuss a few of these methods.

A. IMPERIALIST COMPETITIVE ALGORITHM

On 2007, Atashpaz and Lucas [28] proposed the ICA. The algorithm puts control over many countries and uses their sources once colonies are dominated by rules. Differently from the working steps of most meta-heuristic algorithms inspired by an evolutionary phenomenon based on nature, this process is called the socio-political process of imperialism [29]. Several types of research have been conducted in the context of using ICA in WSN clustering. The first application was made by Marjan and Mansoureh [11] in 2014. Authors enhanced the well-known clustering algorithm LEACH using this algorithm. In the same year, Moslem [19] used it to choose the best sensor in the cluster as a CH. To increase the network lifetime, the proposed algorithm uses different parameters. These parameters are based only on energy: the residual energy in the CH required energy to send a message toward the sink node and the required energy for receiving a k-bit message by the CH and send it to sink. Chaitra and Ravikumar [20] have also proposed a new and efficient cluster model named ICACO (ICA Cluster Optimization). A new fixed-clustering algorithm named ICA-Clustering was introduced by Rostami *et al.* [21]. In this protocol, firstly, nodes send their location information to the BS. Then, this latter form k_{opt} clusters using this information as input parameters for the ICA algorithm. Authors aimed to find the k_{opt} point as the center of the cluster. The last publication found in this context was proposed by Hosseinirad [17] in 2018. The author proposed a novel dynamic Multi-layer Clustering Topology (MCT) without taking into account sleep schedule.

According to Hosseinia and Khaled [30], ICA is compatible with different kinds of optimization problems, even that of WSN. Moreover, a notable feature of ICA is the ease of combining the algorithm with other algorithms which may result in better solutions. In addition, ICA is established based on a systematical mathematical calculation which makes it easier for researchers to investigate its convergence and robustness. Furthermore, ICA has a reasonable computational time. On the other hand, ICA has some unavoidable shortcomings. It does not guarantee an optimal solution because it does not has a theoretical convergent property. It may result in premature convergence. Another issue that deserves attention is that ICA has more parameters in comparison to Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), which leads to not being able to adjust them, especially the deviation parameter of the assimilation process.

B. GREY WOLF OPTIMIZER

On 2014, Mirjalili *et al.* [31] proposed the GWO algorithm. This method mimics the hunting and leadership hierarchy process of grey wolves in nature. Kalpana [23] proposed a GWO to minimize the energy dissipation between nodes. They used only packet delivery ratio and energy as GWO parameters. Al-Aboody and Al-Raweshidy [22] had the same objectives. However, in this time, authors proposed a Multi-Level Hybrid energy-efficient clustering routing Protocol named MLHP instead of using GWO for clustering. Diwan and Khan [26] combined the GWO with fuzzy logic to achieve energy efficiency. This latter was used for cluster formation and the GWO was used for CH election. In 2017, Sharawi and Emary [24] introduced another idea. Authors used GWO to optimize the formation process of each cluster by minimizing the distances in intra-cluster. The latest works found were those of Jabinian *et al.* [25] and Zhao *et al.* [27]. The first team implemented an energy-optimized GWO method for WSN data communication with the aim of finding the desired energy consumption conditions. The second one used a Fitness value based Improved GWO (FIGWO) as the weights of the GWO algorithm to determine the final position of the optimal CH. The algorithm fully takes the nodes current state into consideration. However, the fitness used to combine only two parameters: the distance to the BS and the residual energy of the node.

GWO has caused much attention due to its simpleness, implementation ease, and fewer control parameters, only two main parameters to be adjusted (a and C) [32]. It can also be generalized to large-scale issues [25]. But the success of this meta-heuristic algorithm depends upon the equilibrium between exploration and exploitation. It has a special capability to strike the right balance between them during the search which leads to favorable convergence. GWO has one vector of position, so it requires less memory. Contrary to other population-based heuristics, GWO saves three best solutions [33]–[35].

C. WHALE OPTIMIZATION ALGORITHM

WOA is an algorithm inspired by the killer whale life introduced by Mirjalili and Lewis [36]. It is based on the hunting behavior of humpback whales. Humpback whales have a unique hunting method called bubble-net feeding method. It usually involves creating bubbles along a circle around the prey while hovering around the prey. Usually, there are two maneuvers associated with this hunting technique. First one is upward-spirals, where the whale dives 12m down and creates bubbles in spiral shape while swimming towards the surface and the other one is more complex and has three stages namely, lobtail, capture loop, and coral loop. This unique spiral bubble-net hunting behavior can only be seen

TABLE 1. Comparative table for ICA clustering.

Authors	Protocol name	Clustering scheme	Heterogeneity Aware	Cluster balance	Coverage	Latency	Energy efficiency	Throughput	Benchmarks
Marjan and Mansoureh (2014) [11]	ICA-LEACH	Distributed	No	/	/	/	Yes	/	LEACH
Moslem (2014) [19]	CHEI	Distributed	No	Yes	/	Yes	Yes	/	LEACH / Genetic
Chaitra and Ravikumar (2015) [20]	ICACO	Distributed	No	/	/	/	Yes	Yes	LEACH
Rostami et al. (2017) [21] []	ICA-Clustering	Centralized	No	Yes	/	Yes	Yes	/	Itself / LEACH / SEP / DEEC
Hosseinirad (2018) [17]	MCT	Distributed	No	/	Yes	/	Yes	/	WEEC / LEACH-ICA

TABLE 2. Comparative table for GWO clustering.

Authors	Protocol name	Clustering scheme	Heterogeneity Aware	Cluster balance	Coverage	Latency	Energy efficiency	Throughput	Benchmarks
Al-Aboody and Al-Raweshidy (2016) [22]	MLHP	Centralized Distributed	Yes	/	/	/	Yes	Yes	LEACH / DEEC / SEP
RajaRajeswari and Kalpana (2016) [23]	GWO	Distributed	No	/	/	/	Yes	Yes	AODV / BeeSensor
Sharawi and Emary (2017) [24]	GWO	Distributed	No	/	Yes	/	Yes	Yes	LEACH
Jabinian et al. (2018) [25]	PGWO	Distributed	No	Yes	/	/	Yes	/	GA
Diwan and Khan (2016) [26]	Fuzzy-GWO	Distributed	Yes	/	/	/	Yes	Yes	LEACH
Zhao et al. (2018) [27]	FIGOW	Distributed	No	Yes	/	/	Yes	Yes	SEP / LEACH

in humpback whales [37]. Jadhav and Shankar [38] proposed an algorithm called WOA-Clustering (WOA-C) for homogeneous networks. Authors used a fitness function which considers only two parameters: the residual energy of the node and the sum of energy of adjacent nodes.

III. PRELIMINARIES

A. ENERGY MODEL

The energy model used is as in [39], where the energy consumed to send a l-bit message over distance d is:

$$E_{T_x}(l,d) = \begin{cases} l \times E_{elec} + l \times \epsilon_{fs} \times d^2 & \text{if } d < d_0 \\ l \times E_{elec} + l \times \epsilon_{mp} \times d^4 & \text{if } d > d_0 \end{cases} \quad (1)$$

where E_{T_x} is the transmitted energy, E_{elec} is the energy dissipated per bit in the transmitter or receiver circuit. ϵ_{fs} and ϵ_{mp} depend on the transmitter amplifier model. If the distance between the transmitter and the receiver is less than a threshold d_0, the free space model is used; otherwise, the multi-path model is used. d_0 is usually calculated as:

$$d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \quad (2)$$

On the other side, the energy consumption for the receiver to receive a l-bit long packet is calculated as follows:

$$E_{R_x} = l \times E_{elec} \quad (3)$$

B. NETWORK MODEL

In this article, several WSN scenarios are used. A different number of heterogeneous sensor nodes are distributed in the network area. There are m_1 nodes equipped with α times more energy and buffer size acting as advanced nodes and m_2 nodes equipped with β times more energy and buffer size acting as super nodes. The total number of nodes (N) can be calculated as in [22]:

$$N = \Big(1 - \big(m_1 + m_2\big)\Big) \times N + m_1 \times N + m_2 \times N \quad (4)$$

In the network, the selected nodes are selected from advanced nodes according to the proposed GWO and WOA based CH selection algorithms. The remaining nodes in the sensor network have normal node property. Here, it is important to note that, in the proposed algorithms, each CH is an advanced node, but not every advanced node can be assigned a CH.

C. SYSTEM MODEL AND ASSUMPTIONS

The network model is considered to be a free space model. It has a transmitter and a receiver with a distance of separation d. The amplifier circuits are also present at both T_x and R_x. The WSN scenario considered for simulation has all the following properties and limitations to form the system model:

- All sensor nodes are distributed using a Poisson homogeneous distribution.
- All nodes are heterogeneous and have limited buffer size.
- The BS is stationary and it can be located inside or outside the sensing area.
- Data fusion is used to minimize the total amount of forwarded data.
- All deployed nodes are static, which means no node is able to change its own location once the distribution process is done.
- Each node has a fixed communication range R_{Tx}.
- The node with maximum energy, the maximum number of neighbors, maximum buffer size and which is located around the center of gravity of the cluster is the most suitable candidate for the role of CH.
- Other nodes not satisfying that criteria have a low probability or no chance to become a CH.
- If two nodes have the same odds and are in the neighborhood, one of them will be shut off at that point to help extend the lifetime of the network.
- If the distance between a node and its corresponding CH is greater than its distance to the BS, this node will send its sensed information to the BS directly.
- It should be noted that as the number of parameters increases, consequently, calculations, time, and energy consumption increase.

TABLE 3. Similarity between GWO and WSN.

GWO	WSN
Grey wolf	Sensor nodes
Food placement (Prey)	Objective function
The position of the prey in the region is not known by grey wolves.	The nodes do not know the cost function in advance. The nodes can only detect changes in the values of the cost function.
Best value of fitness function	CH node
The average number of wolves per herd is between 5 and 12.	The ideal number of members in a cluster is lower than 12 and higher than 5.
Positions of wolves	Nodes positions $(pos(t) = x(t), y(t))$

IV. HYBRIDIZED SOLUTION FOR THE BUFFER OVERFLOW PROBLEM ON NORMAL NODES

Over the last years, many research results have been published in the field of evolutionary algorithms. The results show that the hybridizations of several metaheuristics techniques are exceptionally successful. However, in most of the papers that have been reviewed, each heuristic is implemented without conjunction with other heuristic algorithms. The ICA can be hybridized with some well-known algorithms such as GWO and WOA. This hybridization can be implemented in different ways. For example, GWO or WOA can be employed as a starting point to generate a good initial solution and the rest of the search can be taken care of by the ICA. In some cases, ICA can be used as a starting point to generate an initial solution and others can be used to conduct the search. Furthermore, GWO and WOA may be hybridized into the ICA as a complementary tool to promote the capability of exploitation and produce high-quality solutions. GWO or WOA can be employed to select the CHs and ICA to form clusters. In this paper, the last hypothesis is adopted.

A. GWO MODEL

The leadership hierarchy used in grey wolves' colonies employs four sorts of wolves: alpha, beta, delta, and omega [32]. Alpha is the leader. It is responsible for making decisions. It is also called the dominant wolf since it orders should be followed by the pack. Alpha is the best member to manage the pack. But it is not automatically the strongest wolf. This shows that strength is not much important than discipline and organization of the pack. Beta wolves are in the second level of the hierarchy. The betas help alpha wolves to make decisions but should respect them. Betas command on wolves of the lower level. In the case where alpha passes away, betas are the appropriate candidates to replace alpha. Omega is the lowest ranking grey wolf. This category of wolves must apply all the decisions of other dominant wolves. If a wolf is not an alpha, beta, or omega, it is called delta (or subordinate in some references). Wolves of this level dominate omegas but have to submit to alphas and betas. Grey wolves have another interesting social behavior in addition to the social hierarchy. This behavior is group hunting. The main phases of this technique are as follows: searching for prey, encircling prey and attacking prey [33]. The aforementioned are detailed in what follow.

In this study, the selection of CH in the heterogeneous cluster network was inspired by the structure of the grey wolf. Table 3 presents the similarity and comparison between the grey wolf and the WSN.

B. THE PROPOSED CH SELECTION ALGORITHM USING GWO

In this subsection, we will explain how to select the CHs using GWO.

1) SEARCH FOR PREY (EXPLORATION)

In accordance with the alpha, beta, and delta positions, grey wolves mostly search for the prey. They converge from each other to attack prey and diverge to search for prey. In order to mathematically model divergence, we utilize A with random values greater than 1 or less than -1 to oblige the search agent to diverge from the prey. The value of this parameter is controlled by a, which linearly decreases from 2 to 0. C is another parameter in GWO that favors exploration. It contains random values in [0; 2]. This contribution is strong when $C < 1$, the solution gravitates more towards the prey, favoring exploration and local optima avoidance [33].

2) ENCIRCLING PREY

As mentioned above, the prey is encircled during the hunt by grey wolves. To mathematically model this behavior, the following equations are proposed:

$$X(t+1) = X_p(t) - A \times D \tag{5}$$

where $X(t+1)$ is the next location of the wolf, $X(t)$ is current location, A is a coefficient matrix and D is a vector that depends on the location of the prey X_p and is calculated as follows:

$$D = \left| C \times X_p(t) - X(t) \right| \tag{6}$$

where

$$C = 2 \times r_2 \tag{7}$$

Note that r_2 is a randomly generated vector from the interval [0, 1]. With these two equations, a solution is able to relocate around another solution. Note that the equations use vectors, so this is applied to any number of dimensions. Note that the random components in the above equations simulate different step sizes and movement speeds of grey wolves. The equations to define their values are as follows:

$$A = 2 \times a \times r_1 - a \tag{8}$$

where a is a vector where its values are linearly decreased from 2 to 0 during the course of the run. r_1 is a randomly generated vector from the interval [0, 1]. The equation to update the parameter a is as follows:

$$a = 2 - t \times \left(\frac{2}{T}\right) \tag{9}$$

where t shows the current iteration and T is the maximum number of iterations [33].

A modified version of GWO was proposed by Mittal et al. [40] to enjoy the better exploration. Instead of decreasing the value of a linearly, authors used an exponential function as given:

$$a = 2 \times \left(1 - \left(\frac{t^2}{T^2}\right)\right) \tag{10}$$

3) HUNTING (OPTIMIZATION)

In GWO, it is obvious that alpha, beta, and delta are always the three best solutions obtained so far. The global optimum of optimization problems is unknown, so it has been assumed that alpha, beta, and delta have a good idea of its location, which is reasonable because they are the best solutions in the entire population [33]. Therefore, other wolves should be obliged to update their positions. For the first round, the positions are updated as follows:

$$\begin{aligned} X_1 &= X_{alpha}(t) - A_1 \times D_{alpha} \\ X_2 &= X_{beta}(t) - A_2 \times D_{beta} \\ X_3 &= X_{delta}(t) - A_3 \times D_{delta} \end{aligned} \tag{11}$$

where D_{alpha}, D_{beta} and D_{delta} are calculated using formula 12 as follows:

$$\begin{aligned} D_{alpha} &= |C_1 \times X_{alpha} - X| \\ D_{beta} &= |C_2 \times X_{beta} - X| \\ D_{delta} &= |C_3 \times X_{delta} - X| \end{aligned} \tag{12}$$

$X(t+1)$ is obtained using formula 13.

$$X(t+1) = \frac{1}{3}X_1 + \frac{1}{3}X_2 + \frac{1}{3}X_3 \tag{13}$$

For the other rounds, positions are updated using one of these cases as follows:

- case 1:

$$\begin{aligned} D_{alpha} &= |C_1 \times Fitness_{alpha} - Fitness_X| \\ D_{beta} &= |C_2 \times Fitness_{beta} - Fitness_X| \\ D_{delta} &= |C_3 \times Fitness_{delta} - Fitness_X| \\ X_1 &= Fitness_{alpha} - A_1 \times D_{alpha} \\ X_2 &= Fitness_{beta} - A_2 \times D_{beta} \\ X_3 &= Fitness_{delta} - A_3 \times D_{delta} \\ X(t+1) &= \frac{1}{3}X_1 + \frac{1}{3}X_2 + \frac{1}{3}X_3 \end{aligned} \tag{14}$$

- case 2:

$$\begin{aligned} X_1 &= |Fitness_{alpha} - Fitness_X| \\ X_2 &= |Fitness_{beta} - Fitness_X| \\ X_3 &= |Fitness_{delta} - Fitness_X| \\ X(t+1) &= \frac{1}{3}X_1 + \frac{1}{3}X_2 + \frac{1}{3}X_3 \end{aligned} \tag{15}$$

- case 3:

A modified version of GWO was proposed in [41]. Authors aimed to have faster convergence by updating the position of wolves based on incorporating a step size that is proportional to the fitness of the individual in the search space in the current generation as given:

$$X(t+1) = \frac{1}{iteration}^{\frac{Fitness_{alpha} - Fitness_X}{Fitness_{alpha} - Fitness_{worst}}} \tag{16}$$

- case 4:

$$X(t+1) = \left|\frac{Fitness_{alpha} - Fitness_X}{Fitness_{alpha} - Fitness_{worst}}\right| \tag{17}$$

4) ATTACKING PREY (EXPLOITATION)

As mentioned above the grey wolves finish the hunt by attacking the prey when it stops moving. Exploitation is promoted when $-1 < A < 1$. As mentioned above, a good balance between exploitation and exploration is required to find an accurate approximation of the global optimum using stochastic algorithms. This balance is done in GWO with the decreasing behavior of the parameter a in the equation for the parameter A [33]. With decreasing A, half of the search is dedicated to exploration ($A \geq 1$) and the other half is devoted to exploitation ($A < 1$), as it is shown in figure 2.

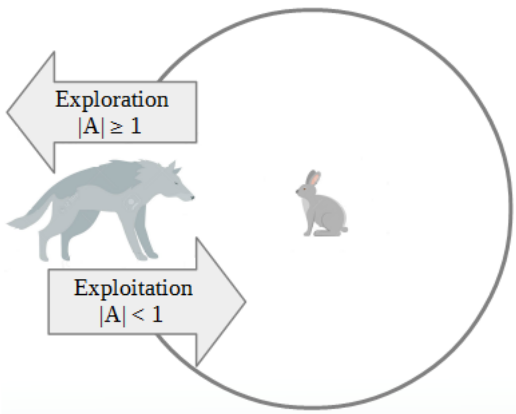


FIGURE 2. Exploration versus exploitation periods depending on the value of A in GWO.

5) TERMINATION CONDITION

It depends on three modes: if the number of iterations is specified, convergence to an optimal answer when it is not aware of the optimal value, no change after a certain number of repetitions [25]. In this paper, the termination condition is done by the number of iterations.

Our goal is to maximize the fitness function, choosing the node with more energy, more memory space, and more neighborhood. The Fitness function is mentioned above:

$$fitness(i) = coef_1 \times NumberOfNeighbors + coef_2 \times CurrentBatteryPower(i) + coef_3 \times Capacity(i) + Suitability(i) \quad (18)$$

where:

$$coef_1 = 0.2 \quad coef_2 = 0.3 \quad coef_3 = 0.5$$

These coefficients are chosen by priority. That's mean that we favor more the memory space of the node ($coef_3$), then the residual energy ($coef_2$), then the neighborhood ($coef_1$). This is to solve the buffer overflow problem and minimize the loss of data as much as possible.

In what follows, we will explain the parameters of the fitness function in detail. The node's location is calculated as in formula 19. Where *DistMaxToBS* and *DistMinToBS* are formulated from the equations 20 and 21 respectively. Coordinates $(X_{min}, Y_{min}) = (100, 100)$ and $(X_{max}, Y_{max}) = (0, 0)$ represent the closest and the farthest points from BS respectively.

$$Location(i) = \begin{cases} if\ i\ != farthest\ and\ closest\ node\ from\ BS \\ \dfrac{DistMaxToBS - Dist(i)ToBS}{DistMaxToBS - DistMinToBS} \\ other\ situations \quad 1 \end{cases} \quad (19)$$

$$DistMaxToBS = \max\left(\sqrt{\left(X_{min} - X_{BS}\right)^2 + \left(Y_{min} - Y_{BS}\right)^2}\right) \quad (20)$$

$$DistMinToBS = \min\left(\sqrt{\left(X_{max} - X_{BS}\right)^2 + \left(Y_{max} - Y_{BS}\right)^2}\right) \quad (21)$$

To calculate the neighborhood of the node, we consider what we call the radius. Using uniform clustering strategy in a randomly arranged network can create an unbalanced network structure [42]. The radius used to find a node's neighbor is calculated using the following equation.

$$R = \sqrt{\frac{A}{\pi \times k}} \quad (22)$$

where $A = M \times M$ is the area of the network and k is the optimum number of clusters. We can find k as following.

All nodes are located in the area S with $(d < d_0)$. Therefore, the energy dissipation to transmit an l-bit message in the CH is:

$$\begin{aligned} E_{CH} &= E_{Tx} + E_{Rx} + E_{DA} \\ &= \left(l \times E_{elec} + l \times \epsilon_{fs} \times d^2_{ToBS}\right) \\ &\quad + \left(\left(\frac{N}{k} - 1\right) \times l \times E_{elec}\right) \\ &\quad + \left(\frac{N}{k} \times E_{DA}\right) \end{aligned} \quad (23)$$

where d_{ToBS} is the distance from the CH node to the BS. The energy dissipated from the other nodes is calculated as follows:

$$E_{nonCH} = T_{Tx} = l \times E_{elec} + l \times \epsilon_{fs} \times d^2_{ToCH} \quad (24)$$

The area of sensing is formulated by several clusters which have CHs as a center. It is calculated as follows:

$$S = \left(2 \times \pi \times d^2_{ToCH}\right) \times k \simeq A \quad (25)$$

From equation 25, we deduce the distance separating a node from the CH:

$$d^2_{ToCH} = \frac{A}{2 \times \pi \times k} \quad (26)$$

So :

$$E_{nonCH} = l \times E_{elec} + l \times \epsilon_{fs} \times \frac{A}{2 \times \pi \times k} \quad (27)$$

Now, the energy dissipated in a cluster in one frame is:

$$E_{Cluster} = l \times \left(E_{CH} + \frac{N}{k} \times E_{nonCH}\right) \quad (28)$$

and the total energy is:

$$E_{Total} = k \times E_{Cluster} = k \times l \times \left(E_{CH} + \frac{N}{k} \times E_{nonCH}\right) \quad (29)$$

To find k, the number of clusters formed in a round, we equate equation 29 to zero and differentiate with respect to k:

$$\begin{aligned} k &= \left(\frac{\sqrt{N \times A}}{\sqrt{2 \times \pi}} \times \frac{1}{d_{toBS}}\right) \times 100 \\ &= \frac{\sqrt{(1 - (m_1 + m_2)) \times N + m_1 \times N + m_2 \times N \times A}}{\sqrt{2 \times \pi}} \\ &\quad \times \frac{100}{d_{toBS}} \end{aligned} \quad (30)$$

where N is the number of sensor nodes calculated from equation 4 and d_{toBS} is the average distance from a node to BS is giving by [43]:

$$d_{ToBS} = 0.765 \times \frac{A}{2} \quad (31)$$

According to [44], an increasing number of clusters leads to small size cluster distribution, which is better in term of energy consumption. The fixed cluster count increases the stability of a sensor network.

Since our goal is to maximize the fitness function, to calculate the memory capacity of a node, we use the fraction between the Initial Buffer Size (IBS) and the remaining space of the buffer.

$$Capacity(i) = \frac{IBS(i)}{IBS(i) - CurrentBufferSize(i)} \quad (32)$$

The energy of the node is calculated using the fraction between the initial energy of the node IBP and the remaining energy as in equation 33.

$$Energy(i) = \frac{IBP(i)}{IBP(i) - CurrentBatteryPower(i)} \quad (33)$$

TABLE 4. Similarity between whales' life and WSN.

WOA	WSN
Whale	Sensor nodes
Food placement (Prey)	Objective function
The position of the prey in the region is not known by whales.	The nodes do not know the cost function in advance. The nodes can only detect changes in the values of the cost function.
Best value of fitness function	CH node
Positions of whales	Nodes positions $(pos(t) = x(t), y(t))$

The last parameter of the fitness function is calculated as in the formula 34.

$$Suitability = \frac{CurrentBatteryPower(i)}{Energy(i) \times Location(i)} \tag{34}$$

Algorithm 1 Pseudo Code of CH Election Using GWO

Require: Network of N nodes
Ensure: CH nodes
- Initialize the grey wolf population $X_i (i = 1, 2, \ldots, n)$
- Initialize a, A and C
- Calculate the *fitness* of each agent
- Choose three best solutions X_{alpha}, X_{beta} and X_{delta}
while *iteration < max number of iterations* **do**
  for each search agent **do**
    - Update the position of the current search agent
    if round = 0 **then**
      Use formulas 11, 12 and 13
    else
      Use cases 1, 2, 3 or 4
    end if
  end for
  - Update a, A and C
  - Calculate *fitness* of all search agents
  - Update X_{alpha}, X_{beta} and X_{delta}
  - iteration = iteration + 1
end while
return CHs

C. WOA MODEL

WOA algorithm starts with a randomly generated population of whales (solutions) each with a random position. In the first iteration, the search agents update their positions in the reference to a randomly chosen search agent. However, from the second iteration onwards the search agents update their positions with respect to the best solution obtained so far. A random search agent is chosen if the value of $|A| > 1$, this helps in exploration. When the best solution is selected, $|A|$ is set to $|A| < 1$. This induces exploitation as all the search agents will converge. The hunting behavior can be explained in 3 phases: searching, encircling, and attacking the prey. The two first steps do not differ from GWO algorithm. The difference is only in how to attack the prey.

In this study, the selection of CH in the heterogeneous cluster network was inspired by the structure of the whales. Table 4 presents the similarity and comparison between the whales and the WSN.

D. THE PROPOSED CH SELECTION ALGORITHM USING WOA

In this subsection, we will explain how to select the CHs using WOA.

1) ATTACKING PREY (EXPLOITATION)

The attacking behavior is modeled with respect to the bubble net attacking strategy, shrinking encircling mechanism and spiral updating position mechanism. Humpback whales apply these two mechanisms with a probability of 50% for each mechanism. A random variable p is introduced where p varies between [0, 1]. The updating model can be given by this equation.

$$X(t+1) = \begin{cases} X'(t) - A \times D & \text{if } p < 0.5 \\ D \times e^{bl} \times \cos(2\pi l) + X'(t) & \text{if } p \geq 0.5 \end{cases} \tag{35}$$

where D indicates the best solution so far (the distance of the i^{th} whale to the prey), l is a random number in the range $[-1, 1]$, b is a constant value that defines the logarithmic spiral[1] ($b = 0.1759 = \frac{\ln \epsilon}{\pi}$ for Nautilus shell).

E. ICA FOR CLUSTER MEMBERS JOIN

Nodes consider BS as the next hop if it is in communication range, otherwise, they will select the CH using ICA. To create initial empires, ICA generates a set of random solutions in the search space of the optimization problem. This initial population is called country. To determine the power of each country, a cost function is used. Based on the value of the cost function, the best initial countries become Imperialists that take control of the other countries called colonies. Imperialists and colonies form what we call an Empire. Assimilation and Revolution are the two main operators of this algorithm. Assimilation is used to approach the imperialist state. Revolution changes the position of some countries randomly to explore more the search space. A better position might be reached by a colony during these two processes. At this stage, It will control the entire empire and change the present imperialist state. During the competition process, powerful empires take its possession of their colonies and

[1] A logarithmic spiral is a kind of spiral seen in the natural world. Several examples are found in the shells of some mollusks, such as that of spider webs and the fossil ammonites and also in the nautilus.

TABLE 5. Similarity between ICA and WSN.

ICA	WSN
Countries	Sensor nodes
Colonies	Normal nodes (non-CH)
Empire	Cluster
Assimilation	CH selection algorithm
Power of each country	Objective function
Imperialist (Best country) takes control of the entire empire	CH plays the role of the leader in relation to its members
During the competition, weak empires collapse and powerful ones take possession of their colonies.	CHs with a low value of fitness function will let their positions to other nodes.

Algorithm 2 Pseudo Code of CH Election Using WOA

Require: Network of N nodes
Ensure: CH nodes
- Initialize the whales population $X_i (i = 1, 2, \ldots, n)$
- Calculate the *fitness* of each agent
- Choose the best search agent X'
while *iteration* $<$ *max number of iterations* **do**
  for each search agent **do**
    - Update a, A, C, l and p
    - Update the position of the current search agent
    if round = 0 **then**
      if $p < 0.5$ **then**
        if $|A| < 1$ **then**
          - Use $X(t+1) = X'(t) - A \times D$
        else
          - Select a random search agent X_{rand}
          - Use $X(t+1) = X_{rand} - A \times D$
        end if
      else
        - Use $X(t+1) = D \times e^{bl} \times \cos(2\pi l) + X'(t)$
      end if
    else
      Use cases 1, 2, 3 or 4
    end if
  end for
  - Check if any search agent goes beyond the search space and amend it
  - Calculate the fitness of each search agent
  - Update X' if there is a better solution
  - iteration = iteration + 1
end while
return X' (CH is the nearest node to the position of X')

weak one's collapse. The algorithm stops until a condition is satisfied [11], [19], [20], [45], [46].

In this study, the cluster formation in the heterogeneous network was inspired by the socio-political process of imperialism of controlling countries. Table 5 presents the similarity and comparison between the ICA and the WSN.

F. THE PROPOSED CLUSTER JOIN USING ICA

In this subsection, we will explain how nodes join the best CH using ICA.

1) GENERATING INITIAL EMPIRES

The country is represented by the nodes. Best countries are called imperialists and the worst ones are colonies. The cost of a country is found by evaluating several parameters $(p_1, p_2, p_3, \ldots, p_n)$ at the node. In our case, we used the fitness function described in 18.

$$cost = f(country) = f(p_1, p_2, p_3, \ldots, p_n) \tag{36}$$

To start the optimization algorithm we generate the initial population of size N_{pop}. We select $N_{imp} = CHs$ of the most powerful countries to form the empires using GWO. The remaining $N_{col} = Normal\ nodes$ of the population will be the colonies each of which belongs to an empire. After that, colonies are distributed among imperialists based on their power to form the initial empires. To do that, we define the normalized cost of an imperialist by:

$$Fitness_{Normalized} = fitness_n - \max\left(fitness_i\right) \tag{37}$$

where $fitness_n$ is the cost of n^{th} imperialist and $Fitness_{Normalized}$ is its normalized cost. Colonies that should belong to the imperialist are defined by the normalized power calculated by:

$$Power_{Normalized} = \left| \frac{Fitness_{Normalized}}{\sum_{i=1}^{N_{imp}} fitness_i} \right| \tag{38}$$

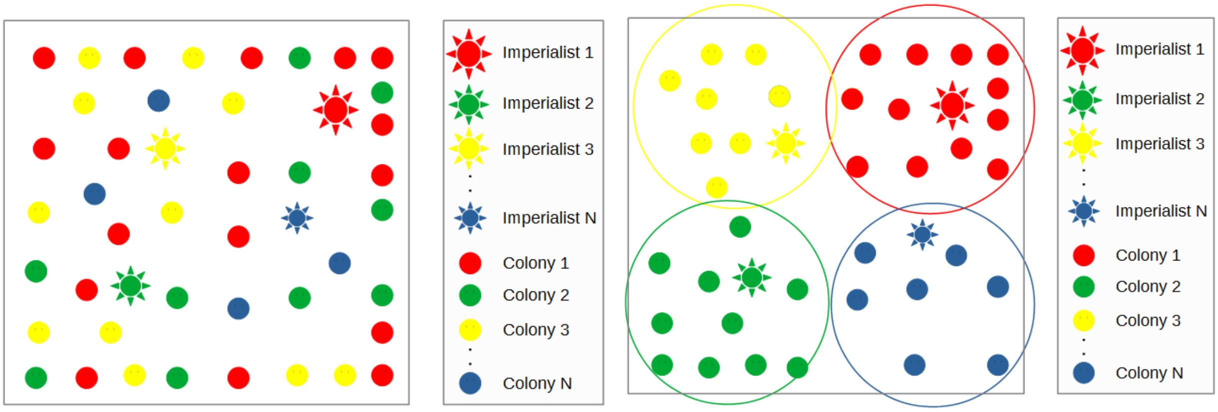
Then, the initial number of colonies of an empire will be:

$$InitNumCol_n = \left(Power_{Normalized} \times Num_{col}\right) \tag{39}$$

where Num_{col} is the number of all colonies and $InitNumCol_n$ is the initial number of colonies of the n^{th} empire. For each imperialist, we randomly choose $InitNumCol_n$ and give them to it in order to divide the colonies. Weaker empires have a smaller number of colonies while bigger ones have more.

Following this method exactly like it is, the distribution of empires will be unbalanced as shown in the figure 3. We notice also that, a colony can belong to a far imperialist, which generates great means for acquiring the wealth of the colonized country. To remedy these problems already mentioned, we add the distance criterion when the imperialist colonizes the countries. That means that they only choose a country within a specified area.

We note also that there will be certain countries that do not belong to any imperialist (Empires with no colonies). In the original algorithm, these empires will be eliminated.



(a) Without the distance parameter.

(b) With the distance parameter.

FIGURE 3. Cluster join using ICA.

In our solution, these countries elect themselves as Dependent countries which must satisfy their needs alone, if they have much wealth. Otherwise, they request help from the closest colony.

Algorithm 3 Pseudo Code of Member Joining Using ICA

Require: N Countries
Ensure: Colonies for each imperialist
  - Initialize the imperialists using GWO
  while *iteration* $< N$ **do**
    - Move the colonies toward their relevant imperialist (Assimilating) using distance criterion.
    - Compute the total cost of all empires
    - Pick the weakest colony from the weakest empire and give it to the empire that is more likely to possess it (Imperialistic competition)
    if There is an empire with no colonies **then**
      if This country has much wealth **then**
        - Elect this country as Dependent
      else
        - The country requests help from the closest colony.
      end if
    end if
    - iteration = iteration + 1
  end while
  return Colonies for each imperialist

G. DATA TRANSMISSION

After the clustering phase, the nodes send their data to the concerned CH as in the LER-GA [6], [7] protocol. The cluster, in this case, is subdivided into regions and zones. Each member knows his predecessor to wait to receive the data of the lower level, and successor to transmit his data to the higher level. After the CH is received all the data of its members, he sends them to the BS.

V. ILLUSTRATIVE CASE

We suppose that the coordinates of our BS are (500,500). The network is composed of 10 nodes dispersed randomly with an initial battery power of 0.75J and buffer size of 256 bits. Nodes will be grouped according to GWO and WOA for one round and one iteration. At a given moment, the values of the different parameters were recorded in table 6.

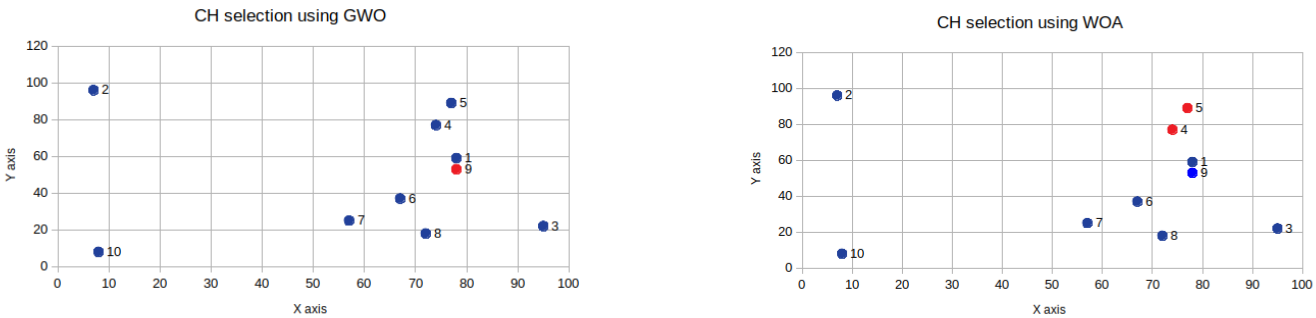
When we apply the original algorithm of GWO or the modified one, the CH elected is the node 9. We notice by the figure 4a that the CH is positioned in a strategic place to receive the maximum of data. On the other hand, when we apply the algorithm WOA, the choice depends on the value of p and A. If $p < 0.5$ and $A < 1$, the node chosen as CH is 4. Otherwise, if $p \geq 0.5$, the CH is node 5. Figure 4b represents clustering using WOA. In both cases, the position of the CHs is not beneficial in terms of data collection. Other comparisons will be found in the Results section.

VI. MODELING AND VERIFICATION

Formal methods are used to model and verify the complex systems using mathematical entities. There are four categories of the formal methods: formal specification, formal proofs, abstraction and model checking [47]. This latter has been under the considerable attraction of researchers these days, due to its capacity to detect worst case scenarios. These scenarios are not possible in computer simulations and other testing techniques. Most hidden errors and bugs in different systems, codes and protocols can be easily found. Some important properties like deadlock-free, safety, etc, need to be guaranteed in critical systems such as e-health applications, where human lives are concerned. The failure of some system parts may result in severe damage to equipment or the environment. It is necessary to find bugs using formal verification before system implementations. Some properties such as safety and liveness cannot be validated using simulation. Thats why when designing safety-critical systems, using formal methods with verified desirable properties is a necessity. To perform formal verification, PRISM, SPIN and UPPAAL are the most frequently used tools [48]. According to [48], UPPAAL is recommended to use in modeling of communication protocol which involves time. We find out that Broadcast transmission in UPPAAL enables it as a

TABLE 6. **Values of the different parameters at t = 5ms.**

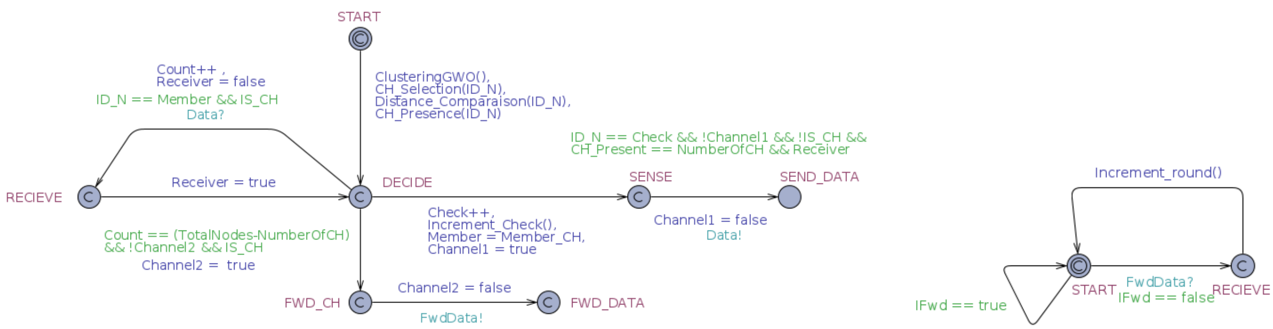
Nodes	X	Y	OccupiedMemory	Current battery power	fitness
1	78	59	32	0.266297	113,293041842
2	7	96	16	0.265841	120,391234588
3	95	22	32	0.266285	113,142636991
4	74	77	8	0.26583	24,870323651
5	77	89	1	0.266274	64,851620409
6	67	37	64	0.265818	97,37376436
7	57	25	32	0.266263	113,473284611
8	72	18	16	0.265816	121,23507957
9	78	53	16	0.266252	121,304446822
10	8	8	16	0.265806	120,251344027



(a) CH selection using GWO

(b) CH selection using WOA.

FIGURE 4. **Illustrative case.**



(a) Node.

(b) BS.

FIGURE 5. **GWO/WOA protocols state transition for one round.**

perfect tool to model WSN protocols, a feature not possible in SPIN. Whereas SPIN can provide buffered and rendezvous message passing, a feature not possible in UPPAAL. This channel features like collisions and message loss can be modeled in SPIN easily. According to [47], UPPAAL is more flexible in terms of usability and easiness. On the other hand, PRISM provides good features for probabilistic models.

A. MODELING A SINGLE ROUND IN GWO/WOA PROTOCOLS

To verify and validate our system, we have modeled GWO/WOA protocols for a single round, in a simplified way using UPPAAL 4.0 version. Figure 5 shows the transition automate for the system in a single round.

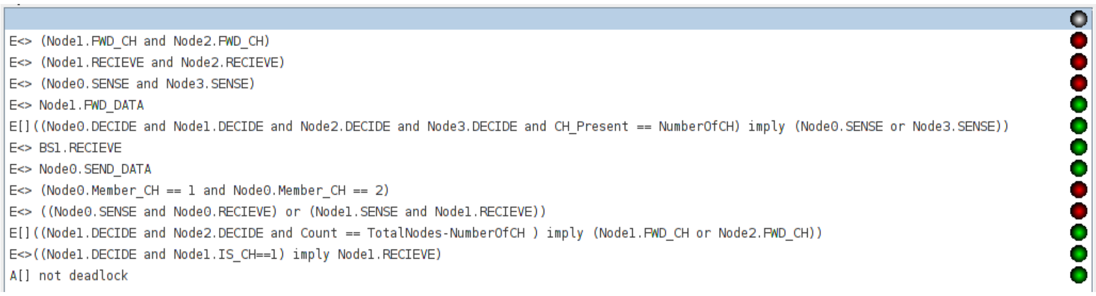
The model of GWO/WOA has been used to prove essential properties described in the figure 6. These proofs are valid for several topologies. In what follows we will detail these properties verified for a network composed of 10 nodes where only 2 CHs are elected. Message sequence chart is illustrated in figure 7.

Lemma 1: If the automate does not loop or block, deadlock property is satisfied.

Proof 1: According to the figure 7, the clustering phase and the transmission data phase terminates without interruptions.

Lemma 2: If a node belongs to only one cluster and a cluster has only one CH, safety property is satisfied.

$E <> \big(Node0.Member_{CH} == 1 \; and \; Node0.Member_{CH} == 2\big)$ (true if not satisfied).



```
E<> (Node1.FWD_CH and Node2.FWD_CH)
E<> (Node1.RECIEVE and Node2.RECIEVE)
E<> (Node0.SENSE and Node3.SENSE)
E<> Node1.FWD_DATA
E[]((Node0.DECIDE and Node1.DECIDE and Node2.DECIDE and Node3.DECIDE and CH_Present == NumberOfCH) imply (Node0.SENSE or Node3.SENSE))
E<> BS1.RECIEVE
E<> Node0.SEND_DATA
E<> (Node0.Member_CH == 1 and Node0.Member_CH == 2)
E<> ((Node0.SENSE and Node0.RECIEVE) or (Node1.SENSE and Node1.RECIEVE))
E[]((Node1.DECIDE and Node2.DECIDE and Count == TotalNodes-NumberOfCH ) imply (Node1.FWD_CH or Node2.FWD_CH))
E<>((Node1.DECIDE and Node1.IS_CH==1) imply Node1.RECIEVE)
A[] not deadlock
```

FIGURE 6. Essential properties verified on UPPAAL.

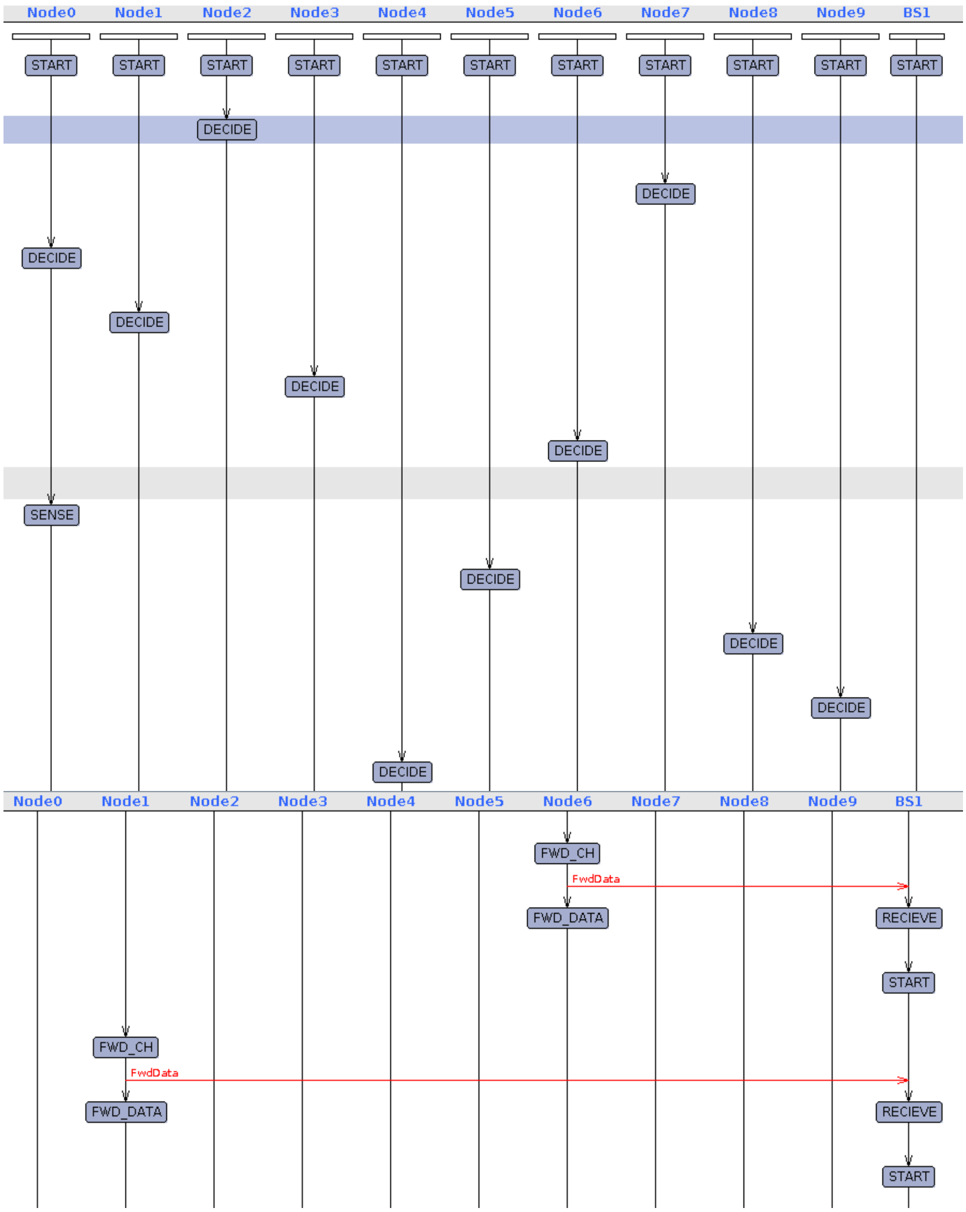


FIGURE 7. Message sequence chart illustrating a single round on GWO/WOA protocols.

Proof 2: The CH election depends on the value of the fitness function. If a node has the maximum value it will be chosen as a CH. In addition, two nodes can never have an equal fitness value because it depends on some case on the position of nodes and two nodes cannot be at the same place at the same time.

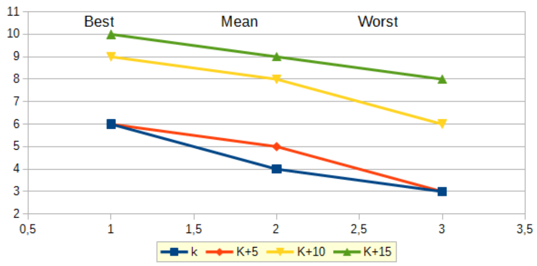
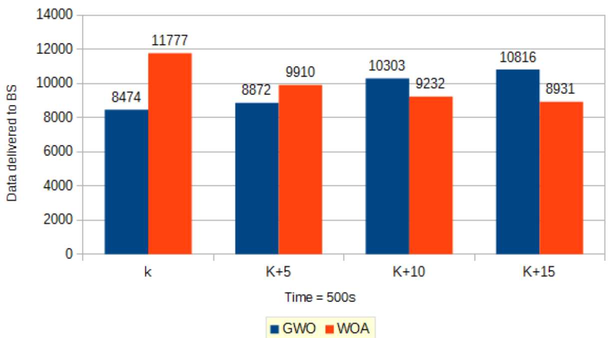
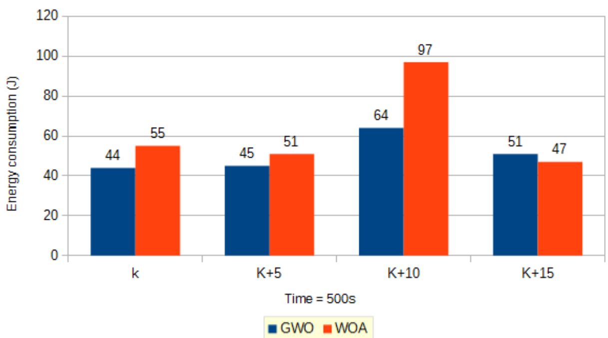


FIGURE 8. **Fitness values over 20 independent runs for k, k + 5, k + 10 and k + 15 iterations.**

(a) Number of alive nodes.



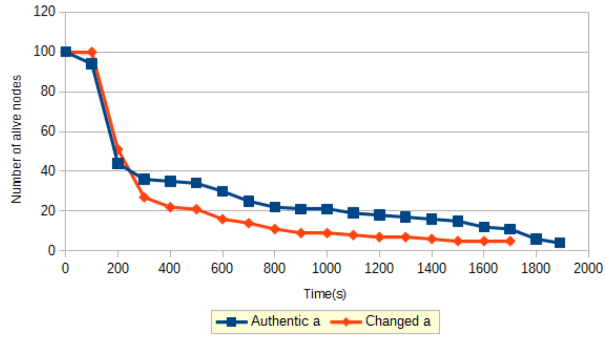
(b) Data packets received.



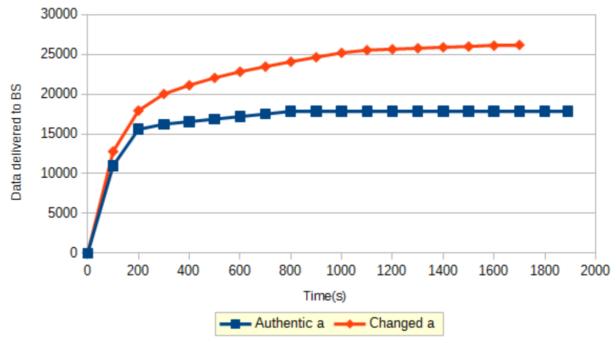
(c) Energy consumed by nodes.

FIGURE 9. **Different number of iterations for GWO and WOA.**

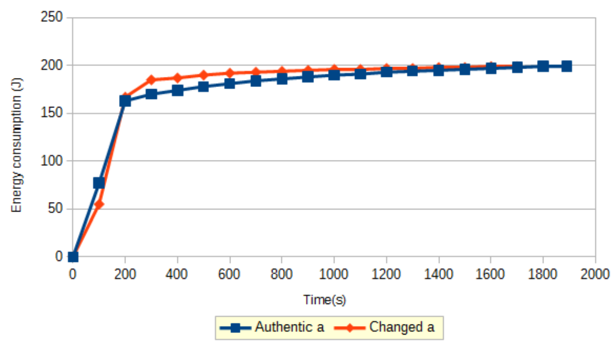
Lemma 3: If the clustering phase ends and a node can be an ordinary node or a CH at a giving time, liveness property is satisfied.



(a) Number of alive nodes.



(b) Data packets received.



(c) Energy consumed by nodes.

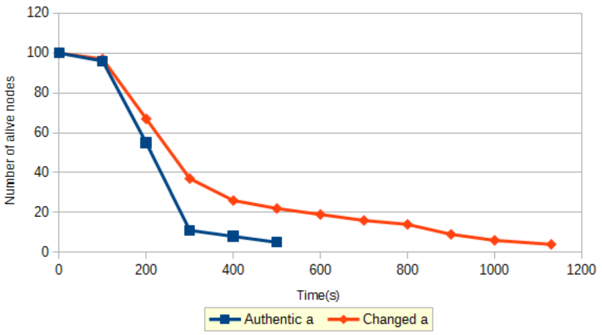
FIGURE 10. **Different values of the parameter a for GWO.**

$E <> \Big((Node0.SENSEandNode0.RECIEVE)$ or $(Node1.SENSEandNode1.RECIEVE)\Big)$ (true if not satisfied).

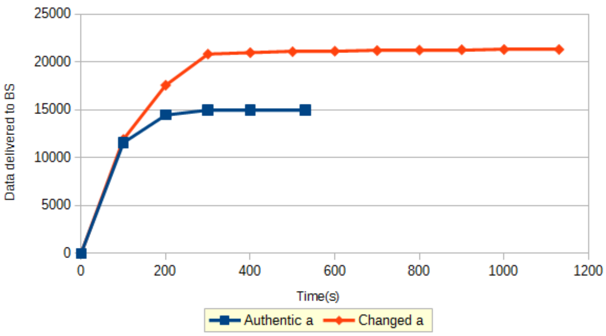
Proof 3: According to figure 5, a process in the state DECIDE cannot transit to states SENSE or RECIEVE at the same time. Its transition depends on the guard IS-CH updated in the clustering phase that ends correctly within Lemma 1 and Lemma 2.

VII. SIMULATION RESULTS

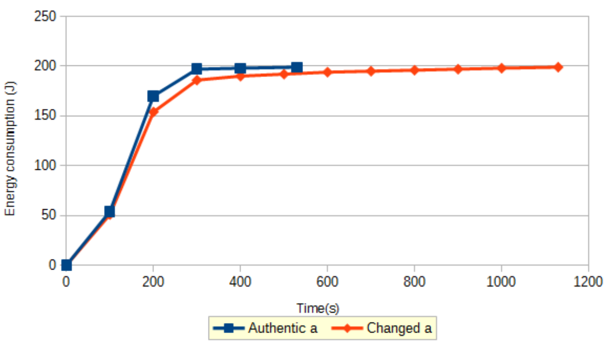
The network sensing area was assumed to be $100 \times 100\ m^2$. The initial simulations are on WSN #1 with 100 nodes. Then, the simulations were also performed on WSN #2 with 50 nodes and the third scenario WSN #3 with 250 nodes. The BS position was also varied for



(a) Number of alive nodes.

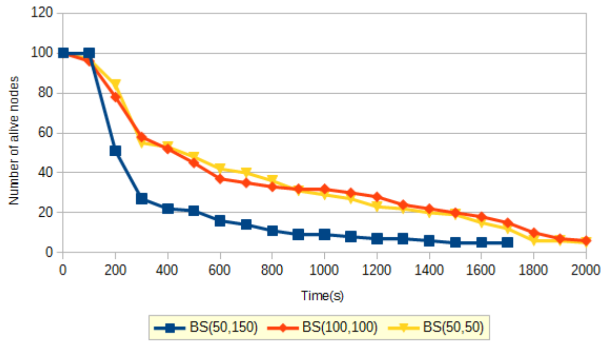


(b) Data packets received.

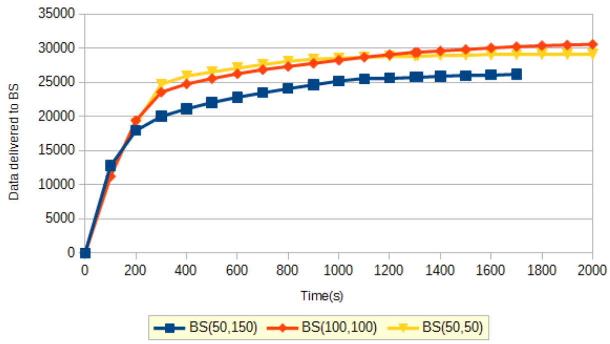


(c) Energy consumed by nodes.

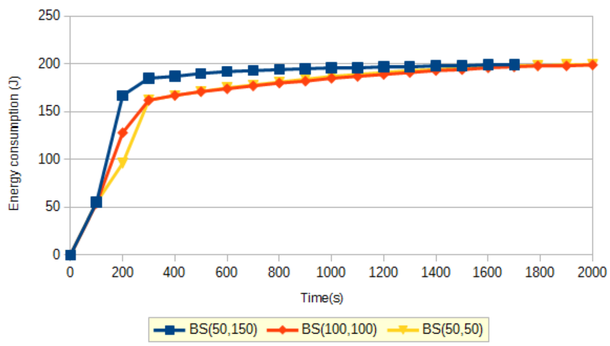
FIGURE 11. Different values of the parameter a for WOA.



(a) Number of alive nodes.



(b) Data packets received.



(c) Energy consumed by nodes.

FIGURE 12. Different BS locations for GWO.

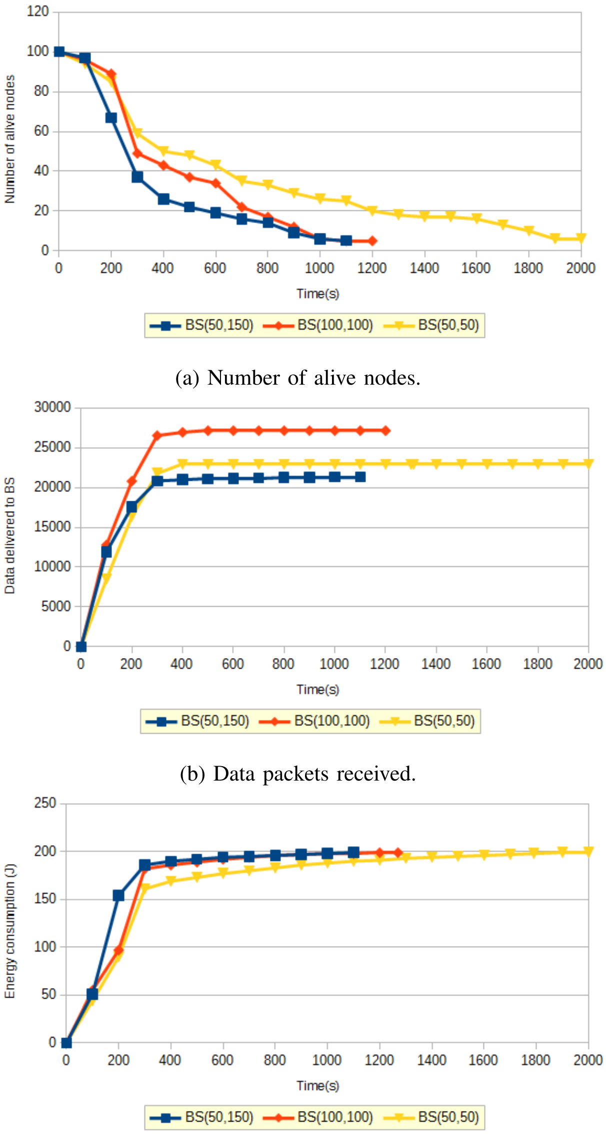
different scenarios. In the first scenario, the BS was positioned inside the sensing area at (50, 50), then in the second scenario the BS is positioned at the edge of sensing area at (100, 100) and finally, in the third scenario, the BS is positioned outside the sensing area at (50, 150). The various parameters considered for simulations are given in Table 7.

A. PERFORMANCE EVALUATION

The algorithm was run for 20 times and the average of the instances of the resultant data was chosen for plotting results. The simulations were made based on many parameters and by varying several conditions. Firstly, we changed the parameters for both GWO and WOA protocols. Then, we compared these two protocols together to study the energy consumed and the number of packets received by the BS. Finally, these two algorithms are compared to Fuzzy-GWO [26]

TABLE 7. List of the network parameters.

Parameter	Value
Target area	100×100 m^2
BS position	$(50, 50), (100, 100), (50, 150)$
Number of nodes	$50 - 100 - 250$
Initial energy of node	$2J$ or $4J$
Buffer size	$2^{24} bits$
Transmitter/Receiver electronics - E_{elec}	$50nJ/bit$
Transmitter amplifier (free space) - ϵ_{fs}	$100pj/bit/m^2$
Transmitter amplifier (multipath) - ϵ_{mp}	$0.013pJ/bit/m^4$
Data aggregation energy cost - E_{DA}	$50nJ/bit$
Packet size	$4000 bits$
Number of iterations	Variable

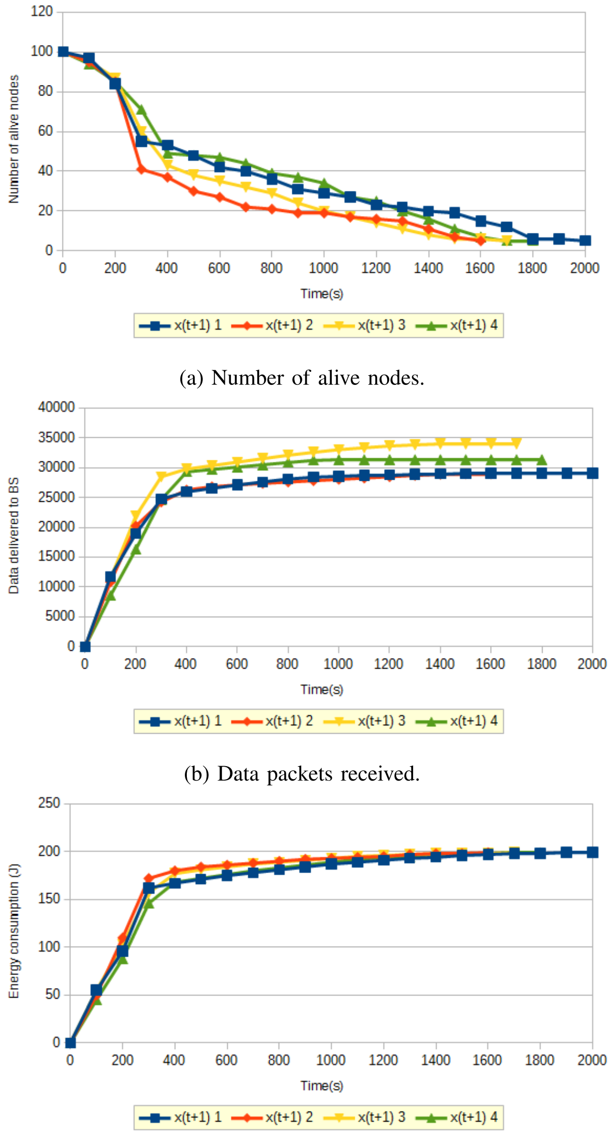


(a) Number of alive nodes.

(b) Data packets received.

(c) Energy consumed by nodes.

FIGURE 13. Different BS locations for WOA.



(a) Number of alive nodes.

(b) Data packets received.

(c) Energy consumed by nodes.

FIGURE 14. Different values of $x(t+1)$ for GWO.

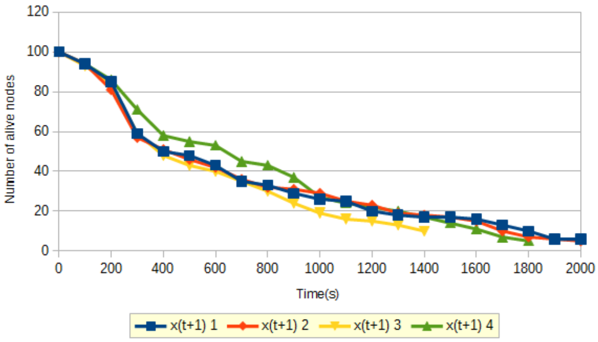
from literature. The results and analysis of these simulations are given in the subsequent sections.

1) FITNESS VALUE VS NUMBER OF ITERATIONS

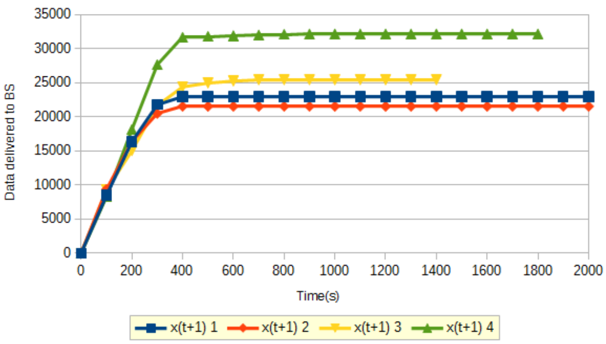
Figure 8 represents the relationship between the number of iterations and the fitness value. Note that, more the number of iterations increases, more the value of fitness is improved, except that it affects the calculation time that will inevitably increase. As a result, it is necessary to find a compromise between the number of iterations and the desired optimum.

2) DIFFERENT NUMBER OF ITERATIONS

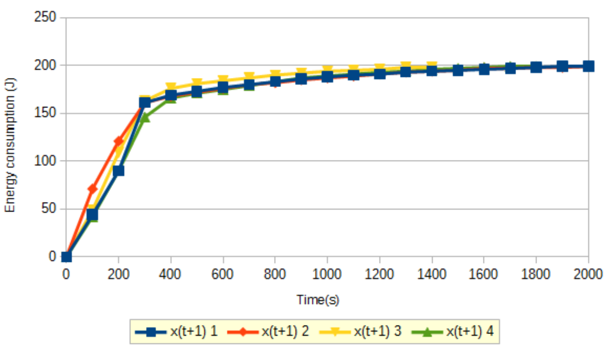
Figure 9 represents the results obtained when changing the number of iterations. Note that at time $t = 100s$, the number of alive nodes is greater when the number of iterations is equal to the number of clusters k for the WOA algorithm. The opposite of this observation is noted when the algorithm used is GWO. Same for the amount of data received by the BS which is also plenty when using WOA and very low when using GWO. When the number of iterations equals $k + 5$, $k + 10$ and $k + 15$, the number of alive nodes and the number of received packets decrease sequentially for the algorithm WOA and increase for the GWO algorithm. This difference in values between the two algorithms can be explained by the fact that, when using WOA algorithm, there are two supplement parameters l and p, randomly selected, compared to the GWO algorithm. These parameters help to explore more quickly the search space. In that case, the number of iterations requested is then minimized and the complexity of calculation



(a) Number of alive nodes.

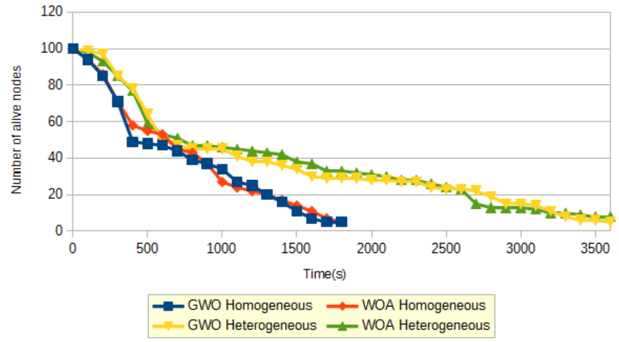


(b) Data packets received.

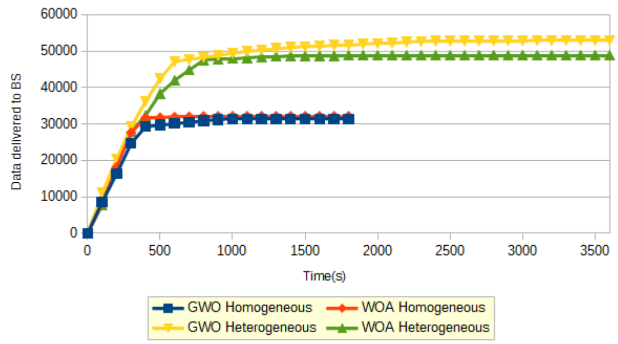


(c) Energy consumed by nodes.

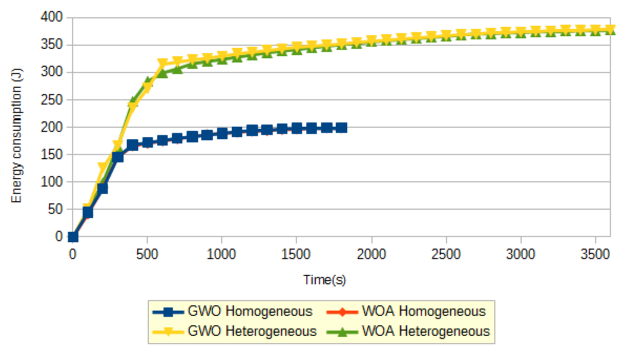
FIGURE 15. Different values of $x(t+1)$ for WOA.



(a) Number of alive nodes.



(b) Data packets received.



(c) Energy consumed by nodes.

FIGURE 16. Homogeneous Vs Heterogeneous networks.

is decreased. On the other hand, the GWO algorithm needs more iterations to arrive at satisfactory results in terms of lifespan and data quantity.

3) DIFFERENT VALUES OF THE PARAMETER *A*

Figures 10 and 11 represent the results obtained when the parameter a is changed using the two formulas 9 and 10 for the two algorithms GWO and WOA respectively. Curves (a), (b) and (c) are results obtained for number of alive nodes, data packets received and energy consumed by nodes respectively. We note from these figures that, when the time is less than 200s, the number of living nodes decreases rapidly. This is observed for both values of a using both algorithms. This remarkable energy consumption is due to the number of packets sent to the BS, an average of 17000 packets sent in only 200s. Beyond this time, the nodes die in a gradual manner for both values of a. More in detail, they die faster when the value of a is changed in the GWO algorithm. The opposite of this observation is seen when using the WOA algorithm. As far as energy consumption is concerned, it changes almost in a similar way for the two values of a in both algorithms. The amount of data is much larger by $\frac{1}{3}$ compared to that obtained when a is authentic for both algorithms. Unlike the results obtained by the GWO algorithm, when we modify the value of a by the formula 10 in the WOA algorithm, the lifetime increases and the amount of data is considerable. This can be explained by the fact that when using the formula 10, we decrease the value of a exponentially instead of doing it linearly as in the formula 9, which allows us to explore the search space much more quickly.

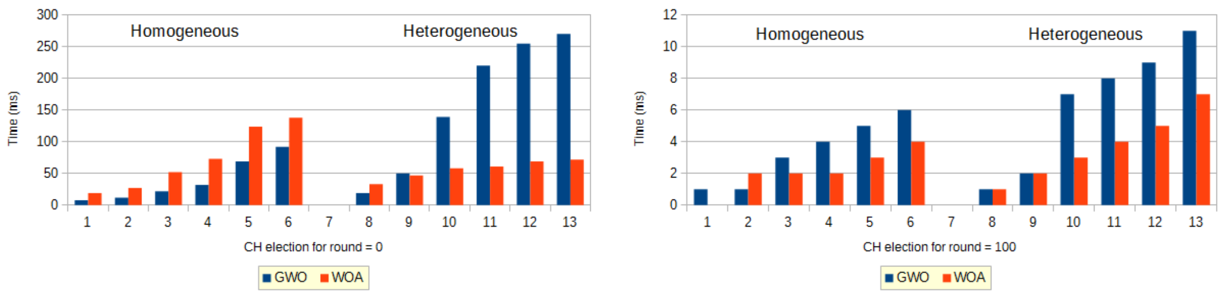


FIGURE 17. Time complexity.

4) DIFFERENT BS LOCATIONS

Figures 12 and 13 represent the results obtained when we change the BS location (50,50), (100,100) and (50,150). Curves (a), (b) and (c) are results obtained for number of alive nodes, data packets received and energy consumed by nodes respectively. Results show that, further the BS is from the monitored area, shorter is the life span of sensors. The amount of data is spectacular when the BS is located at the center of the network.

5) DIFFERENT VALUES OF $X(T+1)$

Figures 14 and 15 represent the results obtained when we change the value of $x(t+1)$ using the formulas 14, 15, 16 and 17. Curves (a), (b) and (c) are results obtained for number of alive nodes, data packets received and energy consumed by nodes respectively. From curves for both algorithms, we notice that using the formula 14, the life span reaches up to 2000s comparing with the other formulas. While the amount of data is definitely better when using formulas 16 and 17. In this case, the way we choose to calculate $x(t+1)$ depends largely on the goal we want to achieve. If we favor the energy factor, formula 14 will be the right choice. If the amount of data received is our goal, formulas 16 and 17 will do the job. The formula 15 is deprecated when using GWO algorithm.

6) HOMOGENEOUS VS HETEROGENEOUS NETWORKS

Figure 16 represents the results obtained for the two algorithms GWO and WOA in homogeneous and heterogeneous networks. According to the curves, we notice that there is not a big difference between the results. The curves are almost superimposed. Except that, in a heterogeneous network, the GWO algorithm delivers more packets comparing to its counterpart WOA algorithm.

7) TIME COMPLEXITY

Figure 17 represents the estimated complexity time to find the CHs in a homogeneous and heterogeneous network, with two different rounds, using the two algorithms GWO and WOA. According to the curves, we notice that, in the first round, using a homogeneous network, WOA algorithm takes more time to choose CHs compared to GWO algorithm. This observation is different when the network is heterogeneous. Same for the round 100, where WOA algorithm proves its performance in computing time comparing to GWO algorithm. Overall Computational complexity of the developed GWO/WOA algorithms is as follows:

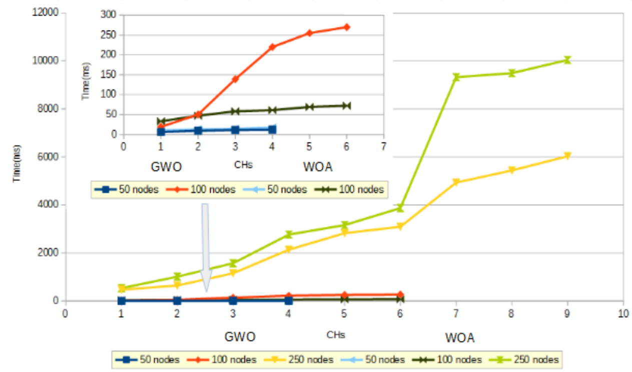
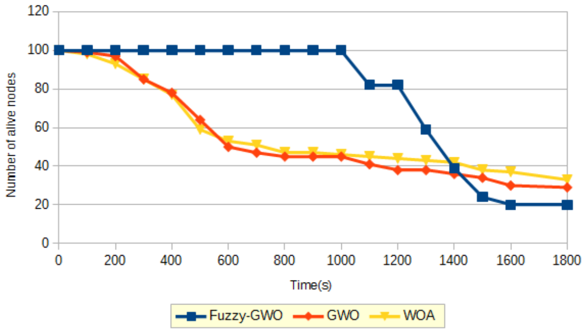


FIGURE 18. Network size vs time.

Theorem 1: Time complexity of the algorithms GWO and WOA is $O\Big(k \times (n^2 + n\log n)\Big)$ where n is the number of grey wolves/whales and k is the maximum number of iterations.

Proof: The sorting mechanism, the iteration size and the number of grey wolves/whales, determine the computational complexity of the developed GWO/WOA algorithms. Quicksort mechanism has been implemented for sorting. Complexity for best case is $O(n\log n)$ and for worst case is $O(n^2)$.

Due to better convergence, high exploration, low computational complexity, faster local minima avoidance and less computational time required to solve real problems, proposed GWO/WOA algorithms can be used to solve large-scale optimization problems. Moreover, from figure 9, we have $k_{whale} < k_{wolf}$. Therefore, $O(WOA) < O(GWO)$. WOA algorithm is definitely better in terms of computation time, which favors it, especially for real-time applications.



(a) Number of alive nodes.

(b) Data packets received.

FIGURE 19. Data delivery and number of alive nodes vs time.

8) NETWORK SIZE

Figure 18 represents the time spent for clustering using the two algorithms in several network sizes (50, 100 and 250). Note that in a network of 50 nodes, the curves are superimposed, which indicates that the clustering time is the same for both algorithms. For a network of 100 nodes, the WOA algorithm is more beneficial in time than its counterpart GWO algorithm. On the other hand, beyond this size, this remark is reversed. GWO algorithm offers more performance when the network size increases.

9) DATA DELIVERY AND NUMBER OF ALIVE NODES VS TIME

Figure 19 represents a comparison in terms of lifetime and data delivery between GWO, WOA, and Fuzzy-GWO [26] algorithms. Note that in Fuzzy-GWO algorithm the first node dies at $t = 1000s$, unlike the other protocols where the death of the nodes is gradual. On the other hand, GWO and WOA algorithms provide 4 times more the amount of data delivered to BS.

VIII. CONCLUSION

The hunting and search behavior of social animals, such as grey wolves and whales, has been the emerging area of swarm intelligence. In this paper, a GWA/WOA was developed to solve the buffer overflow problem when collecting data in a heterogeneous network. It was compared using different parameters and to other algorithms such as Fuzzy-GWO. The experimental results revealed that GWA/WOA outperformed the other algorithm in most of the cases. Furthermore, we deduced that GWO algorithm is more beneficial for large-scale networks and WOA algorithm is more efficient in heterogeneous networks under 100 nodes. As perspectives, to solve the buffer overflow problem at the CH step, we allow nodes with low sampling rate and large cache capacity to store partial data of the CH when the buffer of the CH is full. Also, beta CH and delta CH help alpha CH to cache the data temporarily to increase the buffer overflow time and reduce the data loss.

REFERENCES

[1] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 38–43, Mar. 2017.

[2] R. Hamidouche, Z. Aliouat, and A. M. Gueroui, "Bio-inspired vs classical solutions to overcome the IoT challenges," in *Proc. 3rd Cloudification Internet Things (CIoT)*. Paris, France, Jul. 2018, pp. 1–7.

[3] J. Wang, J. Cao, R. Sherratt, and J. H. Park, "An improved ant colony optimization-based approach with mobile sink for wireless sensor networks," *J. Supercomput.*, vol. 74, no. 12, pp. 6633–6645, Dec. 2018.

[4] M. Aboubakar, M. Kellil, A. Bouabdallah, and P. Roux, "Toward intelligent reconfiguration of RPL networks using supervised learning," in *Proc. Wireless Days (WD)*, Manchester, U.K., Apr. 2019, pp. 1–4.

[5] A. A. A. Ari, B. O. Yenke, N. Labraoui, I. Damakoa, and A. Gueroui, "A power efficient cluster-based routing algorithm for wireless sensor networks: Honeybees swarm intelligence based approach," *J. Netw. Comput. Appl.*, vol. 69, pp. 77–97, Jul. 2016.

[6] R. Hamidouche, Z. Aliouat, and A. Gueroui, "Low energy-efficient clustering and routing based on genetic algorithm in WSNs," in *Proc. Int. Conf. Mobile, Secure, Program. Netw. (MSPN)* (Lecture Notes in Computer Science), vol. 11005. Paris, France: Springer, Jan. 2018, pp. 143–156.

[7] R. Hamidouche, Z. Aliouat, and A. M. Gueroui, "Genetic algorithm for improving the lifetime and QoS of wireless sensor networks," *Wireless Pers. Commun.*, vol. 101, no. 4, pp. 2313–2348, Aug. 2018.

[8] A. Djedouboum, A. A. A. Ari, A. M. Gueroui, A. Mohamadou, and Z. Aliouat, "Big data collection in large-scale wireless sensor networks," *Sensors*, vol. 18, no. 12, p. 4474, Dec. 2018.

[9] R. Hamidouche, Z. Aliouat, A. M. Gueroui, A. A. A. Ari, and L. Louail, "Classical and bio-inspired mobility in sensor networks for IoT applications," *J. Netw. Comput. Appl.*, vol. 121, pp. 70–88, Nov. 2018.

[10] R. Hamidouche, M. Khentout, Z. Aliouat, A. M. Gueroui, and A. A. A. Ari, "Sink mobility based on bacterial foraging optimization algorithm," in *Proc. IFIP Int. Conf. Comput. Intell. Appl., (CIIA)*, vol. 522. Oran, Algeria: Springer, 2018, pp. 352–363.

[11] M. Maadi and M. Maadi, "Optimization of cluster heads selection by imperialist competitive algorithm in wireless sensor networks," *Int. J. Comput. Appl.*, vol. 89, no. 19, pp. 29–34, Mar. 2014.

[12] S. Noritaka, M. Hiromi, M. Hiroki, and M. Michiharu, "Centralized and distributed clustering methods for energy efficient wireless sensor networks," in *Proc. Int. Multi Conf. Eng. Comput. Scientists* (Lecture Notes in Engineering and Computer Science), vol. 1, Mar. 2009, pp. 978–988.

[13] S. K. Gupta, N. Jain, and P. Sinha, "Clustering protocols in wireless sensor networks: A survey," *Int. J. Appl. Inf. Syst.*, vol. 5, no. 2, pp. 38–51, Jan. 2013.

[14] J. Wang, Y. Gao, W. Liu, W. Wu, and S.-J. Lim, "An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks," *Comput., Mater. Continua*, vol. 58, no. 3, pp. 711–725, 2019.

[15] J. Wang, C. Ju, Y. Gao, A. K. Sangaiah, and G.-J. Kim, "A PSO based energy efficient coverage control algorithm for wireless sensor networks," *Comput., Mater. Continua*, vol. 56, no. 3, pp. 433–446, Jan. 2018.

[16] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, and H.-J. Kim, "An intelligent data gathering schema with data fusion supported for mobile sink in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 3, 2019, Art. no. 1550147719839581.

[17] S. M. Hosseinirad, "Multi-layer clustering topology design in densely deployed wireless sensor network using evolutionary algorithms," *J. AI Data Mining*, vol. 6, no. 2, pp. 297–311, 2018.

[18] J. Wang, Y. Gao, X. Yin, F. Li, and H.-J. Kim, "An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 2018, Nov. 2018, Art. no. 9472075.

[19] M. Mehr, "Cluster head election using imperialist competitive algorithm (CHEI) for wireless sensor networks," *Int. J. Mobile Netw. Commun. Telematics*, vol. 4, no. 3, pp. 1–9, Jun. 2014.

[20] H. V. Chaitra and G. K. Ravikumar, "An energy efficient cluster selection optimization using evolutionary imperialist competitive algorithm," *Int. J. Comput. Appl.*, vol. 975, p. 8887, Oct. 2015.

[21] A. S. Rostami, M. Badkoobe, F. Mohanna, A. A. R. Hosseinabadi, and V. E. Balas, "Imperialist competition based clustering algorithm to improve the lifetime of wireless sensor network," in *Proc. Int. Workshop Soft Comput. Appl.* Arad, Romania: Springer, 2016, pp. 189–202.

[22] N. A. Al-Aboody and H. S. Al-Raweshidy, "Grey wolf optimization-based energy-efficient routing protocol for heterogeneous wireless sensor networks," in *Proc. 4th Int. Symp. Comput. Bus. Intell. (ISCBI)*. Olten, Switzerland, Sep. 2016, pp. 101–107.

[23] B. Kalpana, "An energy efficient grey wolf optimizer for wireless sensor networks," *Int. J. Comput. Sci. Inf. Secur.*, vol. 14, no. 11, pp. 286–291, Nov. 2016.

[24] M. Sharawi and E. Emary, "Impact of grey wolf optimization on WSN cluster formation and lifetime expansion," in *Proc. 9th Int. Conf. Adv. Comput. Intell. (ICACI)*, Doha, Qatar, Feb. 2017, pp. 157–162.

[25] Z. Jabinian, V. Ayatollahitafti, and H. Safdarkhani, "Energy optimization in wireless sensor networks using grey wolf optimizer," *J. Soft Comput. Decis. Support Syst.*, vol. 5, no. 3, pp. 1–6, Apr. 2018.

[26] P. Diwan and M. R. Khan, "Energy efficient communication for WSNs using grey-wolf optimization algorithm," *Int. J. Eng. Comput. Sci.*, vol. 5, no. 12, pp. 19793–19805, Dec. 2016.

[27] X. Zhao, H. Zhu, S. Aleksic, and Q. Gao, "Energy-efficient routing protocol for wireless sensor networks based on improved grey wolf optimizer," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 6, pp. 2644–2657, Jun. 2018.

[28] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition," in *Proc. IEEE Congr. Evol. Comput.*, Singapore, Sep. 2007, pp. 4661–4667.

[29] S. Shamshirband, A. Amini, N. B. Anuar, M. L. M. Kiah, Y. W. Teh, and S. Furnell, "D-FICCA: A density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks," *Measurement*, vol. 55, pp. 212–226, Sep. 2014.

[30] S. Hosseini and A. A. Khaled, "A survey on the imperialist competitive algorithm metaheuristic: Implementation in engineering domain and directions for future research," *Appl. Soft Comput.*, vol. 24, pp. 1078–1094, Nov. 2014.

[31] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[32] S. Zhang and Y. Zhou, "Grey wolf optimizer based on powell local optimization method for clustering analysis," *Discret. Dyn. Nat. Soc.*, vol. 2015, pp. 1–17, Nov. 2015.

[33] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, 2018.

[34] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Comput. Appl.*, vol. 26, no. 5, pp. 1257–1263, Jul. 2015.

[35] B. Mahdad and K. Srairi, "Blackout risk prevention in a smart grid based flexible optimal strategy using Grey Wolf-pattern search algorithms," *Energy Convers. Manage.*, vol. 98, pp. 411–429, Jul. 2015.

[36] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[37] T. R. Biyanto, Matradji, S. Irawan, H. Y. Febrianto, N. Afdanny, A. H. Rahman, K. S. Gunawan, J. A. Pratama, and T. N. Bethiana, "Killer whale algorithm: An algorithm inspired by the life of killer whale," *Procedia Comput. Sci.*, vol. 124, pp. 151–157, Jan. 2017.

[38] A. R. Jadhav and T. Shankar, "Whale optimization based energy-efficient cluster head selection algorithm for wireless sensor networks," 2017, *arXiv:1711.09389*. [Online]. Available: https://arxiv.org/abs/1711.09389

[39] L. Qing, Q. Zhu, and M. Wang, "Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks," *Comput. Commun.*, vol. 29, no. 12, pp. 2230–2237, Aug. 2006.

[40] N. Mittal, U. Singh, and B. S. Sohi, "Modified grey wolf optimizer for global engineering optimization," *Appl. Comput. Intell. Soft Comput.*, vol. 2016, pp. 1–16, Mar. 2016, Art. no. 7950348.

[41] K. Dudani and A. Chudasama, "Partial discharge detection in transformer using adaptive grey wolf optimizer based acoustic emission technique," *Cogent Eng.*, vol. 3, no. 1, Nov. 2016, Art. no. 1256083.

[42] Z. Beiranvand, A. Patooghy, and M. Fazeli, "I-LEACH: An efficient routing algorithm to improve performance & to reduce energy consumption in Wireless Sensor Networks," in *Proc. 5th Conf. Inf. Knowl. Technol.*, May 2013, pp. 13–18.

[43] S. Bandyopadhyay and E. J. Coyle, "Minimizing communication costs in hierarchically-clustered networks of wireless sensors," *Comput. Netw.*, vol. 44, no. 1, pp. 1–16, Jan. 2004.

[44] N. Jain, P. Sinha, and S. Gupta, "Clustering protocols in wireless sensor networks: A survey," *Int. J. Appl. Inf. Syst. (IJAIS)*, vol. 5, no. 2, pp. 41–50, Jan. 2013.

[45] M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo, "Model-based search for combinatorial optimization: A critical survey," *Ann. Oper. Res.*, vol. 131, nos. 1–4, pp. 373–395, Oct. 2004.

[46] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proc. 1st Eur. Conf. Artif. Life*, vol. 142, 1991, pp. 134–142.

[47] A. Naeem, F. Azam, A. Amjad, and M. W. Anwar, "Comparison of model checking tools using timed automata—PRISM and UPPAAL," in *Proc. IEEE Int. Conf. Comput. Commun. Eng. Technol. (CCET)*, Beijing, China, Aug. 2018, pp. 248–253.

[48] T. Fatima, K. Saghar, and A. Ihsan, "Evaluation of model checkers SPIN AND UPPAAL for testing wireless sensor network routing protocols," in *Proc. 12th Int. Bhurban Conf. Appl. Sci. Technol. (IBCAST)*, Jan. 2015, pp. 263–267.