Optimization of the Deployment of Wireless Sensor Networks Dedicated to Fire Detection in Smart Car Parks using Chaos Whale Optimization Algorithm

Slimane Charafeddine Benghelima, *Student Member, IEEE*[*], Mohamed Ould-Khaoua[*‖],
Ali Benzerbadj[†§], Oumaya Baala[‡] and Jalel Ben-Othman, *Senior Member, IEEE*[††]
[*]LRDSI, Department of Informatics, University of Blida 1, Algeria
[†] Ain Temouchent University Belhadj Bouchaib, Ain Temouchent, Algeria
[§]RIIR, University of Oran 1 Ahmed Ben Bella, Algeria
[‡]FEMTO-ST Institute, Univ. Bourgogne Franche-Comté, CNRS, Belfort, France
[§] Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France.
Et Université Sorbonne Paris Nord.
Email: [*]benghelima.slimane@etu.univ-blida.dz, [‖]mouldkhaoua@univ-blida.dz [†]ali.benzerbadj@univ-temouchent.edu.dz,
[‡]oumaya.baala@utbm.fr [††]jalel.benothman@centralesupelec.fr

***Abstract*—Smart Car Parks (SCPs) based on Wireless Sensor Networks (WSNs) are one of the most interesting IoT applications. The present study addresses the deployment optimization problem of two-tiered WSNs dedicated to fire monitoring in smart parking lots. Networks deployed inside the SCP consist of three types of nodes: Sensor Nodes (SNs) which cover the spots within the parking area, Relay Nodes (RNs) which forward alert messages generated by SNs, and the Sink node which is connected to the outside world (e.g, firefighters), through a high bandwidth connection. This paper proposes an algorithm based on chaos theory and Whale Optimization Algorithm (WOA), which minimizes simultaneously the deployed number of SNs, RNs, and the maximum distance from SNs to the sink node (network diameter) while ensuring coverage and connectivity. To evaluate the effectiveness of our proposal, we have conducted extensive tests. The results show that the Chaos WOA (CWOA) outperforms the original WOA in terms of solution quality and computation time and by comparison with the exact method, CWOA finds results very close to the optimal in terms of fitness value and is efficient in terms of computational time when the problem becomes more complex.**

***Index Terms*—Whale optimization algorithm, Chaos map, WSN deployment, Smart parking fire surveillance, Internet of things.**

I. INTRODUCTION

In recent years, the deployment of IoT technologies has increased as they enable the connection of physical devices to one another and to the Internet [1]. We discuss in this work WSNs, which are the very core of IoT systems, a network consisting of SNs, RNs, and sink nodes that have constrained resources, i.e., processing, communication, and storage. The fundamental function of an SNs is to gather data from the local environment and transfer it to the sink node directly or via RNs. A Smart Car Park (SCP) [10] is a typical application of IoT technology. Most existing work on the implementation of SCP systems have focused on the management of the car spots (called also targets) inside the parking area, where the fire surveillance within SCP is a an important safety criterion. In this work, we are interested in the problem of the deployment of WSNs in order to guarantee the coverage of all targets and the connectivity between SNs and the sink node, using RNs with respect to two-tiered architecture [15]. This problem is known to be an NP-hard [5], which requires the use of efficient algorithms such as meta-heuristic in order to tackle large instances.

Recently, meta-heuristics prove their performance in many recent fields like optimization of neural networks [4], cellular network planning [2], and WSN deployment [16]. In this work, we use a recent meta-heuristic called Whale Optimization Algorithm (WOA) [11], which proved its performance in recent works on the deployment of WSN such as in [13] and [8]. Despite the efficiency of this algorithm, it is not able to avoid a local optimum, therefore we propose an enhanced version in order to visit more region in the research space to find better solutions. More details will be discussed in the next sections.

The rest of this paper is organized as follows. Section II presents the related works on deterministic WSN deployment problem. Sections III and IV provide respectively the problem formulation, and the WOA definition. Section V presents the proposed algorithm. Section VI details the experimental results and analysis. Section VII concludes the paper.

II. RELATED WORKS

Several meta-heuristics were used in the literature to solve WSN coverage and connectivity issues [16]. The work in [12] focused on the optimization of the deployment of SNs using a Biogeography Based Optimization where the objectives are the minimization of the sensing interference and the number of deployed SNs, and the maximization of the target coverage, under the the connectivity constraint. Authors in [7], proposed a hybrid meta-heuristic based on the tunicate swarm optimizer and the salp swarm optimizer, with the main aim of determining a minimum number of potential positions selected to place SNs and maximize the target coverage and network connectivity. In [9], the work addressed the deployment of SNs for target coverage using improved gravitational search algorithm (GSA) called oppositional GSA. The objectives are the minimization of the number of SNs, the coverage of targets and the network connectivity. Authors in [18], proposed three adaptive strategies for WOA to solve the RNs deployment problem. The objectives are the minimization of the number of deployed RNs and the energy consumption under the connectivity constraint. In [14], the work dealt with the RNs deployment problem which has been solved using multi-objectives decomposition-based moth flame optimization meta-heuristic. The authors considered the following objectives: the minimization of the average intra-cluster distance and the average hop-count.

It should be noted that the works discussed above can be classified into two categories, SNs deployment for both coverage and connectivity issues, while the second category considered only the deployment of RNs for connectivity. In our case, we consider the deployment problem of both SNs and RNs simultaneously.

There are a few works taking into consideration the deployment of SNs and RNs in a sequential manner like in [6] where authors used an hybrid approach based on particle swarm optimization and iterated local search algorithms (PSO-ILS) to optimize firstly the deployment of SNs for target coverage, than to optimize the placement of RNs based on the deployed SNs in the first step for connectivity requirement.

Since the sequential deployment approach limits the search space, other works focused on the simultaneous deployment of both SNs and RNs. In [15], the authors proposed an exact and heuristic methods to solve respectively, small and large instances. The objectives are the minimization of the number of SNs and RNs and the outage probabilities of the links between all nodes in the network. The issue was formulated as a single objective which is the sum of all objectives without weights, which leads to find only one solution, whereas they have a conflicting objectives. To overcome the issue in [15], we have recently proposed a multi-objective linear program [3], in order to find all efficient solutions. The proposed method in [3] solves the problem of the simultaneous deployment of SNs and RNs in order to minimize their number and the network diameter[1]. The limitation of the work [3] is the high computational time. Therefore, in the current work we propose a meta-heuristic based algorithm to find a good compromise between the solution quality and the computational time.

[1]Represents the length of the longest path among all shortest paths between all the SNs and the sink node

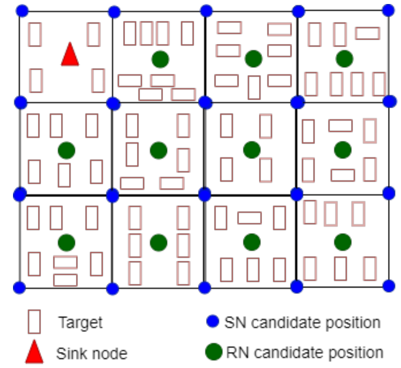


Fig. 1: Discretization of the deployment space (3×4 grid: 20 CSPs, 11 CRPs, 1 sink and 73 targets)

III. PROBLEM FORMULATION

In this work, we express the issue as follows. The deployment space is discretized as a two-dimensional grid, where the square cells corners are the SN candidate positions and the square cells centers are the RNs candidate positions, as depicted in Figure 1. Targets and the single sink node are randomly deployed within the square cells of the grid.

The objective is to select the minimal number of candidate positions to deploy SNs and RNs, and minimal network diameter, under the coverage and connectivity constraints. In fact, each target must be inside the sensing range of at least one SN which have to be connected to the sink node directly or through a path composed only of RNs to build a two-tiered architecture. On the other side, the generated networks should have a minimal diameter in order to minimize the network energy consumption and delay during the routing process.

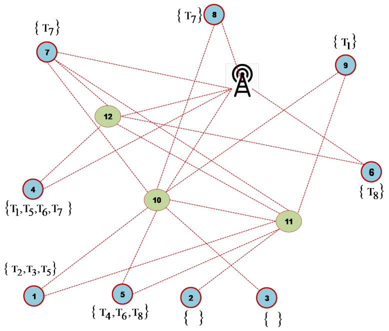
Notations:

Cov_{ij} and Com_{ij} denote respectively, coverage and communication matrices, and can be generated as follow:

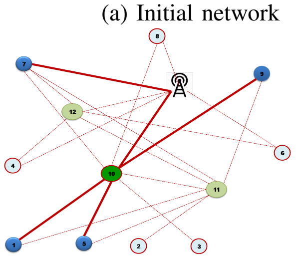
$$Cov_{ij} = \begin{cases} 1 & \text{if a SN } i \text{ covers the target j} \\ 0 & \text{Otherwise} \end{cases}$$

$$Com_{ij} = \begin{cases} 1 & \text{if nodes } (i,j) \text{ communicate directly} \\ 0 & \text{Otherwise} \end{cases}$$

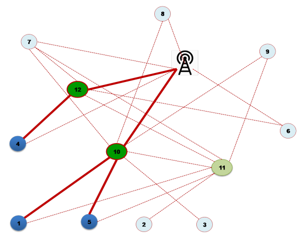
Using the above notations Cov_{ij} and Com_{ij}, we transform the initial problem to a graph, where each vertex of the graph is either SN, RN or sink node and every vertex SN has a list of target which can cover them. For instance, the Figure 2a depicts the problem of the deployment of WSN with 9 SN candidate positions (blue circles), $CSP = \{CSP_1, CSP_2, ..., CSP_9\}$, where each CSP_i has a list of targets, 3 RN candidate positions (green circles), $CRP = \{CRP_1, CRP_2, CRP_3\}$, and eight parking spots or Targets $T = \{T_1, T_2, ..., T_3\}$. This instance of problem contains two solutions as illustrated in Figure 2b and Figure 2c respectively. In the first solution, SNs are deployed



(a) Initial network



(b) Result 1: (4) SNs and (1) RN



(c) Result 2: (3) SNs and (2) RNs

Fig. 2: Initial network and induced sub-networks.

on four positions out of 10 and only one RN is placed rather than three, and for the second one, SNs are placed on three positions out of 10 and two RNs are placed rather than three. It is noteworthy that both constraints, coverage and connectivity, are satisfied in the two resulted networks. For example, as shown in the Figure 2b, the union of the lists of targets covered by the four chosen SNs is equal to the set of all targets $\{\{T_2, T_3, T_5\} \cup \{T_4, T_6, T_8\} \cup \{T_7\} \cup \{T_1\}\} = T$, which means that the coverage constraint is satisfied, and in the same Figure we can see that all chosen SNs are connected to the sink directly or via the RN number ten.

Solution representation: A solution is represented as a vector with a fixed size, which is determined by the number of SN and RN candidate positions in the WSN. Each element of the vector is either a SN or RN candidate position. The value of each element of the vector is set to 1 or 0 to indicate if a SN candidate position is chosen or not, and 2 or 0 to indicate if a RN candidate position is chosen or not. An example of the solution representation is given here after:

X_i	CSP_1	CSP_2	CRP_1	CSP_3	CRP_2	CSP_4
	1	0	0	1	2	1

Objective functions: In order to evaluate each solution, we use the following objective functions:

$$F1 = \min \sum_{i \in CSP} X_i$$

$$F2 = \min \frac{1}{2} \sum_{i \in CRP} X_i$$

$$F3 = \min\{ \max_{i \in CSP^{OPT}} (distance_{(sink,i)})$$

where CSP^{OPT} is the set of the CSPs chosen and sink is the index of the sink node. To solve this problem, we transform it into a mono-objective problem F using the weighted sum approach [7]:

$$F = \min\{\alpha F1 + \beta F2 + \gamma F3\}$$

Where: $\alpha + \beta + \gamma = 0$

IV. Whale Optimziation Algorithm (WOA)

The WOA [11] meta-heuristic performs the hunting activity of humpback whales by simulating their movements and sounds. The steps of the WOA method are namely, random prey search and the bubble net attack. These steps are mathematically modeled as follows [11].

A. Search for prey

When hunting, humpback whales need to find the location of the prey, however, the position of the prey in the search space is typically unknown. Therefore, whale positions are chosen randomly in order to generate new whale positions. The following equation describes this behavior [11].

$$\vec{D} = ||\vec{C} \cdot X_{rand}(t) - X_i(t)|| \tag{1}$$

$$X_i(t+1) = X_{rand}(t) - \vec{A} \cdot \vec{D} \tag{2}$$

Where $\vec{D}$ is the distance between two whales, $X_i(t)$ is the position vector of the whale i (solution i) at iteration t, $X_{rand}(t)$ is a random whale chosen randomly at iteration t, $||\ ||$ denotes the absolute value and $'.'$ denotes an element-by-element multiplication. The rest of parameters $\vec{A}$ and $\vec{C}$ are updated as follows [11].

$$\vec{a} = 2 - 2 \cdot \frac{t}{\max\_iteration} \tag{3}$$

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a} \tag{4}$$

$$\vec{C} = 2 \cdot \vec{r} \tag{5}$$

where $\vec{r}$ is a random vector in range [0, 1], $\vec{a}$ decreases linearly from 2 to 0 during the iterations, $\vec{A}$ is a random value in range $[\vec{-a}, \vec{a}]$ and $\vec{C}$ is a random vector in range [0, 2].

B. The bubble net attack

The bubble net attack is composed of two essential movements, encircling the prey and the spiral-shaped trajectory.

1) Encircling prey: Humpback whales can detect prey places and encircle them in order to grab the prey. To mimic the process, the other search agents move toward the current optimum solution (prey) and update their location, presuming that the current optimal solution is the global optimal solution or near to it. More precisely, as the value of $\vec{A}$ decreases($\vec{A} <$ 1), the search agents move closer to the prey (exploitation), and as the value of $\vec{A}$ increases($\vec{A} \geq 1$), the search agents move away from the prey (exploration). Therefore, the value of $\vec{A}$ is responsible for the choice of exploration or exploitation. Furthermore, as $\vec{a}$ decreases linearly from 2 to 0 during the iterations and $\vec{A}$ is a random value in range $[\vec{-a}, \vec{a}]$, then when $\vec{a}$ reaches the value 1, the value of $\vec{A}$ will be in the range [-1,1]. This will stop the algorithm to run the exploration part.

The following equations describe the behavior of encircling prey [11]:

$$\vec{D} = ||\vec{C} \cdot X^*(t) - X_i(t)|| \tag{6}$$

$$X_i(t+1) = X^*(t) - \vec{A} \cdot \vec{D} \tag{7}$$

where $X^*(t)$ is the best whale (best solution) until iteration t.

2) Spiral-shaped trajectory updating position: At this stage the whale moves around the prey by using a spiral pattern. The following equations describe this behavior [11].

$$\vec{D'} = ||X^*(t) - X_i(t)|| \tag{8}$$

$$X_i(t+1) = \vec{D'} \cdot e^{bl} \cdot cos(2\pi l) + X^*(t) \tag{9}$$

where $\vec{D'}$ is the distance between two whales, b is a constant used to define the logarithmic spiral's shape and l is a random number in [-1,1].

It is worth noting that the whales swim around the prey in a shrinking circle and a spiral-shaped trajectory at the same time. To mimic this concurrent behavior, we assume that there is a 50% chance of selecting either the shrinking encircling mechanism or the spiral model to update the position of whales throughout the optimization. The mathematical model of updating the position of whale using random search for prey, encircling prey and the spiral-shaped trajectory are summarized as follows [11].

$$X_i(t+1) = \begin{cases} \text{If } p < 0.5 & \begin{cases} \text{If } |A| \geq 1 & \text{Eq. (1) and Eq. (2)} \\ \text{If } |A| < 1 & \text{Eq. (6) and Eq. (7)} \end{cases} \\ \text{If } p \geq 0.5 & \text{Eq. (8) and Eq. (9)} \end{cases} \tag{10}$$

where p is a random number in range [0,1].

V. CHAOS AND WOA BASED ALGORITHM (CWOA)

Despite having a better convergence rate, WOA is still unable to outperform in terms of avoiding local optimum, which impacts the algorithm's convergence rate. So, in order to alleviate this effect and increase its efficiency, an enhanced WOA algorithm is proposed by combining chaotic map [17] with the original WOA algorithm.

Two features of chaotic maps are ergodicity and non-repeatability. These two characteristics allow a rapid convergence of meta-heuristic optimization methods by efficiently exploring the search space and helping to avoid local optimum.

In order to benefit from the advantage of the chaos map, this paper considers the combination of the hybrid chaos map (Kent map + Logistic map), proposed in [17], with the original WOA. The hybrid Kent map and Logistic map equations are both described in [17].

The chaotic number $z(t)$ produced using map in [17] will be used for updating the j^{th} randomly chosen dimension of the position of the whale i as follow.

$$X_{ij}(t+1) = lb_j + z(t) \times (ub_j - lb_j) \tag{11}$$

where lb_j and ub_j are respectively the lower and upper bounds of the j^{th} dimension.

To sum up, the CWOA starts by the initialization of all whales (solutions), selects the best whale as the leader of the population, then divides the population into two sub-populations. For the first sub-population, the position of the leader is assigned to each whale, then a random dimension j is chosen to rotate the current position of the whale by only one chosen dimension X_{ij} using the Equation (11). As for the second sub-population, the procedure of the original WOA is applied (see Equation (10)).

Algo	**avg**	**time**	**worst**	**best**	**std**	**#SN**	**#RN**	**D**
Number of targets: 10 targets								
WOA	17,1	140	18,9	13,3	1,59	9,7	30,8	10,9
CWOA	12,9	94,5	16	11,7	1,17	8,9	20,2	9,8
OPT	**9.31**	**7.05**				**8**	**13**	**7**
Number of targets: 30 targets								
WOA	21,4	137	26,5	17,9	2,49	15,3	38,2	10,9
CWOA	15,5	**99,7**	19	13	1,6	14,6	22,7	9,4
OPT	**10.64**	104.83				**11**	**13**	**8**
Number of targets: 50 targets								
WOA	26,2	167	28,9	21,2	2,14	21,4	46,8	11
CWOA	18,4	**107**	20,6	16,6	1,16	18	27,4	9,9
OPT	**12.28**	202				**16**	**14**	**7**
Number of targets: 80 targets								
WOA	29,7	162	35,7	24,5	3,6	28,8	50,8	10,1
CWOA	20,8	**114**	22,9	19,2	0,98	25	28,3	9,5
OPT	**14.26**	1484.4				**19**	**17**	**7**
Number of targets: 100 targets								
WOA	30,8	169	33,1	28,8	1,27	29,4	53,4	10,2
CWOA	21,8	**127**	23,5	20,2	1	27,4	28,3	10
OPT	**14.27**	446.1				**20**	**15**	**8**

TABLE I: Targets number variation in a (12×12) grid

Algo	**avg**	**time**	**worst**	**best**	**std**	**#SN**	**#RN**	**D**
	Grid size: 6×6							
WOA	7,56	28,9	8,62	6,64	0,67	11,2	7,7	**3,9**
CWOA	**7,23**	**20,9**	**8,29**	**5,97**	**0,57**	**10,6**	**7,3**	**3,9**
	Grid size: 8×8							
WOA	13,5	46,1	15,2	11,9	1,02	15,5	18,7	6,6
CWOA	**11,2**	**34,4**	**12,3**	**9,96**	**0,61**	**15,4**	**12,3**	**5,9**
	Grid size: 10×10							
WOA	22,5	105	25,8	19,9	1,98	23,9	35,4	**8,5**
CWOA	**17**	**76,7**	**18,6**	**14,9**	**1,13**	**21,5**	**20,7**	9
	Grid size: 12×12							
WOA	30,8	169	33,1	28,8	1,27	29,4	53,4	10,2
CWOA	**21,8**	**127**	**23,5**	**20,2**	**1**	**27,4**	**28,3**	**10**
	Grid size: 16×16							
WOA	62,9	757	71,4	52,6	6,43	55,2	123	12,4
CWOA	**31,1**	**386**	**33,1**	**28,8**	**1,48**	**37,6**	**44,2**	**12,1**

TABLE II: Grid size variation with 100 Targets

VI. EXPERIMENTAL RESULTS AND ANALYSIS

To show the effectiveness of CWOA, a series of experiments have been conducted. The original WOA [11] and the exact method (OPT for short) proposed in [3] were chosen for comparative study. In all the experiments, we set the size of the population to 30, the maximum number of iterations to 500. Moreover, for each dataset, all algorithms (WOA, CWOA and exact method) are individually performed 10 times. All the experiments were performed using Python 3.7 on a computer having Intel(R) Core(TM) i5-8250U CPU @ 2.50 GHz with 6GB RAM.

At this step, in the absence of real datasets, we generate graphs composed of CSPs and CRPs with different grid sizes $\{6\times6,\ 8\times8,\ 10\times10,\ 12\times12,\ 16\times16\}$ and different number of targets randomly disseminated over the square cells of the grid $\{10,\ 30,\ 50,\ 80,\ 100\}$.

The following metrics have been chosen to evaluate the performance of the three compared algorithms: average of fitness value (avg), running time (time), worst of fitness value (worst), best of fitness value (best), standard deviation (std), sensor number (#SN), relay number (#RN) and network diameter (D).

Finally, we notice that we have set α, β and γ to the same value, namely $\frac{1}{3}$. However, the proposed algorithm can be used for any combination of weights.

A. Impact of the variation of targets density and the network size

Tables I and II show that the proposed algorithm CWOA outperforms the original WOA and finds very close results to the exact method (OPT) in all instances. More accurately, from Table I, we can note that the #SN and #RN required respectively for coverage and connectivity increases when the number of targets increases. However, the proposed CWOA has always the best value in terms of #SN, #RN, and network diameter than the original WOA.

It is noteworthy that as the number of targets increases, the difference between CWOA and WOA in terms of #SN and #RN increases, which demonstrates the effectiveness of the proposed CWOA. On the other side, the CWOA finds the better value in terms of best, worst and std values proving the robustness of the proposed CWOA.

Table II shows the performance of the proposed CWOA when the network size increases. We can see that for all evaluation metrics, CWOA is the better one for all instances. Mostly for the #RN, as the network size increases the difference becomes very large (x3).

Figure 3 displays the average fitness over 10 runs. Compared with the optimal solution, we can note that the CWOA is very competitive since it is much closer to the optimal solution, whereas the original WOA is so far from the optimal solution. Furthermore, the proposed algorithm keeps the gap between it and the optimal solution, while the WOA, increases its gap as the density of targets increases. Therefore, we can predict that as the problem is complex, the WOA becomes worst. Figure 4 shows the average of the fitness values when the network size increases over the 10 runs. We can see that when

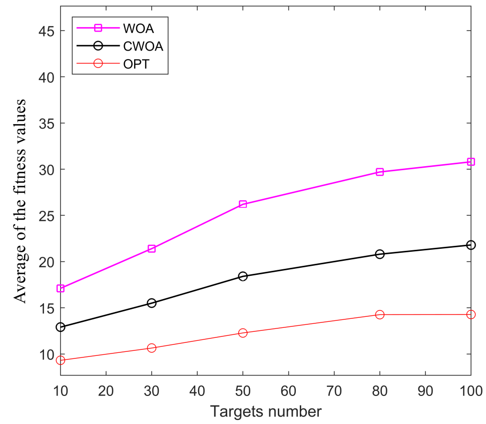


Fig. 3: Average of fitness values according to the variation of targets number in a (12×12) grid

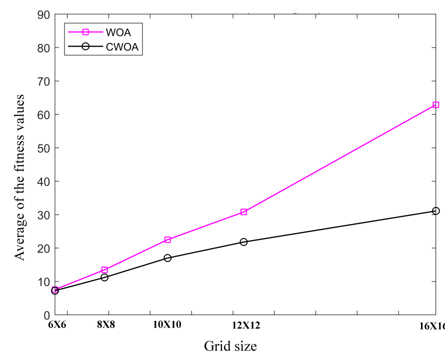


Fig. 4: Average of fitness values according to the variation of the grid size (targets number=100)

the network size is small, CWOA and WOA produce close solutions and as the network size increases, the WOA fails to find good solutions compared to CWOA, which prove the incapability of the WOA to escape a local optimum. Therefore, the integration of the chaos theory in WOA has a good impact to help the original WOA to converge toward more optimal solutions. Thus, the CWOA becomes more efficient when the network size increases.

B. Evaluation of the running time

Figure 5 and Figure 6 show the average running time over 10 executions. As depicted in Figure 5, the running time of the exact algorithm (OPT) increases exponentially with the augmentation of the number of targets while the meta-heuristics based algorithm show their efficiency. Furthermore, the proposed CWOA becomes the best one as the complexity of the problem increases, where the ratio solution quality/ running time is the lowest one for the CWOA compared with the exact method and the WOA.

The Figure 6 plots the running time when the network size increase. We can see that, as the network is large as the running

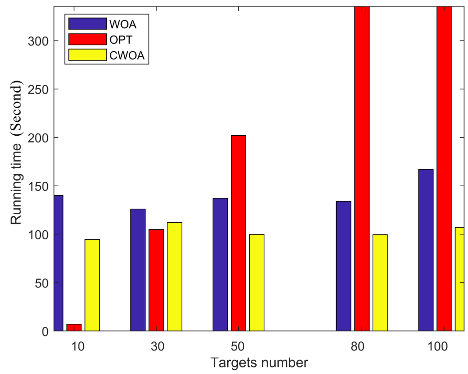


Fig. 5: Running time according to the variation targets number in a (12×12) grid

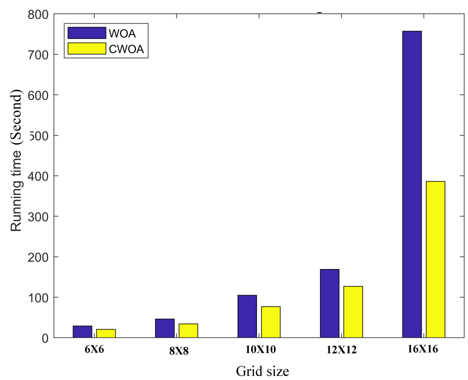


Fig. 6: Running time according to the variation of the grid size (targets number=100)

time of CWOA becomes half of the running time of WOA, which demonstrates that the proposed CWOA is not very affected by the increase of the network size compared with the WOA. Finally, We can deduce from Table II and Figure 6 that CWOA is the suggested algorithm for the optimization of the deployement of WSNs, as it achieves a better ratio solution quality/running time compared to WOA.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have investigated the optimal placement of WSNs for fire surveillance in a SCP by simultaneously minimizing the number of SNs, RNs, and network diameter while meeting coverage and connectivity requirements. As this problem is NP-hard, we have proposed an enhanced Whale Optimization Algorithm (WOA) based on chaos theory in order to find better result in reasonable time comparatively to the original WOA and the exact solution proposed in [3].

We have conducted extensive tests on several instances with the augmentation of the number of targets and the network size. The proposed algorithm has shown interesting results in terms of computational time and quality of solution. As future work, we plan to consider the usage of a realistic physical layer model to take into account obstacles inside SCPs.

REFERENCES

[1] Amir H Alavi, Pengcheng Jiao, William G Buttlar, and Nizar Lajnef. Internet of things-enabled smart cities: State-of-the-art and future trends. *Measurement*, 129:589–606, 2018.

[2] Ouamri Mohamed Amine, Zenadji Sylia, Khellaf Selia, and Azni Mohamed. Optimal base station location in lte heterogeneous network using non-dominated sorting genetic algorithm ii. *International Journal of Wireless and Mobile Computing*, 14(4):328–334, 2018.

[3] Slimane Charafeddine Benghelima, Mohamed Ould-Khaoua, Ali Benzerbadj, and Oumaya Baala. Multi-objective optimisation of wireless sensor networks deployment: Application to fire surveillance in smart car parks. In *2021 International Wireless Communications and Mobile Computing (IWCMC)*, pages 98–104, 2021.

[4] Rabab Bousmaha, Reda Mohamed Hamou, and Abdelmalek Amine. Automatic selection of hidden neurons and weights in neural networks for data classification using hybrid particle swarm optimization, multi-verse optimization based on lévy flight. *Evolutionary Intelligence*, pages 1–20, 2021.

[5] Mihaela Cardei, My T Thai, Yingshu Li, and Weili Wu. Energy-efficient target coverage in wireless sensor networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 1976–1984. IEEE, 2005.

[6] Slaheddine Chelbi, Habib Dhahri, and Rafik Bouaziz. Node placement optimization using particle swarm optimization and iterated local search algorithm in wireless sensor networks. *International Journal of Communication Systems*, 34(9):e4813, 2021.

[7] Jehan Chelliah and Navaz Kader. Optimization for connectivity and coverage issue in target-based wireless sensor networks using an effective multiobjective hybrid tunicate and salp swarm optimizer. *International Journal of Communication Systems*, 34(3):e4679, 2021.

[8] R Deepa and Revathi Venkataraman. Enhancing whale optimization algorithm with levy flight for coverage optimization in wireless sensor networks. *Computers & Electrical Engineering*, 94:107359, 2021.

[9] C Jehan and D Shalini Punithavathani. Potential position node placement approach via oppositional gravitational search for fulfill coverage and connectivity in target based wireless sensor networks. *Wireless Networks*, 23(6):1875–1888, 2017.

[10] Trista Lin, Hervé Rivano, and Frédéric Le Mouël. A survey of smart parking solutions. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3229–3253, 2017.

[11] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in engineering software*, 95:51–67, 2016.

[12] Chandra Naik and D Pushparaj Shetty. Optimal sensors placement scheme for targets coverage with minimized interference using bbo. *Evolutionary Intelligence*, pages 1–15, 2021.

[13] Samineh Nasrollahzadeh, Mohsen Maadani, and Mohammad Ali Pourmina. Optimal motion sensor placement in smart homes and intelligent environments using a hybrid woa-pso algorithm. *Journal of Reliable Intelligent Environments*, pages 1–13, 2021.

[14] Saunhita Sapre and S Mini. Moth flame optimization algorithm based on decomposition for placement of relay nodes in wsns. *Wireless Networks*, 26(2):1473–1492, 2020.

[15] Yahui Sun and Saman Halgamuge. Minimum-cost heterogeneous node placement in wireless sensor networks. *IEEE Access*, 7:14847–14858, 2019.

[16] Chun-Wei Tsai, Pei-Wei Tsai, Jeng-Shyang Pan, and Han-Chieh Chao. Metaheuristics for the deployment problem of wsn: A review. *Microprocessors and Microsystems*, 39(8):1305–1317, 2015.

[17] Oguz Emrah Turgut. Solving time-dependent heat conduction problems using metaheuristic algorithms extended with a novel local search strategy. *SN Applied Sciences*, 3(1):1–25, 2021.

[18] Jiazu Xie, Baoju Zhang, and Cuiping Zhang. A novel relay node placement and energy efficient routing method for heterogeneous wireless sensor networks. *IEEE Access*, 8:202439–202444, 2020.