Journal of Theoretical and Applied Information Technology
31st July 2019. Vol.97. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

WIRELESS SENSOR NETWORK DEPLOYMENT BASED ON MACHINE LEARNING FOR PROLONGING NETWORK LIFETIME AND PDR

ALI NOORI[1], ONG BI LYNN[2], HASNA AHMAD[3], R BADLISHAH AHMAD[4], AMIZA AMIR[5], AYMEN ABD[6], MARWA K IBRAHIM[7]
[1]School of Computer and Communication Engineering University Malaysia Perlis (UniMAP) 02600, Arau, Perlis, Malaysia
E-mail: [1]3li.noori@gmail.com

ABSTRACT

Sensor Deployment (SN) is one of the major challenges in wireless sensor network architecture. One of the most fundamental issues in wireless sensor deployment is to balance the objective to resolve network conflicts. This paper aims to find the Pareto front that maximizes the packet delivery ratio and minimizes sensor energy consumption for prolonging network lifetime. For this proposal, a hyper-heuristic framework for improving the performance of the metaheuristic (LMOJPSO) search optimization process by combining two different searching techniques was designed. The first optimization technique carried out its searches with the help of an extreme learning machine (ELM), whereas the second used a wireless sensor network simulator. In this paper, the proposed method is examined in given wireless sensor network test instances, and the evaluation of its performance is carried out using a WSN performance metric. The results indicate that the proposed model is superior to the non-dominated sorting genetic algorithm (NSGA-II).
Keywords: *Wireless Sensor Network Deployment, NSGA-II, Hyper-heuristic, PSO, Optimization*

1. INTRODUCTION

A Wireless Sensor Network (WSN) consists of many sensor nodes, a processor, a sensory board, a radio, and a battery. All these components help the sensors carry out different sensing, processing, and communicating tasks within a fixed coverage radius. During network operation, the sensor nodes collect all data from the situation under analysis and broadcast it towards the sink node using a multi-hop communication process [1]. These networks are very helpful for accessing harsh or remote areas and for observing the situations in such locations in a cost-effective manner. The coverage of the WSND depends on the number of sensors used [2], along with their position in the area to be monitored. Therefore, to maximize the spatial coverage of these networks, various optimization algorithms have to be applied for determining the best location of every sensor within the network [3,4]. Sensor coverage is a one of fundamental problem in WSNs[5], one which must be considered when improving connectivity and network configuration and for energy conservation. The researchers used optimization algorithms to address issues related to sensor deployment and improve the coverage of the area. They ensured that every region was monitored by at least one sensor node. Good coverage and security is important for an effective WSN [6], [7]. According to many researchers, major WSND issues include the placement of a minimal number of sensors to achieve maximal coverage, maintenance of the best network connectivity, and low energy consumption [8]. Besides, some also stated that the energy consumption costs, implementation, and maintenance costs have to be considered, since the manner in which the sensors are deployed could significantly affect the cost and flexibility of the WSN. One needs to optimize the trade-off between coverage quality and implementation costs before deploying all sensors. Furthermore, the lifetime of a network (i.e., the amount of time which passes before the first sensor in any network becomes inoperative) must also be determined. Energy consumption has to be effectively optimized for increasing the lifetime of a network.

Metaheuristic algorithms offer alternative processes for determining optimal solutions in a short period, whereas other algorithms require a long time to converge effectively [9]. Every metaheuristic algorithm displays characteristic features. Hence, no single metaheuristic algorithm can solve all optimization problems and provide the best result with regards to outcome quality and computation time. In the past few years, researchers have tried to influence the strengths of various

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

metaheuristics through hybrid-heuristic algorithms [9, 10,11, 12]. For instance, one study combined the strengths of GA (i.e., global search) and SA (i.e., local search) and developed a novel hybrid-heuristic algorithm that can balance both local and global searches during the convergence procedure [14]. It must be noted that this integration offers 2 or more metaheuristic outcomes at each iteration. Generally, hybrid-heuristic algorithms require a longer period to provide better results than the single metaheuristic algorithm. To resolve the issue that hybrid metaheuristic algorithms are more computationally expensive than metaheuristic algorithms, some researchers [14,15,11] proposed the hyper-heuristic algorithm. The main idea behind these algorithms was to select a suitable metaheuristic algorithm for determining the best possible outcome for a fixed number of iterations at varying periods during the convergence process. Thereafter, they use a different metaheuristic algorithm to find probable output when the earlier process is stuck in some area. the hybrid-heuristic algorithm consists of a pool of metaheuristics generally known as Low-Level Heuristics (LLH). In comparison to the hybrid-heuristic algorithms, which use 2 or more metaheuristic algorithms for determining the possible outcome at every iteration, the hyper-heuristic algorithm selects a single metaheuristic (LLH) to acquire the best solution. Hence, hyper-heuristic algorithms were seen to be faster than hybrid-heuristic algorithms for every iteration. Similarly, to hybrid-heuristic algorithms, hyper-heuristic algorithms are able to control the strength of the various metaheuristics. Thus, the hybrid-heuristic and hyper-heuristic algorithms provide better solutions than metaheuristic algorithms alone in a majority of cases.

In this study, the researchers aimed to solve the metaheuristic problem using a novel hyper-heuristic algorithm. For this purpose, they applied a machine learning model for predicting the fitness values of random solutions which were generated by a metaheuristic optimization process.This improved the metaheuristic algorithm and elevated it to an advanced search level in the whole solution space.

Research contribution:

1. A hybrid technique consisting of metaheuristics and an extreme learning machine algorithm has been proposed, designed, and analysed. This hybrid technique has been used to further increase network lifetime and maximise packet delivery ratio.
2. An extreme learning machine approximation of WSND simulators using sampled solutions and feature extraction has reduced searching time.
3. The proposed system has been implemented, analysed, and compared with the state-of-the-art multi-objective optimization approach.

The remaining study is organized as follows. Section 2 describes related studies, while Section **Error! Bookmark not defined.** provides the deployment problem formulation, while the methodology is described in Section3960. Section 5 evaluation scenario, while Section 6 presents all results and conclusions in section 7.

2. RALETAD WORKS

Many researchers prefer applying metaheuristic approaches to WSND. Metaheuristic algorithms can offer alternative processes for determining optimal and approximate solutions in a short time period, in comparison to other algorithms, which require a long time to converge effectively. Furthermore, unlike the deterministic and greedy algorithms which fall into the local optimum during their early iterations, metaheuristic algorithms apply various mechanisms for avoiding the local optimum [9]. Many studies have used a metaheuristic approach for WSND. [17] applied an improved Artificial Bee Colony (ABC) metaheuristic algorithm, which displayed a good solution search equation that improved the network's exploitation capability. Furthermore, they also introduced a better population sampling method that used a Student's-t distribution to increase the global convergence of their proposed metaheuristic algorithm. This algorithm maintained a balance between the exploitation and exploration abilities of the network and a minimal memory requirement. Furthermore, it required a compact Student's-t distribution test, which increased its applicability to WSNs. The researchers also introduced an energy-efficient clustering protocol, i.e., the Bee Cluster, which was based on the ABC metaheuristic algorithm. This protocol applied a metaheuristic algorithm to acquire the optimal Cluster Heads (CHs) and improve the energy efficiency of the WSNs. Ahmed et al.[18] applied an Integer Linear Programming (ILP)-based Genetic Algorithm (GA) to improve NDSC scheduling and maximize WSN lifetime. The researchers used a particular arrangement of chromosomes, combining various crossover and mutation strategies in order to encode for the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

solutions. [19] proposed and developed a cuckoo search–based, energy-balanced node-clustering protocol that used a novel objective function for a uniform distribution of the CHs. They also proposed a better harmony search–based routing protocol for routing the data packets between the CHs and the sinks. [20] proposed an improved ABC-based deployment algorithm. This algorithm was able to increase the lifetime of the network by optimizing all network parameters and restricting the number of deployed relays. In another study, Krishna and Doja [21] proposed a multi-objective metaheuristic approach for an Energy-Efficient Secure Data Aggregation (MH-EESDA) protocol in WSNs. This protocol applied the divide-and-conquer process to develop a novel approach for securing clusters and developing a secure data aggregation in the energy-efficient routes of the WSN. This algorithm acts in 3 phases. Clusters are developed in Phase 1, the secure nodes get selected in Phase 2, and energy-efficient data aggregation is carried out across all secure routes of the network in Phase 3. The earlier studies indicated that a metaheuristic search was a common feature of all approaches. However, the major drawback of metaheuristic search optimization is its dependency on the population size and number of iterations.

However, metaheuristic searching optimization algorithms such as NSGA-II [22] only include interaction between solutions within one iteration or two consecutive iterations, which might cause well-fitting solutions to be forgotten as the algorithm runs. The lack of an accurate model of the WSN that can be used for calculating the objective functions values of the solutions is also problematic. Various researchers use simulators [23],[24] to substitute for the objective function, which lengthens the optimization process and makes it impossible to increase both the number of generated solutions within one iteration and the number of iterations, which in turn affects the coverage of the search in the solution space.

3. METHOD

Here, the researchers describe the methodology used for carrying out a hyper-heuristic deployment. The general architecture is presented in Sub-Section 1.3, while the LMOJPSO show in Sub-Section 2.3, and Extreme machine learning presented in 3.3. Finally, feature extraction is described in Sub-Section 5.3.

1.3 General Architecture

In (Figure 1) presents the dataset for the hyper-heuristics based WSND. This dataset consists of various processes like sampling and feature extraction, along with 2 simulators. The sampling process generates random candidate solutions for WSND. Thereafter, the feature extraction process extracts the geometrical and communication features from this solution. Every solution was tested using two simulators, S1 for a Lifetime-based routing protocol that describes the energy-rich route. Using the Dijkstra algorithm, the shortest route routing is introduced in the second simulator a packet delivery ratio-based routing protocol that describes the shortest route.

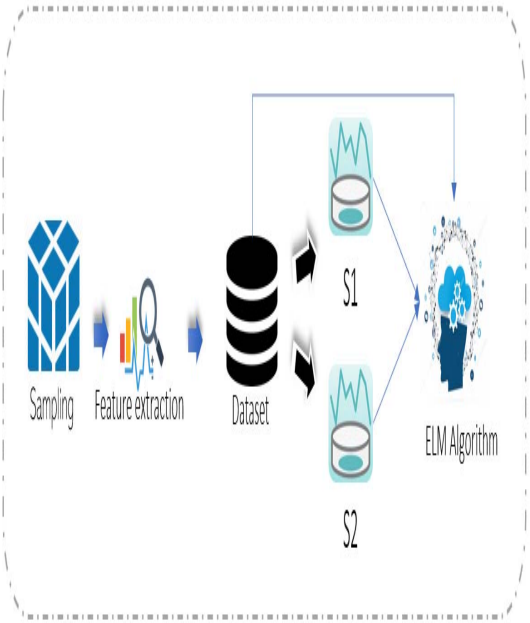


Figure 1. Block diagram of hyper-heuristics based WSND

Additionally, Figure 2 a general architecture for hyper-heuristic based WSND. This dataset was used for training the extreme learning machine (ELM), which was connected to the lagged multi-objective jumping particle swarm optimization LMOJPSO to carry out the first stage of optimization. The results of the first stage included an initial population, which was then used with the LMOJPSO to carry out another optimization, in which the two simulators were used to provide the final set of non-dominated results. The following subsections explain the remaining sub-blocks in further detail.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

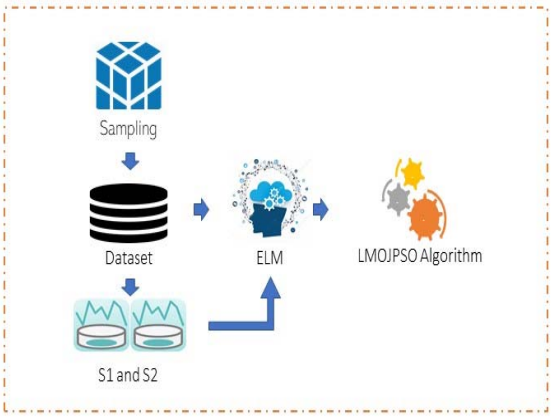


Figure 2 Block diagram of hyper-heuristics-based WSND using LMOJPSO

2.3 LMOJPSO

The pseudocode of LMOJPSO "**lag multi objective jumping particle swarm optimization**" is given in Table 1**Error! Reference source not found.**. As shown, the solution moves towards one of the three Pareto distributions according to the values given to the constants C_1 and C_2. As the Pareto is not a single solution, a random selection is carried out to choose one solution from each Pareto in order to have a subject solution moving forward. After the solution is updated, its local Pareto is also updated. Then, when all solutions in the iteration have finished, we update the global Pareto and iteration Pareto. The combination of two solutions is based on the logic of moving a subject solution towards the target solution. The solution moves towards the target with a random velocity. Overall, C_1, C_2 are control parameters for the number of solutions and the speed with which every solution moves towards the local, global, and iteration Pareto.

The LMOJPSO [4] was developed to determine the best non-dominated solution set with the help of three Pareto designs, i.e., local Pareto, iteration Pareto, and global Pareto. The solution generated from every iteration interacted with the three Pareto designs to maintain avoidance of the local minima and preserve the elite solutions. This algorithm can also present a customized interaction between WSND solutions, which could increase its search capacity. This algorithm is lagged as it provides the concept of an iteration Pareto, a Pareto over the existing Pareto, and includes the Lag of iteration, i.e., L. This feature distinguishes the LMOJPSO from the NSGA-II process, since all solutions in the NSGA-II interact with the same-generation solutions and not with solutions from other generations.

Table 1. Pseudocode of LMOJPSO

```
Input:
Number Particles in The Swarm,
Maximum Number of Iterations,
c1, c2
Output
Pareto
Start
Initialization ()
Evaluation ()
Global Pareto Front= []
Local Pareto Front= []
Iteration Pareto Front= []
Particles in The Swarm
loop =true
Iter =1, sol =1, lag while loop        r= random
                        if 0<r<c1
                            Target=uniform
Random Select (global Pareto Front, lag)
            else
                        if c1<r<c2
                            Target=uniform
Random Select (local Pareto Front)
            else
                            Target=uniform
Random Select (iteration Pareto Front)
          end if
          end if
          New Solution =Combine (particles In
the Swarm(sol), Target)
          Evaluate (New Solution)
                            Update (local
Pareto Front)
          sol = sol + 1
          if sol > number Particles in The Swarm
                            Update (global
Pareto Front)
                            Update (iteration
Pareto Front)
                  Iter= iter + 1
                  if iter > maximum Number of
Iterations
                            return global Pareto
Front
            else
                            sol = 1
end if
end while
```

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
end
```

3.3 Sampling from solution space

The goal of sampling is to generate a wide range of possible solutions for deployment. The generated sampling solutions provide a general image of the nature of the optimization surface. For this purpose, uniform random solutions were generated, ranging from the minimum to the maximum value of each element in the deployment solutions. The pseudocode for the sampling algorithm is shown in Table 2.

Table 2. The pseudocode for the sampling algorithm.

```
Input
Xmin
//the minimum level of the x coordinates of the environment
Xmax
//the maximum level of the x coordinates of the environment
Ymin
//the minimum level of the y coordinates of the environment
Ymax
//the maximum level of the y coordinates of the environment
Rcmin
//the minimum level of the connectivity radius
Rcmax
//the maximum level of the connectivity radius
Rsmin
//the minimum level of the sensing radius
Rsmax
//the maximum level of the sensing radius
N
//size of dataset
Output
D
//Sampled Dataset
Start
D=[];
for Index=1 until N
x=rand(Xmin,Xmax)
y=rand(Ymin,Ymax)
Rc=rand(Rcmin,Rcmax)
Rs=rand(Rsmin,Rsmax)
D.add([x y Rc Rs])
End
```

Each generated solution represents a potential solution for WSND. The solution will be tested in both simulators S1 and S2 in order to calculate its PDR and lifetime. Thus, we arrange the dataset according to records, where each record R_i =

$[solution_i\ lifetime_{1,i}\ PDR_{1,i}\ lifetime_{2,i}\ PDR_{2,i}]$

, where

$solution_i$ denotes solution number i

$lifetime_{1,i}$ denotes the lifetime of solution i when tested on simulator S1

$PDR_{1,i}$ denotes the packet delivery ratio for solution i when tested on simulator S1

$lifetime_{2,i}$ denotes the lifetime of solution i when tested on simulator S2

$PDR_{2,i}$ denotes the packet delivery ratio for solution i when tested on simulator S2

4.3 Extreme Learning Machine

The extreme learning machine can predict objectives based on all provided features. The researchers developed a Single-Hidden-Layer-Feed-forward-Neural-network (SLFN) which consists of (L) hidden neurons along with a $g(x)$ activation function. The SLFN was trained using the datasets generated by the two simulators, i.e., D_1 and D_2. Table 3 describes the training process used in the study:

Table 3. Training Dataset using ELM

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Input: Dataset
$(features_i, Measure_{i,l})$
$Measure_{i,l}$ denotes lifetime or PDR for feature i
$features_i$ denotes the features that are extracted from the sampling data and provided to the simulator to obtain the measure
Output:
trained ELM
Step 1: Generate input-hidden layer weights of ELM
Step 2: Calculate the output matrix H of ELM using an activation function $g(x)$
Step 3: Calculate hidden output weight using Moore-Penrose

5.3 Feature Extraction

This section describes all solutions extracted from the deployment solution. In total, 18 features were extracted and are described in Table 4. All features are 1-D, indicating that the total number of features included a vector with 18 elements, described below:

Table 4. Feature Etraction

Feature no	Feature symbol	Feature meaning
1	N_{st}	Number of sensing intersections
2	A_{st}	Area of sensing intersection
3	avg_{dn}	Average of distances between nodes
4	avg_{ds}	Average of distances between nodes and sink
5	N	Number of nodes
6	N_c	Number of connections to the sink
7	Avg_{pns}	Average of path node numbers to the sink.
8	max_{pns}	Maximum path node number to the sink
9	min_{pns}	Minimum path node number to the sink
10	Avg_{pls}	Average of path lengths to the sink
11	max_{pls}	maximum path length to the sink
12	min_{pls}	Minimum path length to the sink
13	$stdR_s$	The standard deviation of connectivity radius over distance to the sink
14	Avg_{rds}	Average of connectivity radius over distance to the sink
15	Avg_x	Average of the X coordinates
16	Avg_y	Average of the Y coordinates
17	Avg_{sr}	Average of sensing radius
18	Avg_{cr}	Average of connectivity radius

4. Evaluation Scenarios and all Measures

In a multi-objective optimization problem, the quality of the Pareto front is important. In this context, generating solutions close to the Pareto front and maintaining variety in non-dominated solutions are considered. However, evaluation measures are needed to conduct a multi-objective optimization; there exist various performance metrics to measure the qualities of multi-objective problem approaches.

The first measure includes the set coverage metric or a C-metric. It also uses the input as two optimal sets and generates a set coverage metric as the final output, which is evaluated as follows:

$$C(P_{s1}, P_{s2}) = \frac{|\{y \in P_{s2} | \exists x \in P_{s1} : x \succcurlyeq}{|P_{s2}|} \quad (1)$$

Where C = ratio of the non-dominated solutions in P_{s2} (which was dominated by the non-dominated results in P_{s1}) to the number of solutions in P_{s2}. Hence, during the evaluation of a set PS, one must minimize the $C(x, P_s)$ value when x is a different Pareto set.

The second measure includes the hypervolume metric (HV-metric or S-metric) value, used in evolutionary multi-objective optimization to evaluate the performance of all search algorithms. It calculates the volume of the dominating amount of the objective space with regards to the reference point. A higher value for this metric indicates that the solutions are more desirable. Furthermore, the hypervolume indicator measures the convergence to the actual Pareto front and the diversity of all derived solutions. It is calculated as follows:

Journal of Theoretical and Applied Information Technology
31st July 2019. Vol.97. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$I_H^n(A) = \int_{(0,0,\ldots,0)}^{(1,1,\ldots,1)} \alpha_A(z)dz \quad (2)$$

$$\alpha_A(z) = \begin{cases} 1 & if\ A \succcurlyeq \{z\} \\ 0 & else \end{cases}$$

The final measure includes the number of non-dominated solutions that demonstrate the ability of the algorithm to derive solutions, and is estimated based on the size of the Ps in the following manner:

$$NDS(N) = |P_S| \quad (3)$$

A higher value of NDS is desirable, as it indicates that the number of solutions is adequate.

5. EXPERIMENTAL RESULTS

Pareto optimal solutions or non-dominated solutions are the terms used for the solutions that are not subdued by other solutions, while the Pareto set is the name given to the collection of all of the Pareto optimal solutions, which are mapped to a Pareto front.

A pareto front sample is shown in Fig. 3. This figure shows the intrinsic conflict between the lifetime and packet delivery ratio objective. This study generated a Pareto front for a hybrid algorithm–based LMOJPSO deployment and compared it to other approaches, i.e., simulation-based deployment LMOJPSO, hybrid NSGA-II, and simulation-based NSGA-II.

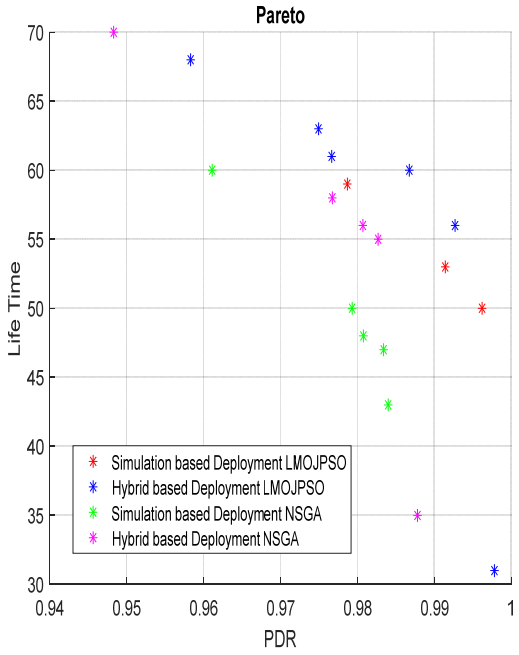


Figure 3. Pareto front for the hybrid-based deployment LMOJPSO

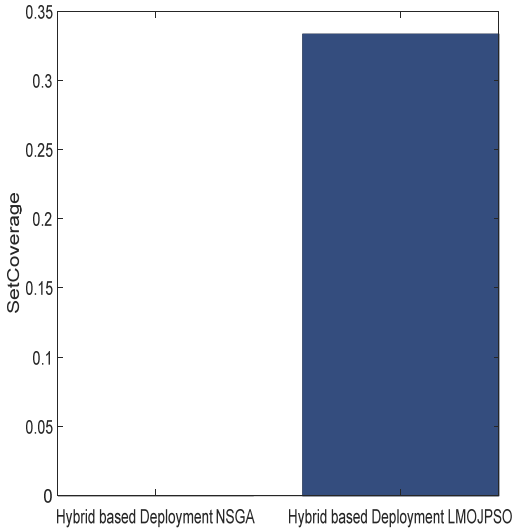
The Pareto front has been generated for the hybrid-based deployment LMOJPSO developed for this study and has been compared with three approaches: simulation-based deployment LMOJPSO, hybrid-based NSGA-II, simulation-based NSGA-II. However, this simulator aspires to achieve the maximum lifetime and packet delivery ratio; in Figure 3, the results clearly show the hybrid-based deployment obtains the best values and achieves the maximum lifetime when compared to all three approaches mentioned previously.

The analysis and discussion of the simulation result of three proposed method were introduced in this article. The simulation results were analyzed to determine the effective of the hybrid performance. Furthermore, the hyper-heuristic framework was evaluated with different arrangement of its blocks and also it was found that LMOJPSO with hyper heuristic architecture is superior in terms of all MOO evaluation measures

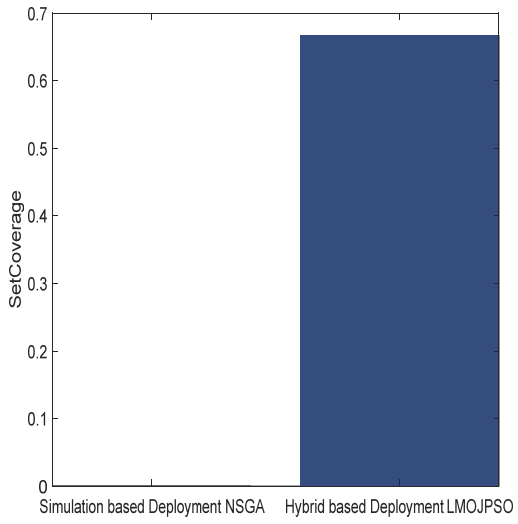
For this purpose, different performance metrics were introduced in the literature to measure approximation qualities obtained through different multi-objective optimization approaches. Because usually a single metric cannot provide sufficient outcomes to investigate the efficiency of solutions, the performance metrics list is used in following:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

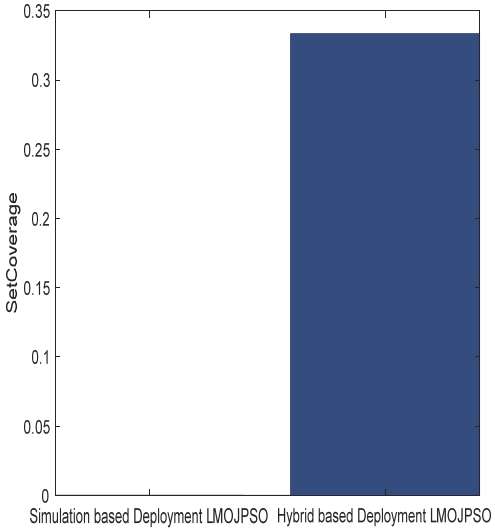
The result show compares the three approaches via the set coverage (or C-metric). In the term of set coverage Figure 4 illustrates that the hybrid based deployment LMJPSO achieved non dominated solutions better than the three approaches. These comparisons show that the selection method for pairs competition is more efficient for both LMOJPSO based deployment and hybrid LMOJPSO based deployment in the terms of C-metric.



(A)



(B)



(C)

Figure 4 Performance metrics for set coverage (A, B and C)

In addition, Figure 4 obviously demonstrates that the first LMOJPSO hybrid version exceeded the other C-metric variants. It implies that the non-dominated solutions of the hybrid variant, using the first suggested LMOJPSO as well as the pairwise tournament selection method, dominated the alternatives corresponding to the other.

In relation to the above outcomes, the hybrid LMOJPSO version is superior the hybrid NSGA-II is illustrating in Fig 4 (A). Also, Fig 4 (B) compares the non-dominated solutions for simulation-based NSGA-II with hybrid LMOJPSO. This figure shows that the hybrid MOJPSO generates better solutions than simulation-based NSGA-II. In order, it is possible to see the superiority of hybrid LMOJPSO over simulation LMOJPSO in Fig 4 (C).

As mentioned before, different performance metric has examined here. The second evaluation measure is the hypervolume (HV) depicted in Fig.5. according to figure the best value of the hypervolume metric we can clearly observe that the hybrid LMOJPSO obtained the highest value when compared to other approaches. It can be concluded that the effects of the hybrid based on machine learning that the best value of the diversity metric.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Accordingly, Fig 5 shows that the version of LMOJPSO which uses the hybrid model as well as selection method Has the best hypervolume performance.

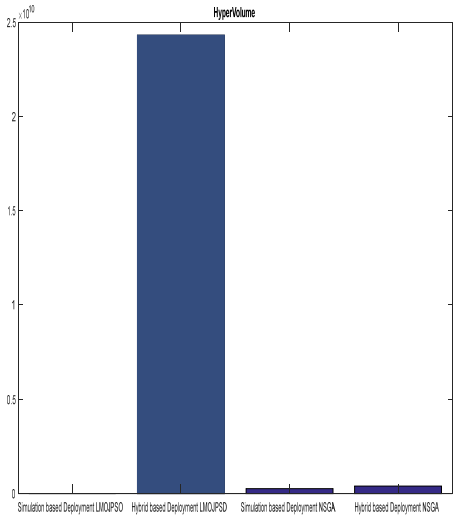


Figure 5. Performance metrics for hypervolume

The last performance metric is non-dominated solutions (NDs). As shown in, the results obtained by the LMJPSO dominated 32% of the non-dominated solutions belonging to the other approaches. This means that the hybrid NSGA-II was not able to achieve more than 25%, the simulation-based NSGA-II achieved an identical 25%, and the lowest values were for the simulation-based LMOJPSO, which found 20% of non-dominated solutions. Consequently, in terms of non-dominated solutions (NDs), one can observe the superiority of hybrid-based deployment LMOJPSO.

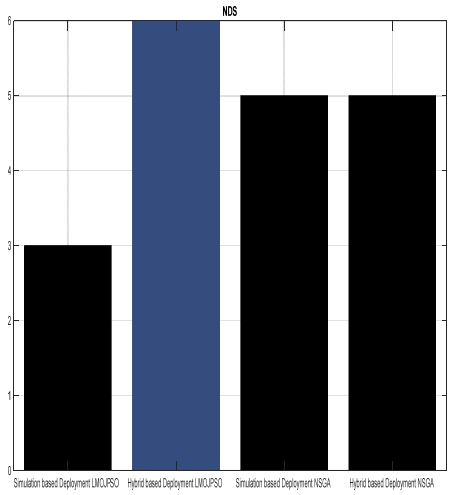


Figure 6. Performance metrics for NDs

6. CONCLUSION

The hybrid model based on machine learning was suggested in this article for the deployment of sensors. Some problem-specific operators were intended in the suggested strategy to maximize both the packet delivery ratio and lifetime while preserving the connectivity between each sensor node and the sink.

In this article we prove that the hyper heuristic approach is an excellent way use to solving the wireless sensor network deployment. Using the machine learning as substitute for the simulator which improve the optimization process and increase both the number of generated solutions within one iteration and the number of iterations, which in turn affects the coverage of the search in the solution space.

Such an approach was known as the hyper-heuristic based WSND method. The evaluation and all experimental results indicated that the proposed technique was better in comparison to the benchmark models which applied NSGA-II for their optimization and used actual simulators. The researchers also investigated different models with the help of their developed hyper-heuristic framework. These included the NSGA-II and the LMOJPSO model, which was developed in an earlier report. All results indicated that the hyper-heuristic WSND model and the metaheuristic optimization LMOJPSO model were able to significantly improve all WSND results. Further research needs to be carried out for evaluating and testing this novel approach using different machine learning models.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In future, we will Automatic selection of LMOJPSO control parameters and lag for better performance and scaling the work to include simulators with routing protocols that will be used to train the neural network.

REFERENCES

[1] F. V. C. Martins, E. G. Carrano, E. F. Wanner, R. H. C. Takahashi, and G. R. Mateus, "A hybrid multiobjective evolutionary approach for improving the performance of wireless sensor networks," *IEEE Sens. J.*, vol. 11, no. 3, pp. 545–554, 2011.

[2] H. Kim and S. W. Han, "An efficient sensor deployment scheme for large-scale wireless sensor networks," *IEEE Commun. Lett.*, vol. 19, no. 1, pp. 98–101, 2015.

[3] A. Downey, C. Hu, and S. Laflamme, "Optimal sensor placement within a hybrid dense sensor network using an adaptive genetic algorithm with learning gene pool," *Struct. Heal. Monit.*, vol. 17, no. 3, pp. 450–460, 2018.

[4] A. L. I. N. Kareem, O. N. G. B. I. Lynn, R. Ahmed, and H. Ahmad, "LAGGED MULTI-OBJECTIVE JUMPING PARTICLE SWARM OPTIMIZATION FOR WIRELESS SENSOR," vol. 97, no. 2, pp. 423–433, 2019.

[5] Z.-J. Wang, Z.-H. Zhan, and J. Zhang, "Solving the Energy Efficient Coverage Problem in Wireless Sensor Networks: A Distributed Genetic Algorithm Approach with Hierarchical Fitness Evaluation," *Energies*, vol. 11, no. 12, p. 3526, 2018.

[6] A. Chehreghan, M. Delavar, and R. Zarei, "An intelligent deployment method of geo-sensor networks in 3D environment," *Ann. GIS*, vol. 22, no. 4, pp. 301–315, 2016.

[7] M. A. Burhanuddin, A. A.-J. Mohammed, R. Ismail, M. E. Hameed, A. N. Kareem, and H. Basiron, "A review on security challenges and features in wireless sensor networks: IoT perspective," *J. Telecommun. Electron. Comput. Eng.*, vol. 10, no. 1–7, 2018.

[8] M. Khalesian and M. R. Delavar, "Wireless sensors deployment optimization using a constrained Pareto-based multi-objective evolutionary approach," *Eng. Appl. Artif. Intell.*, vol. 53, pp. 126–139, 2016.

[9] C. W. Tsai, P. W. Tsai, J. S. Pan, and H. C. Chao, "Metaheuristics for the deployment problem of WSN: A review," *Microprocess. Microsyst.*, vol. 39, no. 8, pp. 1305–1317, 2015.

[10] P. Cowling, G. Kendall, and E. Soubeiga, "A Hyperheuristic Approach to Scheduling a Sales Summit," Springer, Berlin, Heidelberg, 2001, pp. 176–190.

[11] E. G. Talbi, "A taxonomy of hybrid metaheuristics," *J. Heuristics*, vol. 8, no. 5, pp. 541–564, 2002.

[12] R. Malek, "An agent-based hyper-heuristic approach to combinatorial optimization problems," *Proc. - 2010 IEEE Int. Conf. Intell. Comput. Intell. Syst. ICIS 2010*, vol. 3, pp. 428–434, 2010.

[13] J. Grobler, S. S. Member, A. P. Engelbrecht, S. S. Member, G. Kendall, and V. S. S. Yadavalli, "Alternative hyper-heuristic strategies for multi-method global optimization," *2010 IEEE World Congr. Comput. Intell. WCCI 2010 - 2010 IEEE Congr. Evol. Comput. CEC 2010*,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2010.

[14] G. S. GAN Guo-ning, HUANG Ting-Iei, "Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment," *Iciss*, pp. 60–63, 2010.

[15] S. M. A. Bettencourt, *Lecture Notes in Computer Science*. 2004.

[16] J. Grobler, A. P. Engelbrecht, G. Kendall, and V. S. S. Yadavalli, "Alternative hyper-heuristic strategies for multi-method global optimization," *2010 IEEE World Congr. Comput. Intell. WCCI 2010 - 2010 IEEE Congr. Evol. Comput. CEC 2010*, 2010.

[17] P. S. Mann and S. Singh, "Improved metaheuristic based energy-efficient clustering protocol for wireless sensor networks," *Eng. Appl. Artif. Intell.*, vol. 57, no. November 2016, pp. 142–152, 2017.

[18] Y. E. E. Ahmed, K. H. Adjallah, R. Stock, I. Kacem, and S. F. Babiker, "NDSC based methods for maximizing the lifespan of randomly deployed wireless sensor networks for infrastructures monitoring," *Comput. Ind. Eng.*, vol. 115, no. September 2017, pp. 17–25, 2018.

[19] G. P. Gupta and S. Jha, "Integrated clustering and routing protocol for wireless sensor networks using Cuckoo and Harmony Search based metaheuristic techniques," *Eng. Appl. Artif. Intell.*, vol. 68, no. September 2017, pp. 101–109, 2018.

[20] T. Ahmad, M. Haque, and A. M. Khan, "An Energy-Efficient Cluster Head Selection Using Artificial Bees Colony Optimization for Wireless Sensor Networks," Springer, Cham, 2019, pp. 189–203.

[21] M. Bala Krishna and M. N. Doja, "Multi-Objective Meta-Heuristic Approach for Energy-Efficient Secure Data Aggregation in Wireless Sensor Networks," *Wirel. Pers. Commun.*, vol. 81, no. 1, pp. 1–16, 2015.

[22] S. T. Hasson, "Developed NSGA-II to Solve Multi Objective Optimization Models in WSNs," *2018 Int. Conf. Adv. Sci. Eng.*, pp. 19–23, 2018.

[23] A. Aksamovic, M. Hebibovic, and D. Boskovic, "Forest Fire Early Detection System Design Utilising the WSN Simulator."

[24] X. Xian, W. Shi, and H. Huang, "Comparison of OMNET ++ and Other Simulator for WSN Simulation," pp. 1439–1443, 2008.