IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 29, 2021, accepted October 13, 2021, date of publication October 25, 2021, date of current version November 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3122526

An Enhanced Multi-Objective Non-Dominated Sorting Genetic Routing Algorithm for Improving the QoS in Wireless Sensor Networks

MAHMOUD MOSHREF[1], RIZIK AL-SAYYED[2], AND SALEH AL-SHARAEH[1]
[1]Department of Computer Science, King Abdullah II School for IT, The University of Jordan, Amman 11942, Jordan
[2]Department of Information Technology, King Abdullah II School for IT, The University of Jordan, Amman 11942, Jordan

Corresponding author: Rizik Al-Sayyed (r.alsayyed@ju.edu.jo)

ABSTRACT In recent years, Wireless Sensor Networks (WSNs) have benefitted from their integration with Internet of Things (IoT) applications. WSN usage for monitoring and tracing applications shows massive acceleration, whether indoors or outdoors. WSN is constructed from interconnected sensors, limited resource (battery), which requires considerable importance on deployment and routing strategies, to improve the performance of Quality of Service (QoS) in WSNs. Many of the existing strategies are based on metaheuristics algorithms such as Genetic Algorithms to resolve the problem. This research proposes a new algorithm, Enhanced Non-Dominated Sorting Genetic Routing Algorithm (ENSGRA), to improve the QoS in WSNs. The proposed algorithm relies on Non-Dominated Sorting Genetic Algorithm 3 (NSGA-III), but adjusts reference points through the use of a dynamic weighted clustered scheduled vector to obtain new solutions. Moreover, ENSGRA can be used to find an integration between two parents crossover with multi-parent crossover (MPX), to produce multiple children and improve new offspring to obtain the optimal Pareto Fronts (PF). This algorithm excels when compared with the lagged multi-objective jumping particle swarm optimization, Non-dominated Sorting Genetic Algorithm–II and NSGA-III in terms of the QoS model (31% optimization percentage). Results show that the proposed ENSGRA is superior over other algorithms in evaluation measures for multi-objective algorithms.

INDEX TERMS Quality of service, wireless sensor networks, multi-objective algorithms, clustering, scheduling, pareto front.

I. INTRODUCTION

The importance of Wireless Sensor Networks (WSNs) come from using them in different applications, including monitoring various kinds of conditions such as temperature, humidity, pressure, vehicular movements and soil makeup. A WSN consists of a large number of low power wireless sensor nodes, which have limited transmission range and thus cannot directly send data to sink nodes that need multi-hop communication. Several communication techniques are used to connect sensor nodes with sink nodes, such as direct propagation, chain formation and cluster creation [1].

WSNs applications can be classified into two types; first for monitoring by analyzing or supervising a real-time system and second for tracking event change on a person or animal. A new important example of applications, known as IoT application based on WSNs, is a method used to extract big data from things, then mining the data to extract necessary information [2]. Hence, integrating WSNs with the Internet of Things (IoT) is considered an important and essential issue in the future.

Routing and deployment are crucial processes to consider in WSNs, especially when dealing with the performance of multiple Quality of Service (QoS) routing and deployment metrics [3]. Optimization problems are classified into two types, Single–Objective Problems (SOP) and Multi-Objective Problems (MOP). SOP aims to minimize or maximize one objective under various constraints. Selecting the most conspicuous performance metric to be optimized, SOP therefore may be improper and unreasonable for real WSN applications [2]. In MOP, the objects often conflict and clash, and the solution presents the best trade-off.

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

This result is true as conflicts and clashes can be solved using two approaches, Classical and Multi-objective optimization algorithms. The classical method aggregates the weighted sums of all objectives [2] whilst multi-objective optimization algorithms use approaches that are sub-divided into three types: Aggregating functions, Population-based approaches and Pareto-based approaches. Aggregating functions combines all the objectives into one by any arithmetical operation. Population-based approaches use population to diversify the search, in which each generation sub-populations are generated by proportional selection. When using the Pareto approach, multiple objects are simultaneously optimized to find the non-dominated points of Pareto Front (PF).

The objectives are categorized into single ones by pre-multiplying each objective with a user-supplied weight [4]. The second approach is a Multi-objective algorithm, which is designed to solve black box objective optimization problems. These algorithms contain parts of optimization methods and Pareto search methods including: Strength Pareto Evolutionary Algorithm (SPEA) [5]; Multi-Objectives Particle Swarm Optimization (MOPSO) [6]; A Multi Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [7]; Non-dominated Sorting Genetic Algorithm–II (NSGA-II) [8]; and Non-dominated Sorting Genetic Algorithm–III (NSGA-III) [9].

As evidenced by previous literature [10], multi-objective algorithms are commonly used to optimize WSNs parameters. Limited research focus on multi-objective algorithms such as NSGA-III (with the exception of one study that uses NSGA-III at the deployment stage) [11]. Another weak point in this field is that only a few papers previously consider QoS and its metrics, such as coverage, reliability, delay and packet delivery. Furthermore, only a small number of studies employ performance indicators to evaluate PF solutions for multi-objectives algorithms.

Hence, the contribution of this research is to present a new algorithm named Enhanced Multi-objective Non-dominated Sorting Genetic Routing Algorithm (ENSGRA) based on NSGA-III with some changes in finding reference points, and crossover operation, this algorithm avoids the weaknesses in previous algorithms and improves the QoS in WSNs. This improvement can be achieved by optimizing three objectives in deployment and routing stage, namely, the number of active sensor nodes, energy consumption and network coverage. ENSGRA will work as a dynamic protocol in WSNs environment, Finally, considering computation time when comparing the proposed algorithm with others is highly important to avoid time complexity, so it taken in consideration when the proposed algorithm compared with other algorithms.

This paper is arranged as follows. In this Section 1, the introduction is presented. Section 2 illustrates related concepts of using multi-objective algorithms to improve performance of QoS in WSNs, and related work in this field In addition, this section describes the research problems. Section 3 displays the proposed algorithm and its framework. Section 4 discusses the results in comparison with other algorithms alongside the evaluation. Finally, Section 5 presents the conclusion and future works.

II. CONCEPTS, RELATED WORKS, AND PROBLEM DESCRIPTION

The goal of this research is to solve the MOP to achieve high performance in QoS for WSNs. Therefore, this section introduces the most relevant and pertinent concepts to the main ideas of the study.

A. QUALITY OF SERVICE (QoS)

Quality of Service (QoS) has no common or formal definition, but can be considered as the capability to provide assurance that the service requirements of applications are satisfied. However, this assurance depends on the type of application targeted. QoS in WSNs can be recommended for achieving reliability, timeliness, robustness, availability and security. Several QoS parameters can be used to measure the degree of satisfaction of these services, including throughput, delay, jitter and packet loss rate [12]. Other parameters for various applications, including optimizing energy consumption, coverage and connectivity to measure QoS [13], are factors being considered in this research.

B. PARETO FRONT (PF)

Pareto Front is generated by a specific set of solutions, where no multiple objectives can be improved without sacrificing the others [2]. The PF approach has the following goals [14]:

1. Convergence: To find a set of Pareto optimal solutions, these solutions more relevant to each other.

2. Diversity: To find a set of diverse solutions to prevent premature convergence and achieve a well-distributed trade-off PF; Note that diversity is symmetric in a two-dimensional space (two objectives) whilst more difficult to obtain in three-dimensional space (three objectives) [15].

In satisfying multiple objectives, using a 3D space is more realistic but it increases complexity [16], [17]. However Non-dominated sorting, based on optimal PF as shown in Figure 1, represents a number of optimal solutions, with each front set between two objectives. Including several Pareto optimal solutions in the evaluation generation is beneficial [18].

This figure illustrate that dominated solutions which are in black dots are solutions that have other solution dominate on it as the solutions in blue dots.

C. RELATED WORKS

In table 1 represent the recent ones related work that concern in using multi-objective algorithms to optimize objectives in WSNs.

D. PROBLEM DESCRIPTION AND OBJECTIVE FUNCTIONS

In MOP, several objective functions simultaneously need optimisation (minimised or maximised). For example, m objective functions require

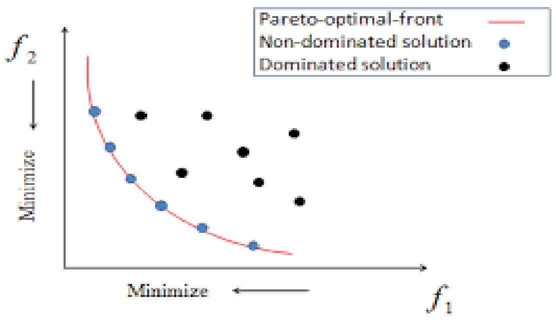


FIGURE 1. Non-dominated sorting of a population Pareto front [19].

TABLE 1. Literature review for related works.

Ref.	Problem	Objectives	Algorithm
(2020) [1]	Solve NP Hard problem based on optimizing four objectives	Energy conservation, network lifetime, coverage, and load balancing	NSGA-II
(2019) [11]	Use and adopt one of the approaches for WSNs deployment (WSND)	Maximize coverage and minimize the energy consumption	NSGA-III
(2019) [20]	Finding the best locations and configuration of sensors in 2D environment in order to prolong the life time of the network	Best coverage, minimal energy consumption, and minimal number of active sensor	Compare LMOJPSO With NSGA-II
(2018) [14]	Solving clustering and routing problem in WSNs	Minimizing energy consumption, Maximizing the number of clustered nodes, maximizing the network throughput	Compare NSGA-II With SMPSO
(2012) [18]	Solve the energy conservation and coverage preservation design problems using cluster-based WSNs	Energy conservation and coverage preservation	Compare MOEA/D With NSGA-II

Minimise or maximise:

$$F(x) = (F_1(x), \ldots, F_m(x)), \tag{1}$$

This research aims to solve MOP that contain three objective functions:

1. To minimise the number of active sensors
2. To minimise the intersection between sensor nodes to reduce energy consumption
3. To maximise separation between sensor nodes to increase network coverage.

1) NUMBER OF ACTIVE SENSOR NODES

A function that finds the sum of sensor status is necessary to determine the minimum number of active sensors. This function [21] uses the following equation,

$$f_1(x) = \sum_{i \in N} Status_i, \tag{2}$$

where N is the number of sensors and $f_1(x)$ provides the minimum number of active sensors in WSN that is deployed randomly in the area of interest.

2) ENERGY CONSUMPTION OBJECTIVE FUNCTION

The function that minimises the intersection between sensors in its sensing area is used to conserve energy (As when density of sensors increase in some areas more than others. It will consume energy so reducing intersection between sensors will decrease energy consumption). This function [20] uses the following equation:

$$f_2(x) = \sum_{i=1}^{N} \sum_{j=1}^{N} Rs_i \cap Rs_j, \tag{3}$$

where N is the number of sensors, Rs is sensing radius and $f_2(x)$ must be as minimised as possible. Without neglected transferring and receiving the packets to support communication between sensor nodes.

3) NETWORK COVERAGE OBJECTIVE FUNCTION

The function that maximises the separation between nodes is used to maximise network coverage. This function [1] uses the following equations,

$$f_3(x) = \max(sep), \tag{4}$$

$$sep = \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij}, \tag{5}$$

where dij is the distance between any two nodes and should be as maximised as possible.

III. ENHANCED NON-DOMINATED SORTING GENETIC ROUTING ALGORITHM (ENSGRA)

With the aim to overcome performance problems in QoS for WSNs, this research proposes an enhanced multi-objective algorithm called ENSGRA, which is based on clustering and scheduling. This algorithm can help avoid previous multi-objective algorithm problems such as premature convergence and negative effects of redundant solutions. In addition, ENSGRA also achieves more convergent and less divergent global solutions than other algorithms. The improvement is achieved in the NSGA-III algorithm using adjusted weighted clustered scheduled reference points, and multi-parent crossover (MPX) operation.

A. ENSGRA PROPOSED FRAMEWORK

This algorithm is proposed to enhance the QoS performance in WSNs. Figure 2 shows the framework block diagram of the WSN architecture with the proposed ENSGRA. Nodes are randomly deployed in the area of interest, then routing is achieved by clustering and scheduling operations. Initially, the algorithm deploys nodes and randomly selects a cluster head (the algorithm initialises the population). After using NSGA-III and updating genetic operations recombination (crossover), two parents crossover are integrated with multi-parent crossover (MPX). Then reference points are updated by the adjusted weighted clustered scheduled reference points to enhance non-dominated PF solutions.

The fitness of each node is calculated based on the proposed multi-objective fitness function to achieve enhanced objectives, which include the number of active sensors,

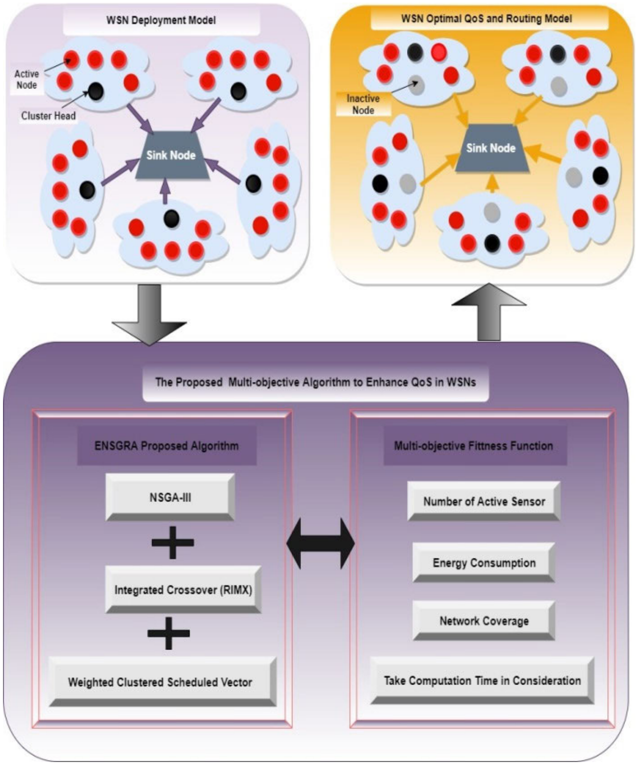


FIGURE 2. ENSGRA algorithm, framework block diagram of WSNs.

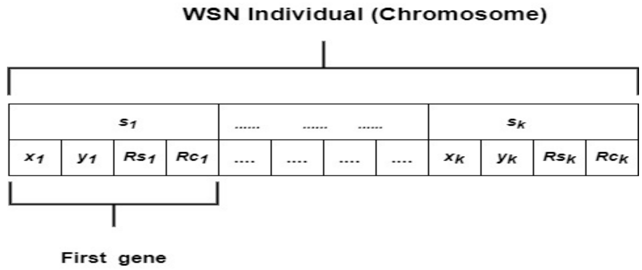


FIGURE 3. Format of original individual.

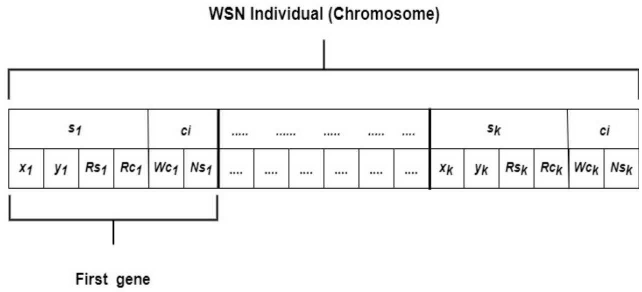


FIGURE 4. Format of updated individual.

energy consumption and network coverage. Moreover, this method maintains path reliability, which is based on clustering, and node connectivity, which is grounded on radius, as constraints. Finally, the final WSN routing solutions (non-dominated PF solutions) are given to the decision maker. In these solutions, the new cluster heads are selected based on node energy, node sensing radius and node communication radius. In addition, in the final solutions, several deployed nodes turn off (become inactive) based on sensor scheduling, which increase energy efficiency, network coverage and network lifetime.

B. ORIGINAL INDIVIDUAL FORMATTED STRUCTURE WITHOUT CLUSTERING

All individuals (chromosomes) are represented by $m \times 2$ matrix, where m is the number of nodes. The number in the first column of row i states to which cluster the node i belongs. The elements of the second column of the chromosome are selected randomly to determine if the node acts as a cluster head or as a natural sensor node [22]. Figure 3 shows the format of the original individual, which contains (k) as the amount of active sensor nodes and (s) as the number of genes. These two must be equal given that each sensor is considered as a gene that contains four parameters: x coordination, y coordination, sensing radius and communication radius for each sensor. In this case, sensors have two types of status, active or non-active node, as shown in Equation (6).

$$Node\ status = \begin{cases} 0, & \text{if node non active} \\ 1, & \text{if node active} \end{cases} \tag{6}$$

As an example let a total of 64 active sensors from 100 sensors that are deployed in the area of interest are obtained after using NSGA-III, whilst the other sensors will go to sleep mode (inactive). Let the first sensor has $id = 1$, and the four parameters are presented as follows: $x_1 = 301$, $y_1 = 59$, $Rs_1 = 110$ and $Rc_1 = 156$.

C. UPDATED INDIVIDUAL FORMATTED STRUCTURE WITH CLUSTERING

In the binary method [1], an individual (chromosome) is represented as a string of 0s and 1s, where 0 indicates that the node is a non-cluster head/member and a 1 indicates that the node is a cluster head.

Figure 4 shows the format of an updated individual, which contains (k) as the amount of sensor nodes and (s) as the number of genes. The two must be equal given that each sensor is considered as a gene that contains four parameters: x coordination, y coordination, sensing radius and communication radius for each sensor (as in Section 3.2). In addition, this format considers clustering by adding a number of cluster heads, with (m) PF and each PF solution has a number of cluster heads c_i, then $i = \{1, \ldots, m\}$ and ci is $2 \leq c_i \leq \frac{k}{10}$. For example, for (100) sensors, the minimal number of cluster heads is (2) and the maximum number of cluster heads is (10). In clustering, two additional parameters (Wc) are the sensor node in any cluster and node status (Ns) that is a natural active or non-active node or cluster head.

D. AN INTEGRATED CROSSOVER OPRATION IN ENSGRA

A new crossover operator, called Random Multi-point Crossover Operator (RMX) [23], is proposed to solve the variable ordering problem. RMX is used for probabilistic

cp_1 cp_2 cp_3 cp_4

p_1	5	6	10	99	77	100	15	87	66	55	149	77	75	49
p_2	123	156	44	8	17	32	59	88	123	2	71	93	102	86
p_3	55	88	43	87	109	58	14	13	75	46	43	73	44	12
p_4	13	28	31	85	105	108	43	27	55	46	42	84	92	73
ch_1	5	6	44	8	17	58	14	13	55	46	42	84	75	49
ch_2	123	156	10	99	77	108	43	27	75	46	43	73	102	86
ch_3	55	88	31	85	105	100	15	87	123	2	71	93	44	12
ch_4	13	28	43	87	109	32	59	88	66	55	149	77	92	73

FIGURE 5. Recombination after using RIMX operator.

graphical models that have directed arcs. This operator can avoid premature convergence to determine good solutions in a reasonable number of generations.

In this research, the proposed algorithm uses the integration between two parents crossover with multi-parent crossover (MPX). Moreover, a Random Integrated Multi-point Crossover (RIMX) is proposed. Figure 5 shows RIMX in algorithm 1, how this operator works based on the RMX algorithm and represents the integration operation between two parents crossover with multi-parent crossover (MPX).

Let m represent the number of parents that are chosen randomly, which may be two or four parents. When RIMX receives two chromosomes (parents), p_1 and p_2 are recombined to create two new chromosomes (child), ch_1 and ch_2, based on p_1 and p_2 recombination. However, if RIMX receives four chromosomes, p_1, p_2, p_3 and p_4 are recombined and returns four new chromosomes, ch_1, ch_2, ch_3 and ch_4 based on p_1, p_2, p_3 and p_4 recombination. Random sets of numbers n of cut points (cp) are to be used. Use $n = 4$, as an example. Next, n positions of cut (cp_1, cp_2, …, cp_n) are randomly chosen. Then, the selected chromosomes are recombined $p1$, p_2, p_3 and p_4, according to positions selected to generate ch_1, ch_2, ch_3 and ch_4. The recombination of genes occurs as follows: specific gene sequences are exchanged between four parents. The first sequence (from cp_1 to cp_2) and the second (from cp_2 to cp_3), then the third (from cp_3 to cp_4) and so on.

Figure 5 presents four individuals to be recombined: $p1$, p_2, p_3 and p_4. Consider that four cut points are randomly set: cp_1, cp_2, cp_3 and cp_4, genes before cp_1 without any changes. The first sequence cp_1 to cp_2 is exchanged between p_1 and p_2 and between p_3 and p_4. The second sequence cp_2 to cp_3 is exchanged between p_1 and p_3 and between p_2 and p_4. The third sequence cp_3 to cp_4 is exchanged between p_1 and p_4 and between p_2 and p_3. The sequence after cp_4 does not change. These operations are repeated with other genes to the end.

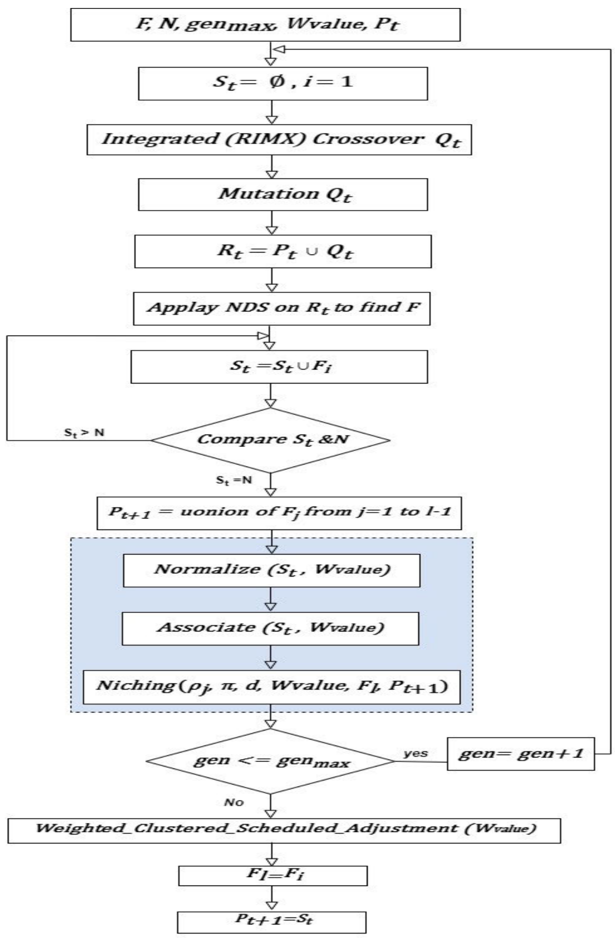


FIGURE 6. ENSGRA flowchart.

E. UPDATED REFERENCE POINTS IN ENSGRA

Based on weighted clustered scheduled vector adjustment, ENSGRA is used to improve the convergence speed and distribution of NSGA-III algorithm. ENSGRA increases the

Algorithm 1 RIMX Procedure Based on RMX

Input: Parent p_1, p_2, p_3, p_4
Output: Child ch_1, ch_2, ch_3, ch_4
n = random (1, number of genes/2) /*n is number of cut points defined randomly
Positions = randomly Chosen Positions (n) /* positions of cuts are randomly selected
m = random [2,4] /*m is number of parents it may be 2 or 4
if $m == 2$
Select p_1, p_2 randomly
ch_1 and ch_2 = recombination (positions, p_1, p_2) /* two parent crossover
else /*$m == 4$
Select p_1, p_2, p_3, and p_4 randomly
ch_1, ch_2, ch_3, and ch_4 = recombination (positions, p_1, p_2, p_3, p_4) /* multi parent crossover

TABLE 2. WSNs settings for experiments.

Parameter	Value
Number of Sensors	100, 200, 300, 400 and 500
Region of Interest	1000x1000 m^2, 1200x1200 m^2, 1400x1400 m^2, 1600x1600 m^2, and 1800x1800 m^2 respectively.
Sensing Radius (*Rs*)	[1-300] cm
Communication Radius (*Rc*)	[1-300] cm
Number of Objectives	3

TABLE 3. Algorithms settings for experiments.

Common Parameters	
Parameter	Value
Number of Scenarios (configurations)	10
Number of populations (solutions)	10, 20, 30, 40, 50, 60, 70, 80, 90, and 100
Number of Iteration	25, 50, 75, and 100
NSGA-II, NSGA-III, and ENSGRA Parameters	
nDivision (for NSGA-III, and ENSGRA)	10
PCrossover	0.5
PMutation	0.5
MutationRate	0.1
LMOJPSO Parameters	
C1	0.05
C2	Between 0.72 to 0.94
TimeLag	0, or 1

individual ability to evolve through new differential evolution strategies, while dynamically adjusting the weight vector [24]. The original structured reference point Z^s with W_{value} in generation procedure is replaced, followed by call weight adjustment function as the following pseudo code in algorithm 2.

The distribution of weighted vectors is important when all individuals are indistinguishable from one other. The weight vectors are adjusted by comparing the density of the entire objective space and subspace [24], based on this principle

Algorithm 2 ENSGRA Based on Weighted Clustered Scheduled Adjustment Procedure

Input: N structured reference points W_{vlue}, P_t
Output: Offspring population P_{t+1}
Initialization (P_t, W_{value})
$gen = 1$
Select Number of clusters randomly based on number of population (N_c)
Select cluster head for each cluster randomly (CH)
While $gen \leq gen\_max$ **do**
Q_t = Crossover(*using RIMX algorithm*) +Mutation(P_t)
$R_t = P_t \cup Q_t$
$(F_1, F_2, \ldots)$ = Non-dominated-sort (R_t)
Repeat
$S_t = S_t \cup F_i$ and $i = i + 1$
until $|S_t| \geq N$
Last front to be included: $F_l = F_i$
if $|S_t|$ = N **then**
$P_{t+1} = S_t$, break,
Else
$P_{t+1} = \bigcup_{j=1}^{l-1} F_j$ and $K = N - |P_{t+1}|$
Normalize-objectives $P_t = Normalize(S_t, W_{value}, N_c)$
Associate each member **s** of S_t with a reference point: $[\pi\ (s)\ , d\ (s)] = Associate(S_t, W_{value}, N_c)$
Compute niche count of reference point $j \in W_{value}$, $\rho_j = \sum_{s \in S_t/F_l} (\pi\ (s) = j?1 : 0)$:
Choose K members one at a time from F_l to construct: $Niching(K, \rho_j, \pi, d, W_{value}, P_{t+1})$
end if
W_{value} = Weighted_Clusterd_Scheduled_Adjustmen (W_{value}, N_c)
$gen{+}{+}$
Select individual scheduling randomly (select active/inactive, individual)
Select cluster heads based on fitness functions
end While
return P_{t+1}

ENSGRA divided into two situations as the following equations.

When the cluster density is less than the population density, whether the cluster density is too low must be determined. Then let t_{min} is the minimum threshold to adjust cluster density. If the cluster density is less than the minimum population density $t_{min} d_0$, then the cluster density is too low. In this case, the weighted clustered scheduled vector should be deleted, otherwise the two closest neighbour weighted clustered scheduled vectors are adjusted using equations (7) and (8),

$$W_{value}(m) = W_{value}(m) + ((t_{min} \times d_i) - dist_{min}) \times W_{value}\left(neig_m\right) \quad (7)$$

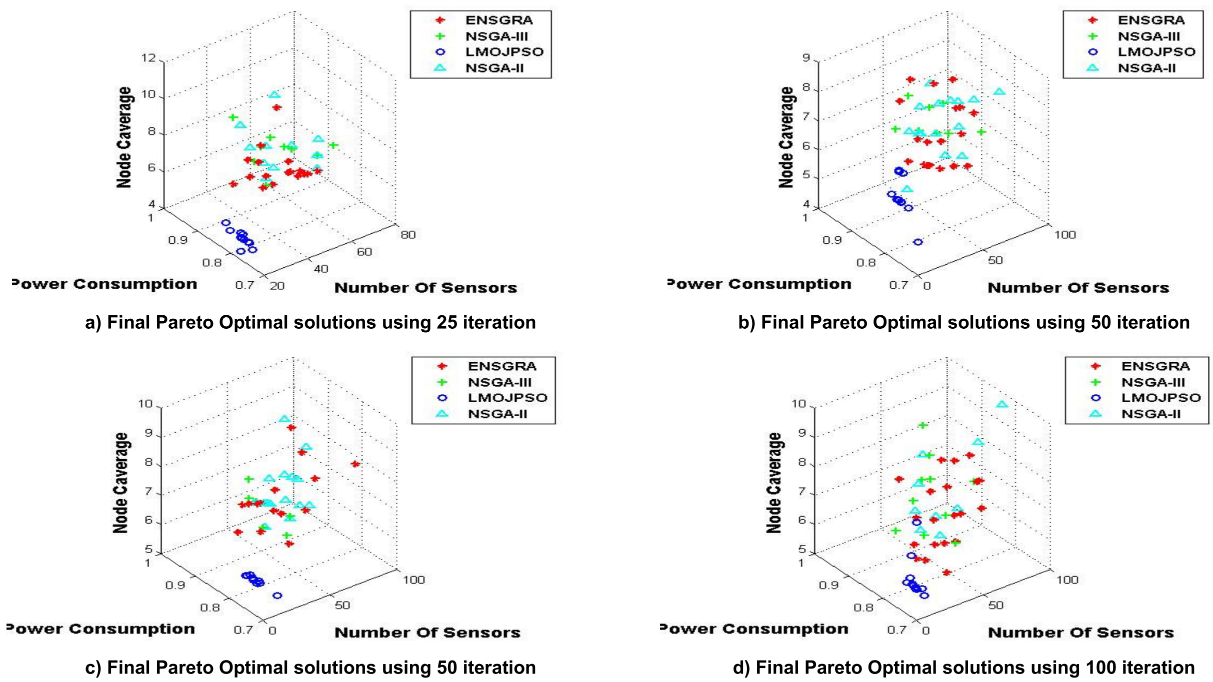
$$W_{value}(n) = W_{value}(n) + ((t_{min} \times d_i) - dist_{min}) \times W_{vlue}\left(neig_n\right) \quad (8)$$

where:

t_{min} = 1.5 minimum threshold to give the best result (proved by experiment).

d_0 : density of population.

d_i : cluster density



a) Final Pareto Optimal solutions using 25 iteration

b) Final Pareto Optimal solutions using 50 iteration

c) Final Pareto Optimal solutions using 50 iteration

d) Final Pareto Optimal solutions using 100 iteration

FIGURE 7. Pareto Optimal solutions for the algorithms for 100 sensor and 50 Population nodes using 25, 50, 75, 100 iteration.

Algorithm 3 *Weighted_Clustered_Scheduled_procedure*

Input: Population weight $W_{value}(w_1, w_2, \ldots, w_m)$
Output: Offspring weight $W_{value}(w_1, w_2, \ldots, w_m)$
$Normalize(P_t, W_{value}, N_c)$
Calculate all population density $d_0(pop)$
Calculate each cluster density $\{d_1(clus), d_2(clus), \ldots, d_k(clus)\}$
If $d_i(clus) < d_0(pop)$
  If $d_i(clus) < t_{min} \times d_0(pop)$
    Delete a weighted clustered scheduled vector
  else
    Adjustment using weighted clustered scheduled vectors by equations (7) and (8)
  end If
else if $d_i(clus) < t_{mix} \times d_0(pop)$
  Adjustment using weighted clustered scheduled vectors by equations (9) and (10)
  else
    Add a weighted clustered scheduled vector
  end if
end If
If $i = m \,\&\&\, length(W) \neq Number\ of\ population$
Adjustment using weighted clustered scheduled vectors for all population
end If

$dist_{min}$: minimum distance between two closest neighbour weighted clustered scheduled vectors.

$W_{value}(neig_m)$ and $W_{value}(neig_n)$: neighbour weights of the respective weights.

2. When the cluster density is greater than the population density, whether the cluster density is too large must be determined. Then let t_{max} is the maximum threshold to adjust density. If the cluster density is greater than the maximum population density $t_{max}d_0$, then the cluster density is too high. In this case, the weighted clustered scheduled vector should be considered, otherwise the two furthest neighbour weighted clustered scheduled vectors must be adjusted using the following equations (9) and (10),

$$W_{value}(m) = W_{value}(m) + \frac{(dist_{max} - (t_{max} \times d_i))}{2} \times W_{value}(n) \tag{9}$$

$$W_{value}(n) = W_{value}(n) + \frac{(dist_{max} - (t_{max} \times d_i))}{2} \times W_{value}(m) \tag{10}$$

where:

$t_{max} = 2.5$ maximum threshold to give the best result (proved by experiment).

$W_{value}(m)$ and $W_{value}(n)$: furthest weight vectors.

$dist_{max}$: maximum distance between two furthest neighbour weighted vectors.

Algorithm 3 illustrates the pseudo code for weighted clustered scheduled adjustment procedure.

Figure 6 shows the flowchart of the proposed algorithm ENSGRA as MPX is added to two-parent crossover and integrated using RIMX. Then, the adjusted weighted clustered scheduled vector is used to find W_{value} and increase the number of reference points to find new associated PF solutions.

TABLE 4. Multi-objective algorithms optimization results and dynamic weights for 100 nodes.

Energy Consumption Results (EC)					*Energy Dynamic Weights (EDW)*				*(EW X EDW)*			
Iter.	LMOJPSO	NSGA-II	NSGA-III	ENSGRA	LMOJPSO	NSGA-II	NSGA-III	ENSGRA	LMOJPSO	NSGA-II	NSGA-III	ENSGRA
25	0.8160	0.8491	0.8468	0.8335	0.4	0.1	0.2	0.3	**0.12**	**0.03**	**0.06**	**0.09**
50	0.8209	0.8423	0.8439	0.8299	0.4	0.2	0.1	0.3	**0.12**	**0.06**	**0.03**	**0.09**
75	0.8195	0.8420	0.8404	0.8330	0.4	0.1	0.2	0.3	**0.12**	**0.03**	**0.06**	**0.09**
100	0.8281	0.8414	0.8378	0.8240	0.4	0.1	0.2	0.3	**0.12**	**0.03**	**0.06**	**0.09**
Number of active Sensors Results (*NS*)					# Sensors Dynamic Weights (*SDW*)				(*SW X SDW*)			
Iter.	LMOJPSO	NSGA-II	NSGA-III	ENSGRA	LMOJPSO	NSGA-II	NSGA-III	ENSGRA	LMOJPSO	NSGA-II	NSGA-III	ENSGRA
25	27	52	54	47	0.4	0.2	0.1	0.3	**0.08**	**0.04**	**0.02**	**0.06**
50	25	52	52	47	0.4	0.1	0.2	0.3	**0.08**	**0.02**	**0.04**	**0.06**
75	24	51	50	46	0.4	0.1	0.2	0.3	**0.08**	**0.02**	**0.04**	**0.06**
100	25	50	49	44	0.4	0.1	0.2	0.3	**0.08**	**0.02**	**0.04**	**0.06**
Network Coverage Results (*NC*)					Coverage Dynamic Weights (*CDW*)				(*CW X CDW*)			
Iter.	LMOJPSO	NSGA-II	NSGA-III	ENSGRA	LMOJPSO	NSGA-II	NSGA-III	ENSGRA	LMOJPSO	NSGA-II	NSGA-III	ENSGRA
25	5.4	7.8	7.8	7.3	0.1	0.4	0.3	0.2	**0.03**	**0.12**	**0.09**	**0.06**
50	5.3	7.8	7.6	7.2	0.1	0.4	0.3	0.2	**0.03**	**0.12**	**0.09**	**0.06**
75	5.2	7.6	7.5	7.1	0.1	0.4	0.3	0.2	**0.03**	**0.12**	**0.09**	**0.06**
100	5.1	7.7	7.6	7.1	0.1	0.4	0.3	0.2	**0.03**	**0.12**	**0.09**	**0.06**
Computation Time Results (*CT*)					Time Dynamic Weights (*TDW*)				(*TW X TDW*)			
Iter.	LMOJPSO	NSGA-II	NSGA-III	ENSGRA	LMOJPSO	NSGA-II	NSGA-III	ENSGRA	LMOJPSO	NSGA-II	NSGA-III	ENSGRA
25	0.30	0.60	0.36	2.64	0.4	0.2	0.3	0.1	**0.08**	**0.04**	**0.06**	**0.02**
50	0.72	1.80	0.78	5.84	0.4	0.2	0.3	0.1	**0.08**	**0.04**	**0.06**	**0.02**
75	0.80	7.00	0.90	7.35	0.4	0.1	0.3	0.2	**0.08**	**0.02**	**0.06**	**0.04**
100	1.59	16.49	1.32	10.54	0.4	0.1	0.3	0.2	**0.08**	**0.02**	**0.06**	**0.04**

IV. RESULTS AND DISCUSSION

A synthetic dataset [25] is used to conduct an experiment of the proposed algorithm. The properties of this dataset are represented as sensors and sink nodes that are placed on the same *2D*-surface of size *Dx* x *Dy*. Sensors regularly and simultaneously capture information packets from its environment with a sensibility radius *Rs* on a basis. Sensors provided by the dataset and sink node coordinates are provided by the optimisation algorithm. Both devices (sink and sensors) can communicate within a communication radius *Rc*. Assuming it will be used MAC Centralized Routing Protocol (MCRP) to avoid packet collision and wormhole attack problem [26]. Other assumption is the sensor nodes are deployed randomly in the area of interest to ensure that, there are well in a dynamic network environment.

Tables 2 and 3 show two types of assumed parameters, WSN and multi-objective algorithms, respectively. This experiment is carried out using Windows 10 Operating System, Intel Core (TM) i5-6200U, 2.4 GHz processor, 8 GB memory (RAM) and MATLAB 2014 as software. The code validated and verificated by programing each part alone using MATLAB code, then testing these parts before do integration between it to know if it give the exact result or not, using different case studies from the synthetic dataset. Table 3 represents parameters that are used to carry out experiments through LMOJPSO, NSGA-II, NSGA-III and ENSGRA. Several parameters are commonly used for all algorithms, special parameters are exclusive for evolutionary algorithms such as genetic algorithm and other parameters are used for particle swarm algorithms.

A. CONFIGURATIONS RESULTS

Each algorithm is experimented upon with a different number of scenarios (configurations) and of iterations. In each scenario, the number of solutions are changed to 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. The same is applied for the different number of iterations, using 25, 50, 75 and 100 iterations. These experiments are repeated with different numbers of sensor nodes, including 100, 200, 300, 400 and 500 sensor nodes, deployed in the region of interest.

Figure 7 shows the results of the density of final PF solutions for each algorithm and its distribution in the *3D* objective space. The *X-axis* shows the number of active sensors, *Y-axis* shows percentage of node intersections (leads to energy consumption) and the *Z-axis* shows node separation (leads to network coverage). This figure is concerned with taking only one scenario (configuration) from 10, when

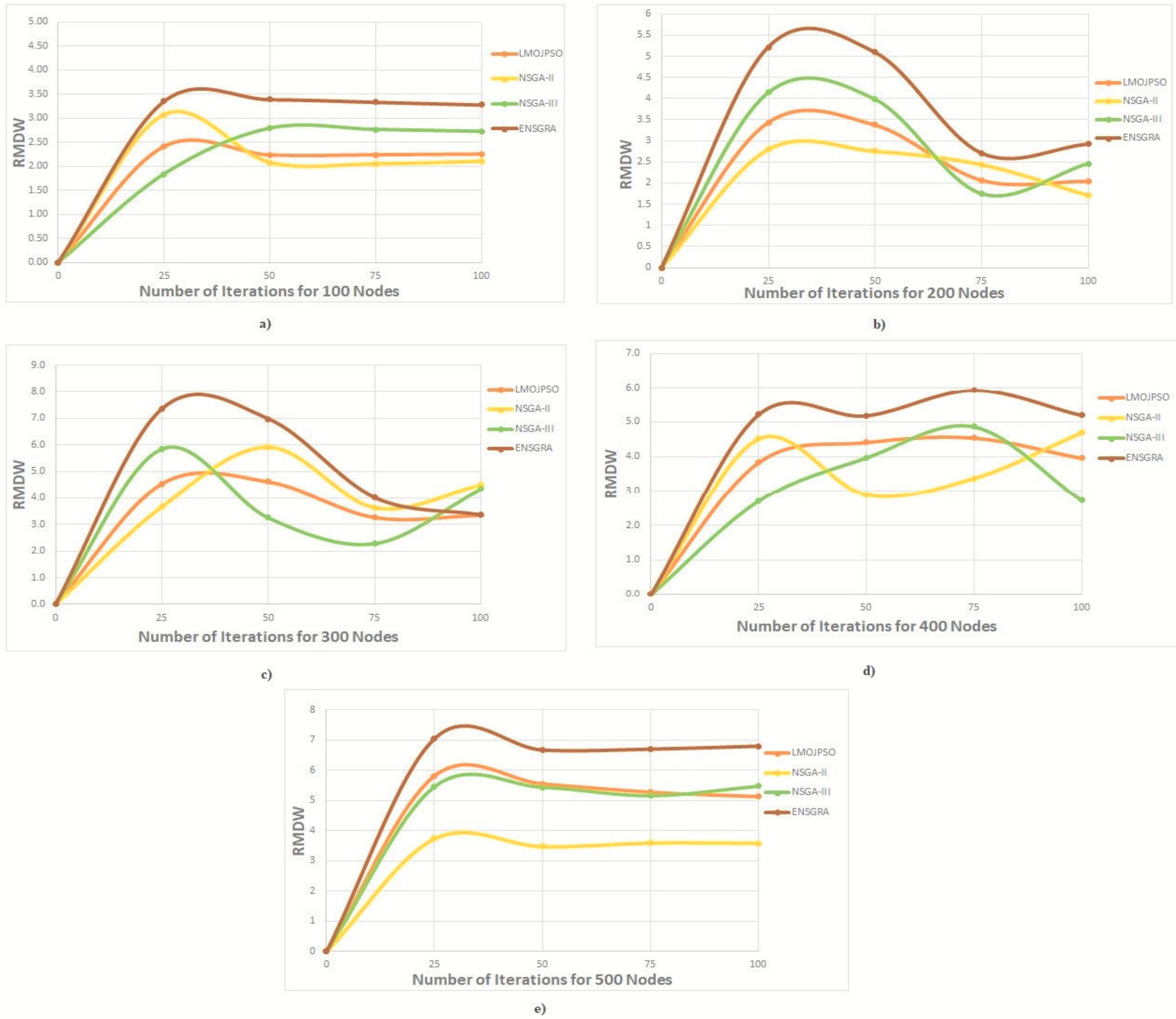


FIGURE 8. **The Resultant of Multi-objectives by Dynamic weights Vs Number of iteration, a) for 100 sensors in 1000 × 1000 m^2, b) for 200 sensors in 1200 × 1200 m^2, c) for 300 sensors in 1400 × 1400 m2, d) for 400 sensors in 1600 × 1600 m^2, and e) for 500 sensors 1800 × 1800 m^2.**

using 100 sensor nodes and 50 population using 25, 50, 75 and 100 iterations.

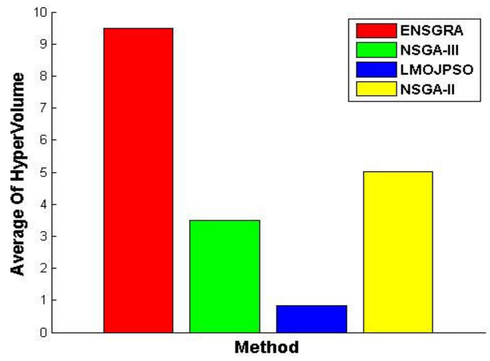
Figures 7-a)–d) represent Pareto optimal solutions using 25, 50, 75 and 100 iterations for the implemented algorithms, respectively. The following colours display the PF for various algorithms.

1. Blue is PF for LMOJPSO algorithm.
2. Cyan is PF for NSGA-II algorithm.
3. Green is PF for NSGA-III algorithm.
4. Red is PF for the proposed algorithm ENSGRA.

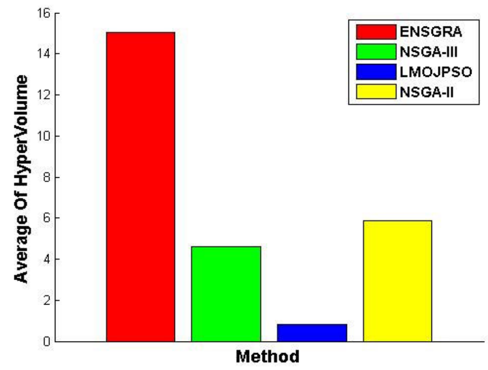
Figure 7 shows that ENSGRA is in between LMOJPSO, NSGA-II and NSGA-III. The PF obtained from ENSGRA enables the decision maker to select the best compromise solution. In turn, the best routing for sensing nodes can be obtained after deployment to achieve QoS for WSN.

The QoS model is used to combine the three conflicting objects, which are number of active sensors, energy consumption, and network coverage. Therefore, at the start, each objective is given weight based on application and determined based on experience about its importance [27], [28]. The weight and coverage threshold are adjusted [29] to obtain the optimum allocation of different business requirements. The results prove that energy consumption and coverage require consideration over (given a greater weight than) any other parameters.

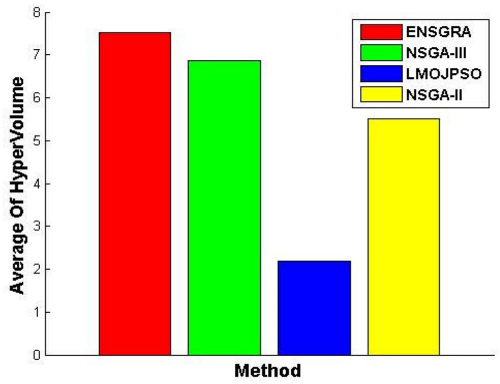
In the present research, the result of the objectives used for each algorithm is calculated using dynamic weights (based on the ranking for each objective, from 1 to 4, with 1 as the lowest and 4 as the highest). Then, this ranking is converted to their respective weights (0.1–0.4), which must be dynamic, based



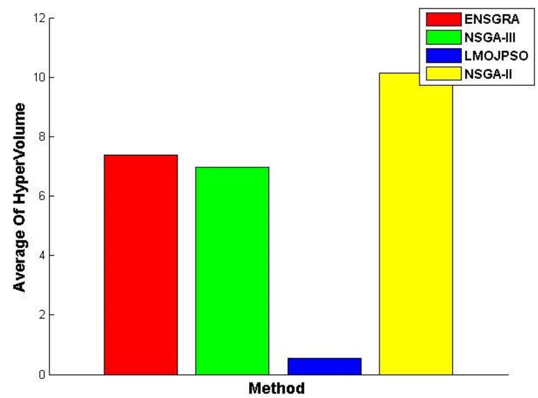
a) Hypervolume for 25 iteration 100 sensors



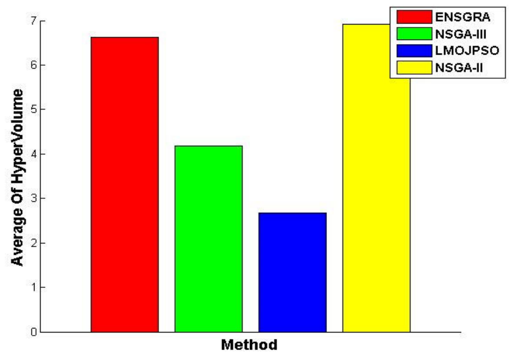
b) Hypervolume for 50 iteration 100 sensors



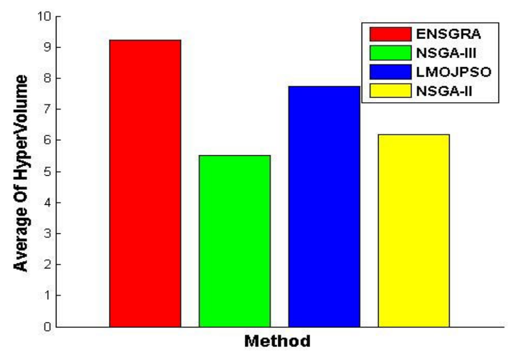
c) Hypervolume for 75 iteration 100 sensors



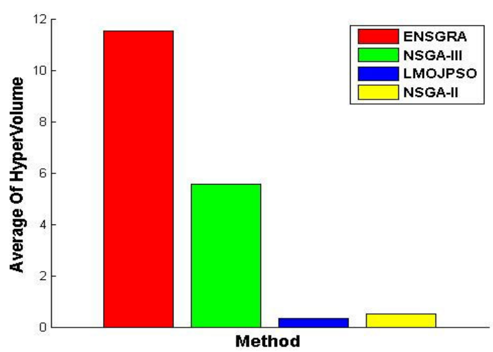
d) Hypervolume for 100 iteration 100 sensors



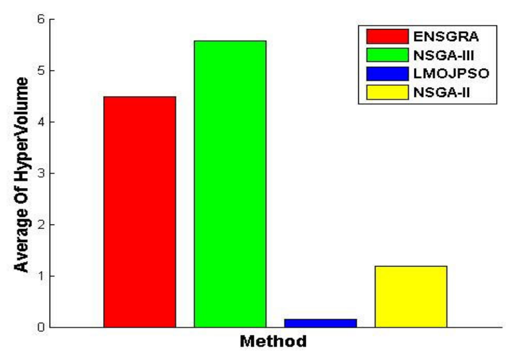
e) Hypervolume for 25 iteration 200 sensors



f) Hypervolume for 50 iteration 200 sensors

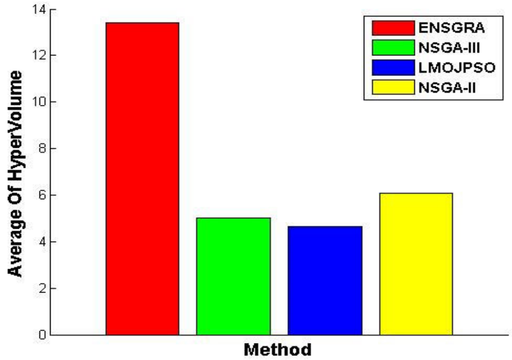


g) Hypervolume for 75 iteration 200 sensors

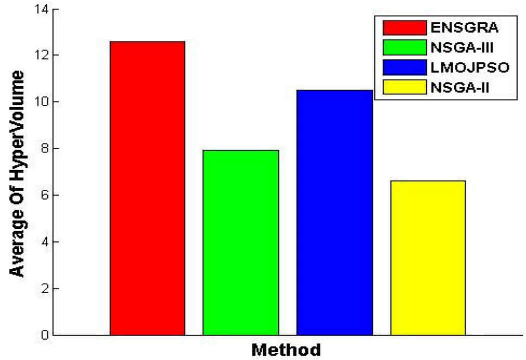


h) Hypervolume for 100 iteration 200 sensors

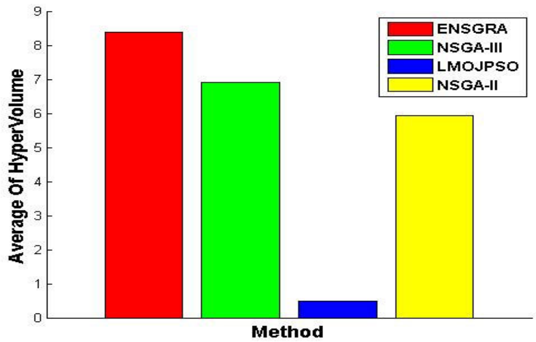
FIGURE 9. **Average Hypervolume for four algorithms (methods): from (a-d) 100 sensors using 25, 50, 75, and 100 iteration. From (e-h) for 200 sensors using 25, 50, 75, and 100 iteration. From (i-l) for 300 sensors using 25, 50, 75, and 100 iteration. From(m-p) for 400 sensors using 25, 50, 75, and 100 iteration. From (q-t) for 500 sensors using 25, 50, 75, and 100 iteration.**



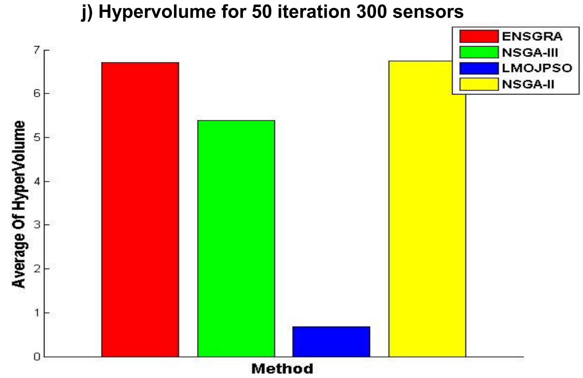
i) Hypervolume for 25 iteration 300 sensors



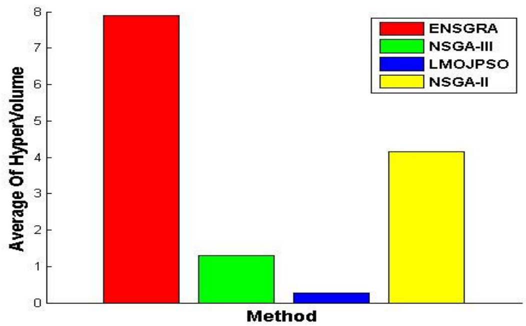
j) Hypervolume for 50 iteration 300 sensors



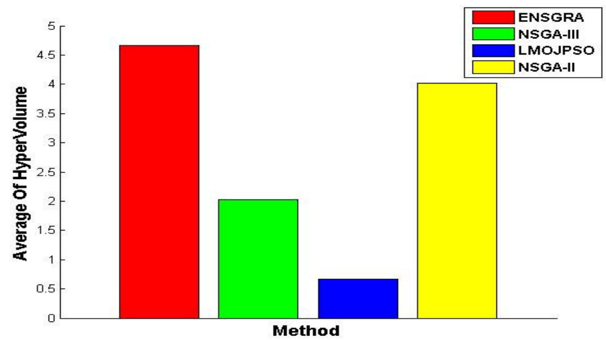
k) Hypervolume for 75 iteration 300 sensors



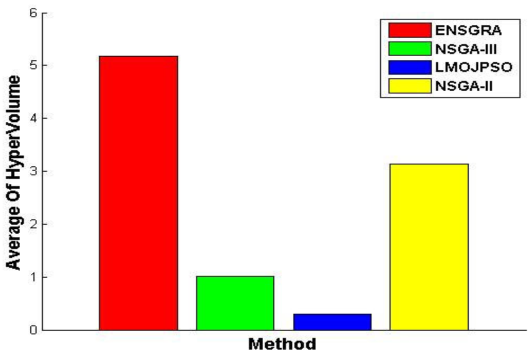
l) Hypervolume for 100 iteration 300 sensors



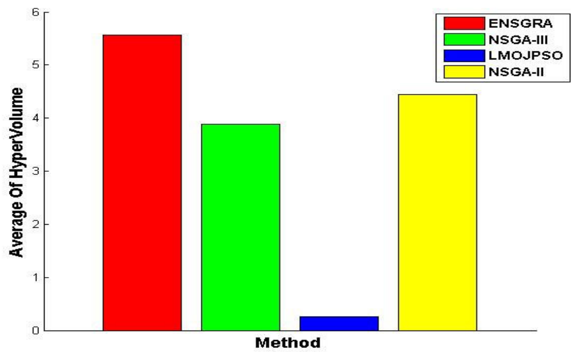
m) Hypervolume for 25 iteration 400 sensors



n) Hypervolume for 50 iteration 400 sensors

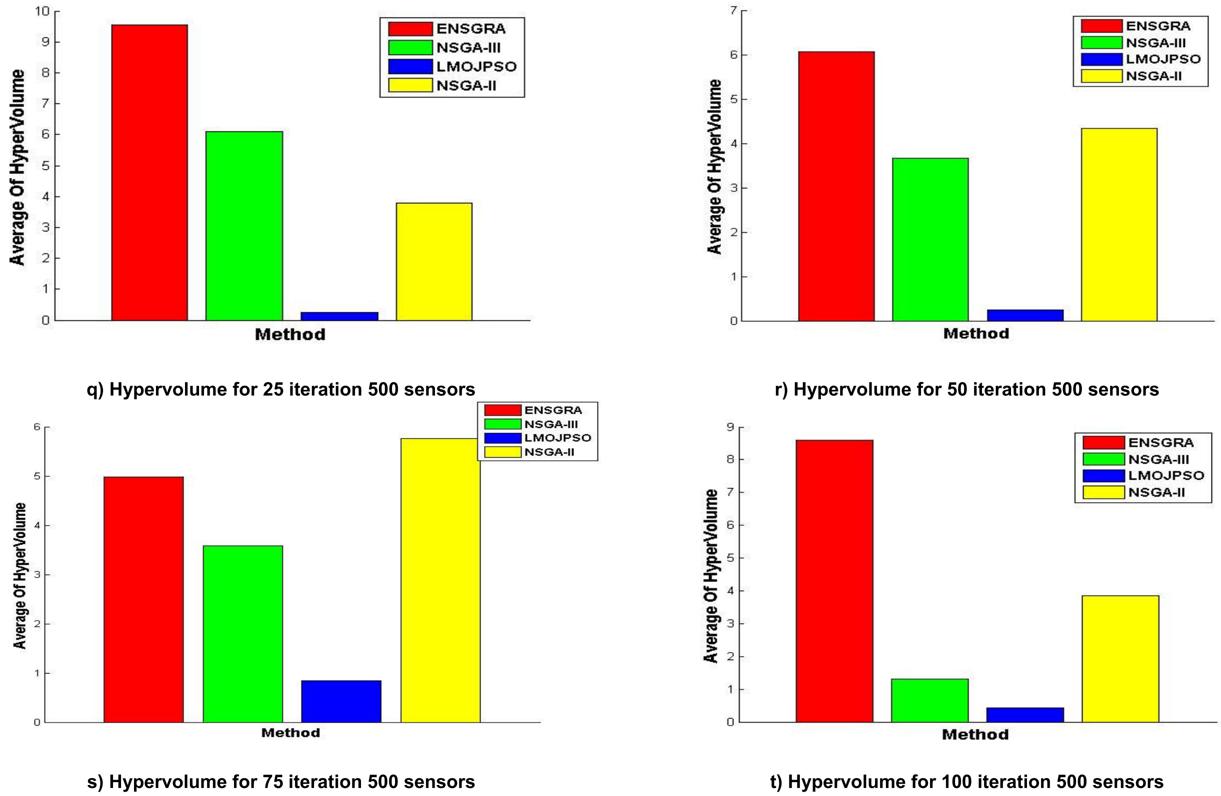


o) Hypervolume for 75 iteration 400 sensors



p) Hypervolume for 100 iteration 400 sensors

FIGURE 9. *(Continued.)* Average Hypervolume for four algorithms (methods): from (a-d) 100 sensors using 25, 50, 75, and 100 iteration. From (e-h) for 200 sensors using 25, 50, 75, and 100 iteration. From (i-l) for 300 sensors using 25, 50, 75, and 100 iteration. From(m-p) for 400 sensors using 25, 50, 75, and 100 iteration. From (q-t) for 500 sensors using 25, 50, 75, and 100 iteration.



q) Hypervolume for 25 iteration 500 sensors

r) Hypervolume for 50 iteration 500 sensors

s) Hypervolume for 75 iteration 500 sensors

t) Hypervolume for 100 iteration 500 sensors

FIGURE 9. ***(Continued.)*** **Average Hypervolume for four algorithms (methods): from (a-d) 100 sensors using 25, 50, 75, and 100 iteration. From (e-h) for 200 sensors using 25, 50, 75, and 100 iteration. From (i-l) for 300 sensors using 25, 50, 75, and 100 iteration. From(m-p) for 400 sensors using 25, 50, 75, and 100 iteration. From (q-t) for 500 sensors using 25, 50, 75, and 100 iteration.**

on an objective ranking value. Table 4 shows that, if energy consumption in LMOJPSO algorithm has the lowest value, then it takes a 0.4 weight value. If ENSGRA has the second rank value, then it takes a 0.3 weight and so on. Computation times are also used as a factor that affects the results for each algorithm. Therefore, computation time is added and given a weight as other parameters to avoid time complexity for the proposed algorithm. Subsequently, the Resultant of Multi-objectives by Dynamic Weights (RMDW) equations are created for each algorithm.

- (EC) indicates Energy Consumption Percentage
- (NS) indicates Number of active Sensors
- (NC) indicates Network Coverage
- (CT) indicates Normalised Computation Time (take time in hour)
- (EW, SW, CW and TW) indicate weights for Energy Consumption, Number of Active Sensor, Network Coverage and Computation Time, respectively
- (EDW, SDW, CDW and TDW) indicate Dynamic Weights or Energy Consumption, Number of active Sensor, Network Coverage and Computation Time, respectively.
- Then let (i) indicate iterations number, which may be (25, 50, 75, 100) each time increasing 25 steps. In addition, let (j) stand for the number of objectives from 1 to 4 given the four algorithms. Therefore, the RMDW equation for any algorithm is:

$$RMDW = \sum_{i=25}^{100} \sum_{j=1}^{4} EC_{ij} \times EW_j \times EDW_{ij} + NS_{ij} \times SW_j \times SDW_{ij} + NC_{ij} \times CW_j \times CDW_{ij} + CT_{ij} \times TW_j \times TDW_{ij} \tag{11}$$

where the objective weights are supposed (*$EW = 0.3$, $CW = 0.3$, $SW = 0.2$, $TW = 0.2$*).

Using an example from Table 4, the resultant RMDW values of LMOJPSO, NSGA-II, NSGA-III and ENSGRA when $i = 25$ for 100 sensor nodes are calculated as follows.

LMOJPSO = (0.8160 X 0.3 X 0.4) + (27 X 0.2 X 0.4) + (5.4 X 0.3X 0.1) + (0.30 X 0.2 X 0.4) = 2.41.

NSGA-II = (0.8491 X 0.3 X 0.1) + (52 X 0.2 X 0.2) + (7.8 X 0.3 X 0.4) + (0.60 X 0.2 X 0.2) = 3.07.

NSGA-III = (0.8468 X 0.3 X 0.2) + (54 X 0.2 X 0.1) + (7.8 X 0.3 X 0.3) + (0.36 X 0.2 X 0.3) = 1.84.

ENSGRA = (0.8335 X 0.3 X0.3) + (54 X 0.2 X0.3) + (7.3 X 0.3 X 0.2) + (0.264 X 0.2 X 0.1) = 3.36.

Figure 8 shows the calculated results for RMDW, where ENSGRA overcomes other algorithms.

Figures 8-(a–(e represent the combination of the average results of 10 configurations, for the three objectives in LMOJPSO, NSGA-II, NSGA-III and ENSGRA. In addition to the objectives, computation time is added to avoid time

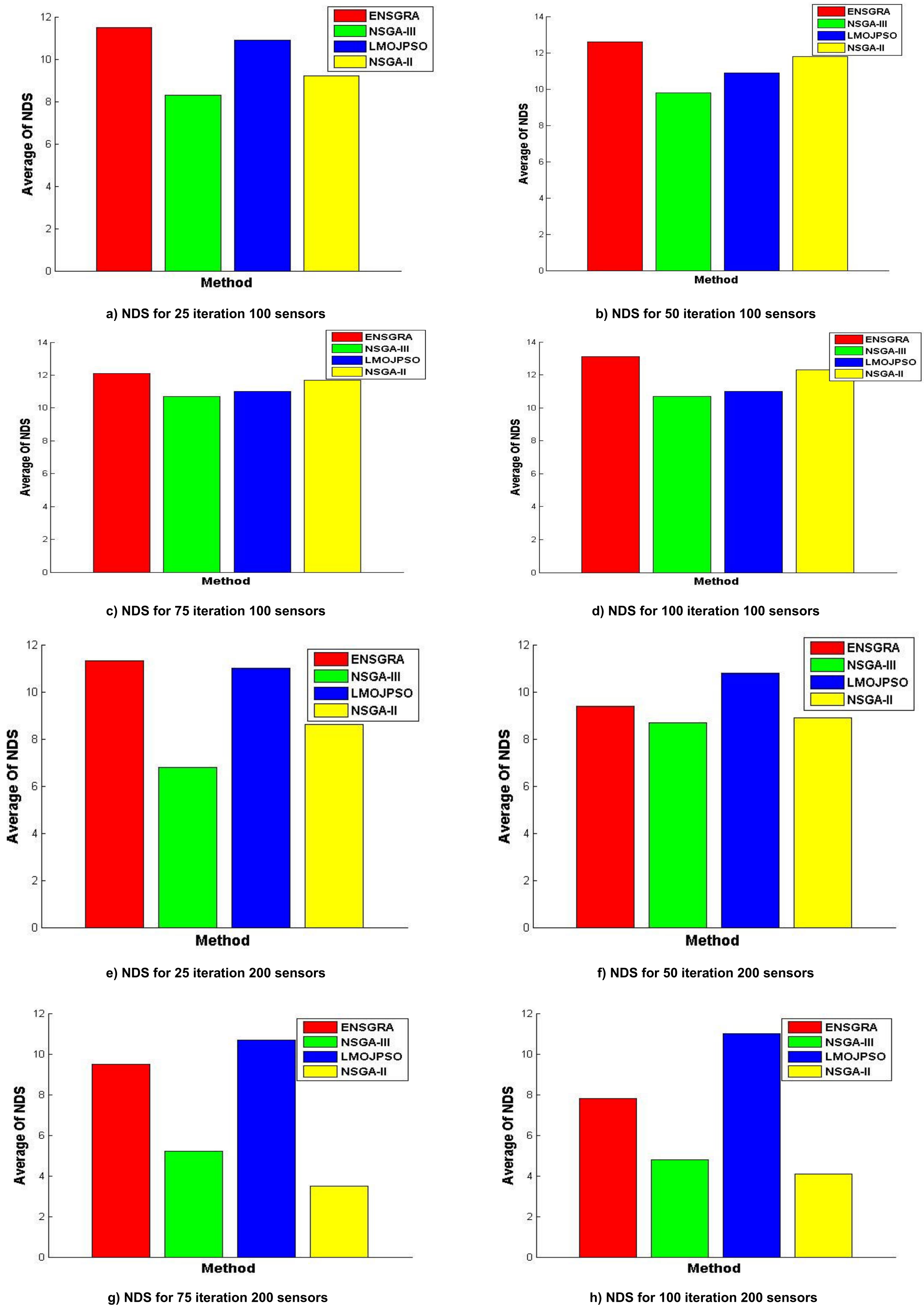
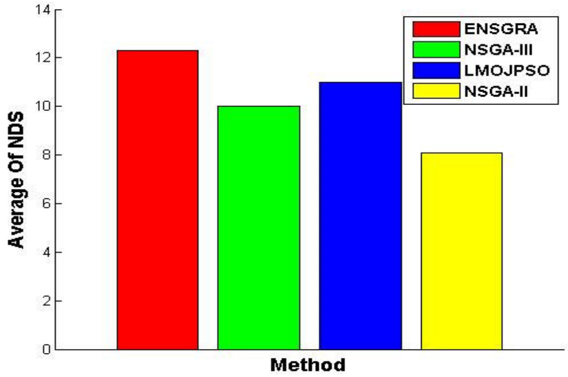
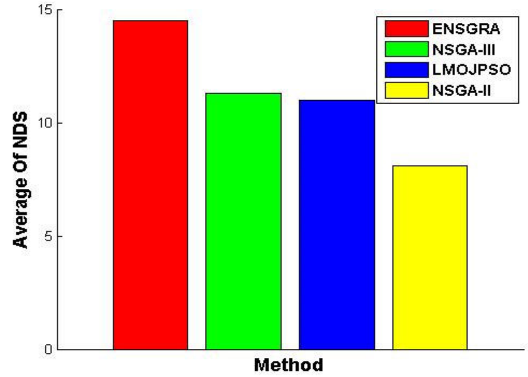


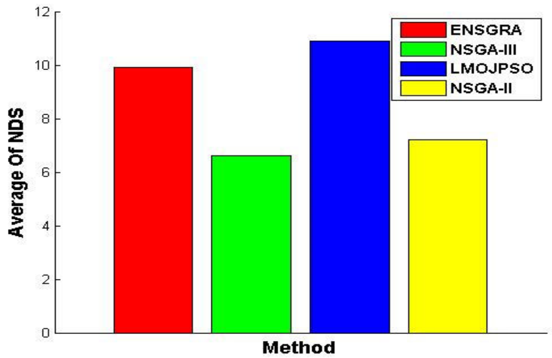
FIGURE 10. Average NDS for four algorithms (methods): from (a-d) 100 sensors using 25, 50, 75, and 100 iteration. From (e-h) for 200 sensors using 25, 50, 75, and 100 iteration. From (i-l) for 300 sensors using 25, 50, 75, and 100 iteration. From(m-p) for 400 sensors using 25, 50, 75, and 100 iteration. From (q-t) for 500 sensors using 25, 50, 75, and 100 iteration.



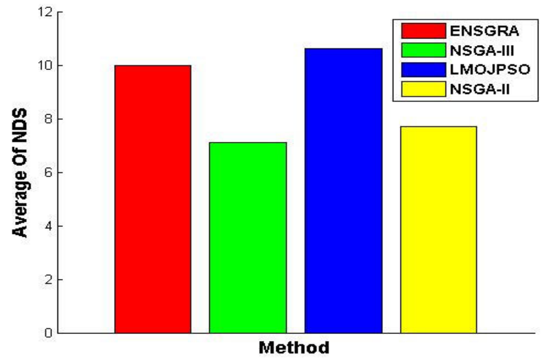
i) NDS for 25 iteration 300 sensors



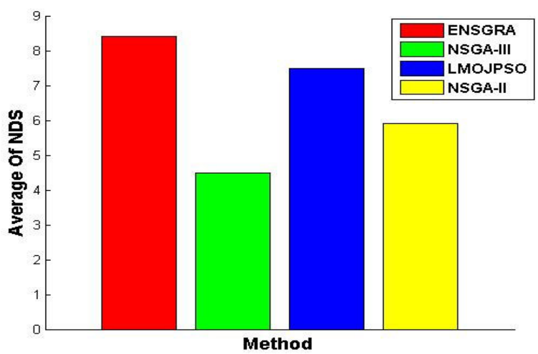
j) NDS for 50 iteration 300 sensors



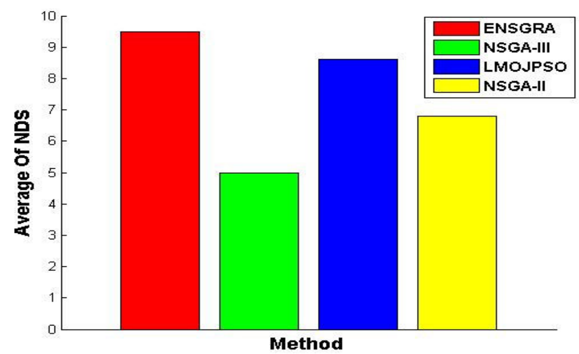
k) NDS for 75 iteration 300 sensors



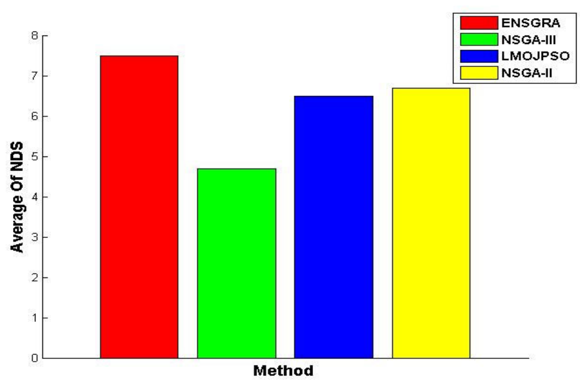
l) NDS for 100 iteration 300 sensors



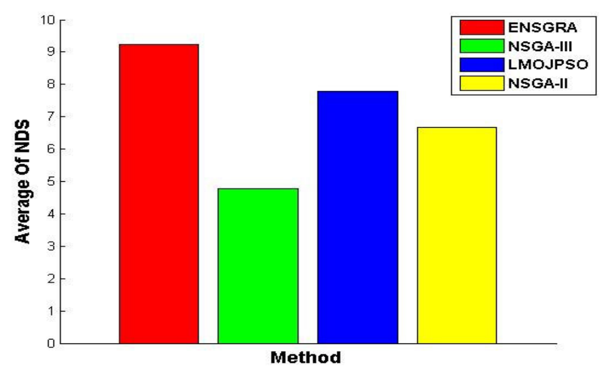
m) NDS for 25 iteration 400 sensors



n) NDS for 50 iteration 400 sensors



o) NDS for 75 iteration 400 sensors



p) NDS for 100 iteration 400 sensors

FIGURE 10. *(Continued.)* **Average NDS for four algorithms (methods): from (a-d) 100 sensors using 25, 50, 75, and 100 iteration. From (e-h) for 200 sensors using 25, 50, 75, and 100 iteration. From (i-l) for 300 sensors using 25, 50, 75, and 100 iteration. From(m-p) for 400 sensors using 25, 50, 75, and 100 iteration. From (q-t) for 500 sensors using 25, 50, 75, and 100 iteration.**

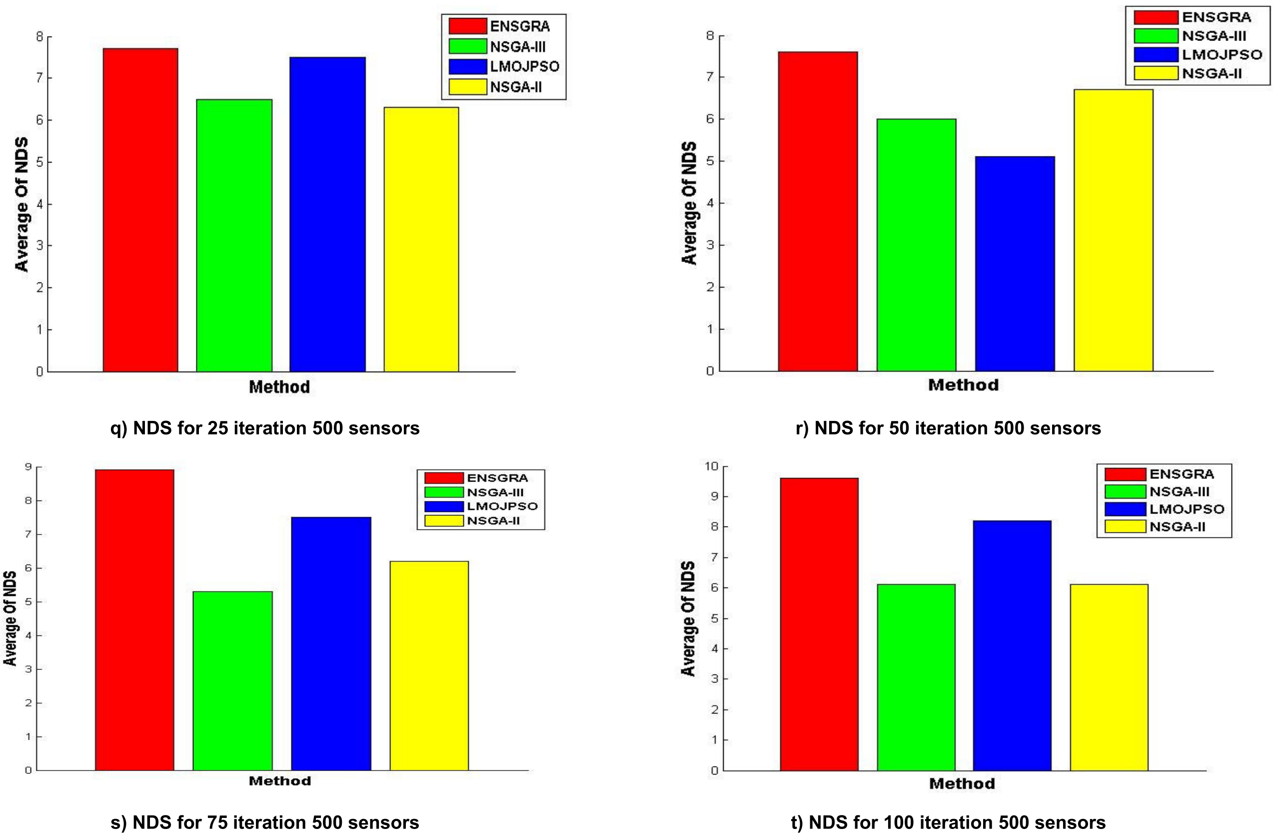


FIGURE 10. ***(Continued.)*** **Average NDS for four algorithms (methods): from (a-d) 100 sensors using 25, 50, 75, and 100 iteration. From (e-h) for 200 sensors using 25, 50, 75, and 100 iteration. From (i-l) for 300 sensors using 25, 50, 75, and 100 iteration. From(m-p) for 400 sensors using 25, 50, 75, and 100 iteration. From (q-t) for 500 sensors using 25, 50, 75, and 100 iteration.**

complexity in the proposed algorithm vs. number of iterations, which are 25, 50, 75 and 100 iterations using 100, 200, 300, 400 and 500 sensing nodes in the RMDW combinations. The calculated results show that the proposed algorithm ENSGRA is superior over the other algorithms and given the highest value, followed by NSGA-II, LMOJPSO and NSGA-III in order. This finding signifies that ENSGRA has the capability to overcome the other algorithms, even though the computation time is added to the combination model (QoS).

B. MULTI-OBJECTVE ALGORITHM EVALUATION

This research adopts the most used evaluation metrics, which are Hyper Volume (HV), Delta (△) and Number of other Non-dominated Solution (NDS) indicators. HV is used to evaluate convergence and diversity (distribution). This indicator is the volume of the objective space dominated by the PF approximation and delimited from above by a reference point. Computation requires a bounded space by the PF and a user defined as a reference point [30], [31].

Whenever the Delta (△) Indicator is used to measure diversity, by obtaining the distribution and spread of solutions, the number of PF solutions is taken and sorted according to the first fitness values. Subsequently, the Euclidean distance between consecutive solutions and the average of the consecutive distances are calculated. Other calculations such as Euclidean distance between the extreme solutions and boundary solutions must be considered to find the diversity metric. This value must be as small as possible because this indicator specifies a uniform distribution [32].

The third indicator is NDS, which is considered a capacity metric that finds the number of optimal solution sets obtained by the optimiser. This indicator, also called Overall Non-Dominated Vector Generation, is easy to use given its low computational complexity [33].

Figure 9 represents the average evaluation of the HV indicator for the used algorithms. The first row shows the HV indicator for 100 nodes using 25, 50, 75 and 100 iterations. The second row shows the HV for 200 nodes using the same number of iterations as in the first row. The third row shows the HV for 300 nodes. The fourth row shows the HV for 400 nodes and the fifth row shows the HV for 500 nodes. Figure 10 represents the average evaluation for the NDS indicator for 100, 200, 300, 400, 500 nodes in each row.

Figure 11 follows the above system to represent the average evaluation of the Delta indicator for 100, 200, 300, 400, 500 nodes, respectively. But we use Delta to evaluate two objectives (energy consumption, and network coverage) as Delta is unsuitable in multi-objective problems with more than two objectives. In these figures illustrate the evaluation

IEEE Access

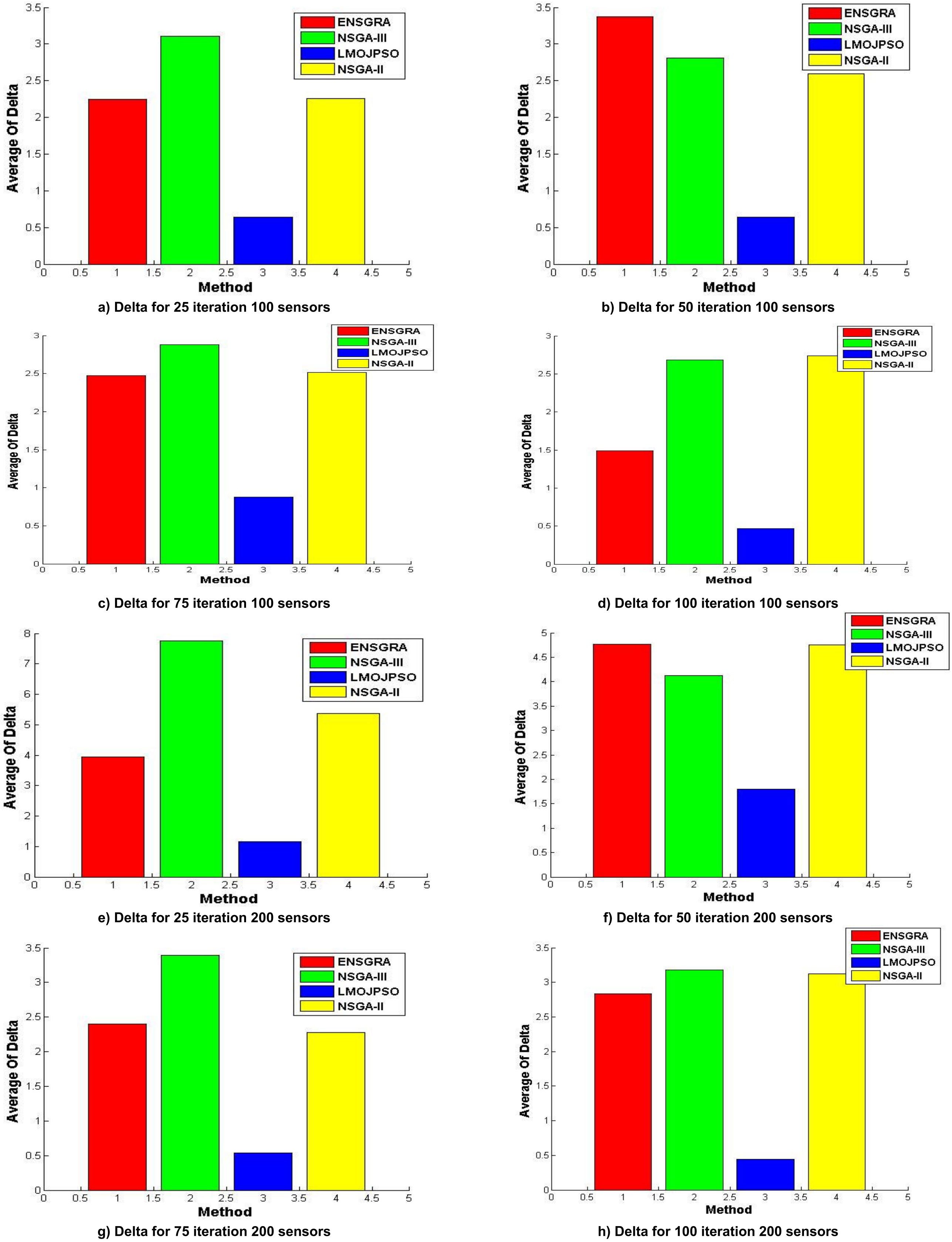
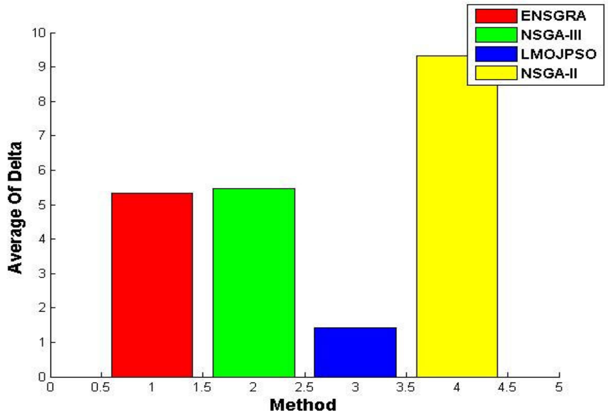
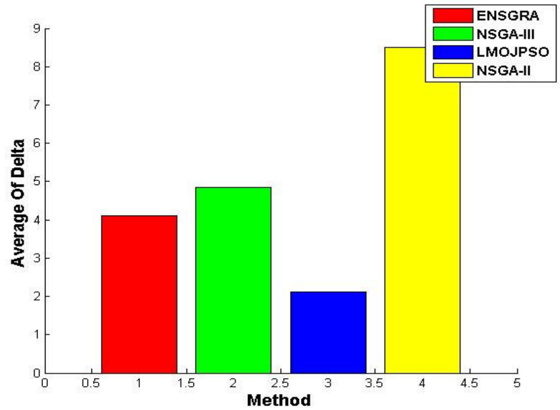


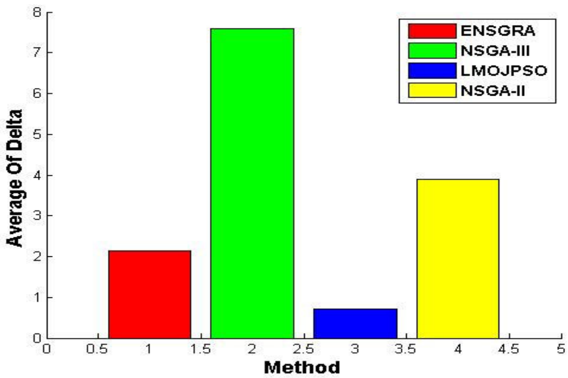
FIGURE 11. Average Delta for four algorithms (methods): from (a-d) 100 sensors using 25, 50, 75, and 100 iteration. From (e-h) for 200 sensors using 25, 50, 75, and 100 iteration. From (i-l) for 300 sensors using 25, 50, 75, and 100 iteration. From(m-p) for 400 sensors using 25, 50, 75, and 100 iteration. From (q-t) for 500 sensors using 25, 50, 75, and 100 iteration.



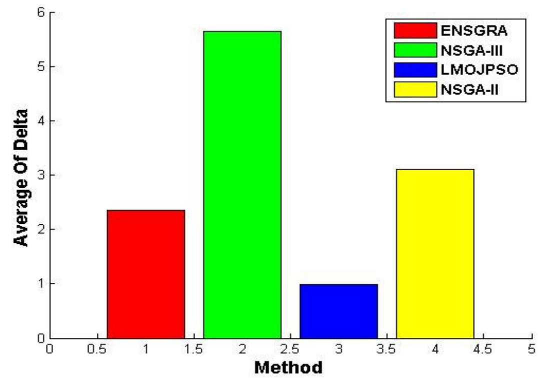
i) Delta for 25 iteration 300 sensors



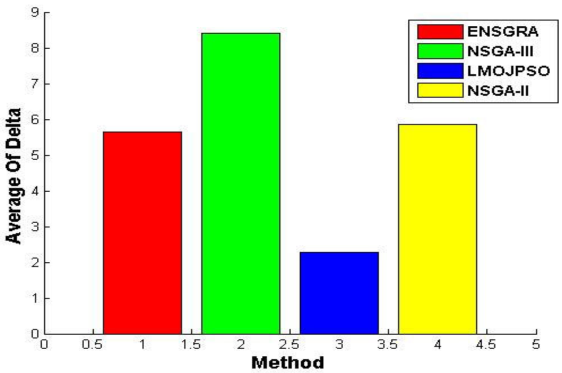
j) Delta for 50 iteration 300 sensors



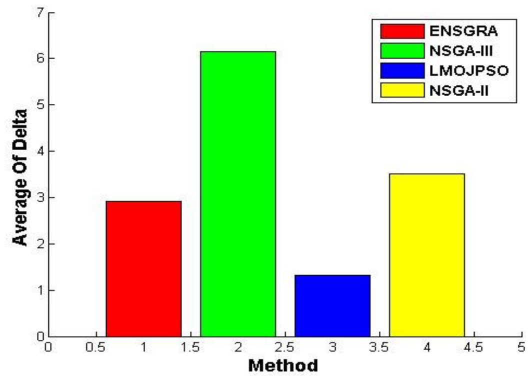
k) Delta for 75 iteration 300 sensors



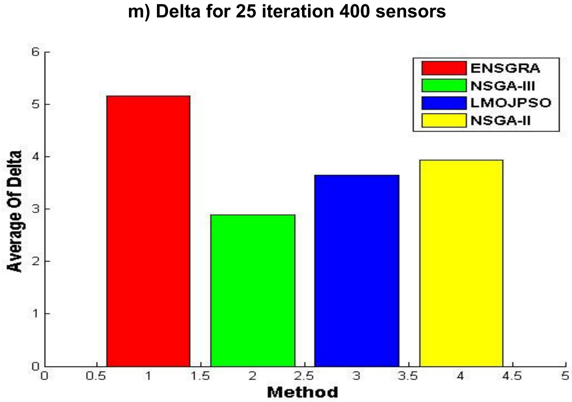
l) Delta for 100 iteration 300 sensors



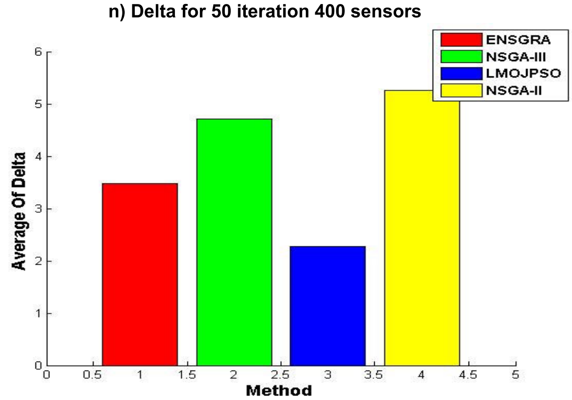
m) Delta for 25 iteration 400 sensors



n) Delta for 50 iteration 400 sensors

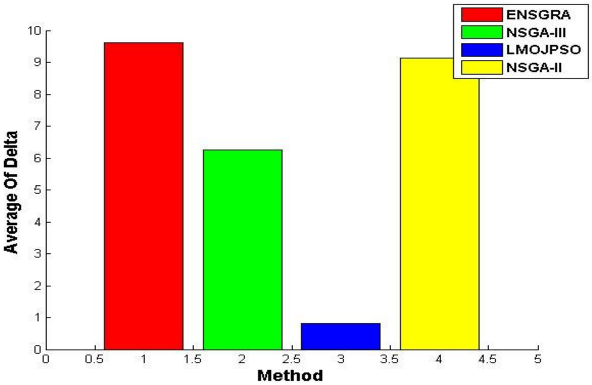


o) Delta for 75 iteration 400 sensors

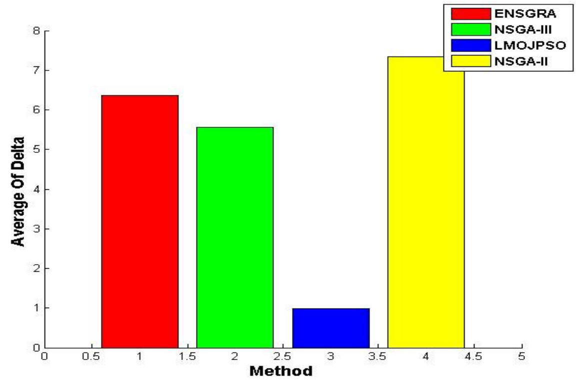


p) Delta for 100 iteration 400 sensors

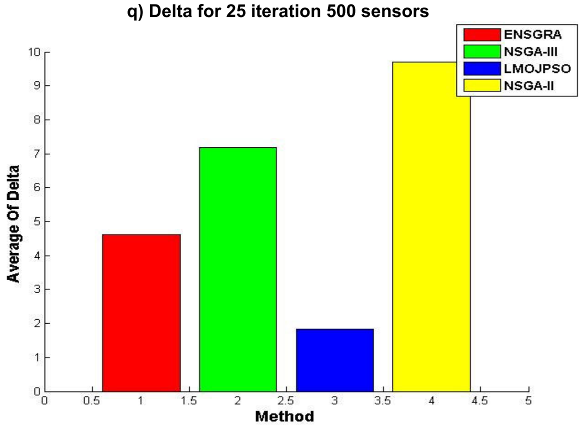
FIGURE 11. *(Continued.)* **Average Delta for four algorithms (methods): from (a-d) 100 sensors using 25, 50, 75, and 100 iteration. From (e-h) for 200 sensors using 25, 50, 75, and 100 iteration. From (i-l) for 300 sensors using 25, 50, 75, and 100 iteration. From(m-p) for 400 sensors using 25, 50, 75, and 100 iteration. From (q-t) for 500 sensors using 25, 50, 75, and 100 iteration.**



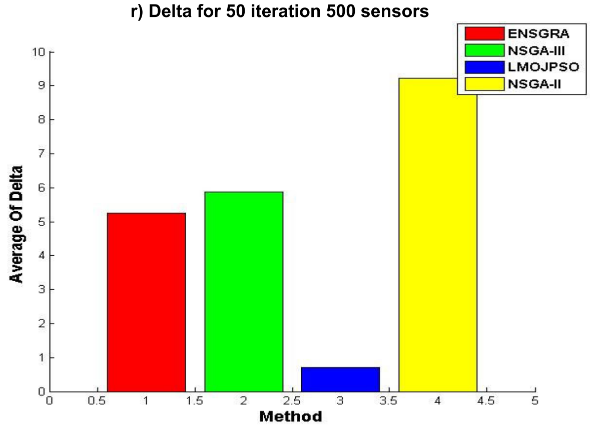
q) Delta for 25 iteration 500 sensors



r) Delta for 50 iteration 500 sensors



s) Delta for 75 iteration 500 sensors



t) Delta for 100 iteration 500 sensors

FIGURE 11. *(Continued.)* **Average Delta for four algorithms (methods): from (a-d) 100 sensors using 25, 50, 75, and 100 iteration. From (e-h) for 200 sensors using 25, 50, 75, and 100 iteration. From (i-l) for 300 sensors using 25, 50, 75, and 100 iteration. From(m-p) for 400 sensors using 25, 50, 75, and 100 iteration. From (q-t) for 500 sensors using 25, 50, 75, and 100 iteration.**

TABLE 5. **Average value for hyper volume, NDS, and delta.**

Algorithm	100 Avg. Hypervolume				200 Avg. Hypervolume				300 Avg. Hypervolume				400 Avg. Hypervolume				500 Avg. Hypervolume			
	25	50	75	100	25	50	75	100	25	50	75	100	25	50	75	100	25	50	75	100
ENSGRA	**9.5**	**15.0**	**7.5**	7.4	6.6	**9.2**	**11.5**	4.5	**13.4**	**12.6**	**8.4**	**6.8**	**7.9**	**4.7**	**5.2**	**8.6**	**9.6**	**6.1**	**5.3**	**5.2**
NSGA-III	3.5	4.6	6.9	7.0	4.2	5.5	5.6	**5.6**	5.0	7.9	6.9	5.4	1.3	2.0	1.0	1.3	6.1	3.7	2.5	3.1
LMOJPSO	0.8	0.8	2.2	0.5	2.7	7.8	0.3	0.1	4.7	10.5	7.4	0.7	1.0	0.7	0.3	0.4	0.2	0.2	0.5	0.1
NSGA-II	5.0	5.9	5.5	**10.1**	**6.9**	6.2	0.5	1.2	6.1	6.6	5.9	**6.7**	4.1	4.0	3.1	3.9	3.8	4.3	4.4	2.5
Algorithm	Avg. NDS				Avg. NDS				Avg. NDS				Avg. NDS				Avg. NDS			
	25	50	75	100	25	50	75	100	25	50	75	100	25	50	75	100	25	50	75	100
ENSGRA	**12**	**13**	**12**	**13**	**11**	**9**	**10**	8	**12**	**15**	**10**	**10**	**8**	**10**	**8**	**9**	**8**	**8**	**8**	**9**
NSGA-III	8	10	11	11	7	9	5	5	10	11	7	7	5	5	5	5	7	6	5	6
LMOJPSO	11	11	11	11	11	9	7	**11**	11	11	8	9	8	9	7	8	**8**	5	**8**	8
NSGA-II	9	12	12	12	9	9	4	4	8	8	7	8	6	7	7	7	6	7	6	6
Algorithm	Avg. Delta				Avg. Delta				Avg. Delta				Avg. Delta				Avg. Delta			
	25	50	75	100	25	50	75	100	25	50	75	100	25	50	75	100	25	50	75	100
ENSGRA	2.2	3.4	2.5	1.5	3.9	4.8	2.4	2.8	5.3	4.1	**2.1**	2.4	5.6	2.9	5.2	3.5	9.6	6.4	4.9	5.1
NSGA-III	3.1	2.8	2.9	2.7	7.8	4.1	3.4	3.2	5.5	4.8	7.6	5.6	8.4	6.2	**2.9**	4.7	6.3	5.6	6.9	5.7
LMOJPSO	**0.6**	**0.6**	**0.9**	**0.5**	**1.2**	**1.9**	**0.4**	**0.4**	**1.4**	**2.1**	5.0	**0.8**	**2.3**	**1.3**	3.6	**2.3**	**0.8**	**1.0**	**1.5**	**0.9**
NSGA-II	2.3	2.6	2.5	2.7	5.4	4.8	2.3	3.1	9.3	8.5	3.9	3.1	5.9	3.5	3.9	5.3	9.1	7.3	7.9	7.5

TABLE 6. QoS model combination percentage (optimization percentage) for the proposed scenarios.

# Sensors	LMOJPSO	NSGA-II	NSGA-III	ENSGRA
100	22%	22%	24%	32%
200	22%	20%	25%	33%
300	23%	25%	23%	29%
400	25%	23%	21%	32%
500	26%	17%	25%	32%
Average	**23%**	**21%**	**24%**	**31%**

for the four algorithms that are used to improve the QoS in WSNs based in three well know indicators Hypervolume (HV), Number of Solutions (NDS), and Delta.

Table 5 shows the average values for HV, NDS and Delta for 100, 200, 300, 400 and 500 nodes using 25, 50, 75 and 100 iteration for ENSGRA, NSGA-III, LMOJPSO and NSGA-II algorithms.

Figures 9 and 10 show that ENSGRA is superior over LMOJPSO, NSGA-II and NSGA-III in terms of HV and NDS in most of the evaluation results. This finding means that the proposed algorithm ENSGRA provides greater convergence, less diversity (distribution) for PF solutions and higher capacity as represented by NDS.

Lastly, Figure 11 shows that the ENSGRA has lower Delta in most of its subfigures than NSGA-II and NSGA-III. However, LMOJPSO overcomes ENSGRA in terms of low Delta. This finding indicates that LMOJPSO provides the lowest distributed solutions. The second in lowest diversity (distribution spread) is ENSGRA, and this result relates to creating equilibrium between convergence and diversity. Diverse solutions are needed to prevent premature convergence and achieve a well-distributed trade-off PF.

Table 5 shows this result that ENSGRA comes first in terms of HV and NDS but second in terms of Delta as illustrated in bold font.

V. CONCLUSION AND FUTURE WORKS

Achieving QoS for WSNs applications, has evolved and become an important and urgent issue to improve WSN performance. Until now, due to the diverse differences in WSN infrastructure, improving these network parameters in deployment and routing have prevailed in most of relevant literature. Nowadays, researches mainly aim to improve multi-object parameters (two or more objects) and solve the MOP, which are considered an NP-hard problem, using meta-heuristic multi-objective optimization algorithms for obtaining solutions for conflicting multi-objectives. In this area, NSGA-II and NSGA-III are considered to be the most popular algorithms to supply Pareto Fronts (PF) solutions, which are provided to the decision maker to determine the best decision.

The need to improve WNS performance is the catalyst behind the idea of proposing an enhanced, non-dominated, sorting genetic routing algorithm. ENSGRA achieves greater convergence and less diversity (spread and distributed) solutions than other algorithms, thereby providing more choices for decision makers. This work proposed the ENSGRA algorithm, which in principle, is based on the NSGA-III algorithm. The ENSGRA allows the proposal of a new QoS model to find a combination between WSNs parameters, followed by adding the computation time. The results are proven to be fruitful, as the proposed algorithm outperforms others in terms of WSN performance metrics parameters with computation time. The multi-objective algorithms evaluation metrics indicate that when compared with NSGA-II and NSGA-III, ENSGRA is superior in terms of Hyper Volume (HV) and Number of Non-Dominated Solution (NDS) indicators. But when Delta (△) is considered, the ENSGRA comes in second place after LMOJPSO; see Table 6 However, when all parameters are combined together, the ENSGRA comes first; Table 6 presents the details of these combination percentage (optimisation percentage) for the proposed scenarios.

Also, observations reveal that most of the previous studies depend on two or three objectives, whilst considering four or more objectives are less likely occurred. Therefore, we plan in the future to consider four-objective to study their performance and add other indicators to evaluate the proposed algorithm.

ACKNOWLEDGMENT

The authors thank Dr. Anas Alzeer, a Lecturer at Birzeit University, and Maher Alndiwee, a Teaching Assistant at Damascus University, for their supports to accomplish this work.

REFERENCES

[1] A. K. Sharma and K. Verma, "NSGA-II with ENLU inspired clustering for wireless sensor networks," *Wireless Netw.*, vol. 26, pp. 3637–3655, Mar. 2020, doi: 10.1007/s11276-020-02281-8.

[2] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 550–586, 1st Quart., 2017, doi: 10.1109/COMST.2016.2610578.

[3] F. Li, M. Liu, and G. Xu, "A quantum ant colony multi-objective routing algorithm in WSN and its application in a manufacturing environment," *Sensors*, vol. 19, no. 15, p. 3334, 2019, doi: 10.3390/s19153334.

[4] M. Iqbal, M. Naeem, A. Anpalagan, A. Ahmed, and M. Azam, "Wireless sensor network optimization: Multi-objective paradigm," *Sensors*, vol. 15, no. 7, pp. 17572–17620, Jul. 2015, doi: 10.3390/s150717572.

[5] E. Zizler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999, doi: 10.1109/4235.797969.

[6] C. A. C. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proc. Congr. Evol. Comput.*, vol. 2, May 2002, pp. 1051–1056, doi: 10.1109/CEC.2002.1004388.

[7] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007, doi: 10.1109/TEVC.2007.892759.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[9] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[10] M. Moshref, R. Al-Sayyed, and S. Al-Sharaeh, "Multi-objective optimization algorithms for wireless sensor networks: A comprehensive survey," *J. Theor. Appl. Inf. Technol.*, vol. 98, no. 14, pp. 4345–4355, 2020.

[11] A. Metiaf, Q. Wu, and Y. Aljeroudi, "Searching with direction awareness: Multi-objective genetic algorithm based on angle quantization and crowding distance MOGA-AQCD," *IEEE Access*, vol. 7, pp. 10196–10207, 2019, doi: 10.1109/ACCESS.2018.2890461.

[12] F. Xia, "QoS challenges and opportunities in wireless Sensor/Actuator networks," *Sensors*, vol. 8, no. 2, pp. 1099–1110, Feb. 2008.

[13] S. Harizan and P. Kuila, "Evolutionary algorithms for coverage and connectivity problems in wireless sensor networks: A study," in *Design Frameworks for Wireless Networks* (Lecture Notes in Networks and Systems), vol. 82, S. K. Das, S. Samanta, N. Dey, and R. Kumar, Ed. Singapore: Springer, 2020, pp. 257–280, doi: 10.1007/978-981-13-9574-1_11.

[14] R. Elhabyan, W. Shi, and M. St-Hilaire, "A Pareto optimization-based approach to clustering and routing in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 114, pp. 57–69, Jul. 2018, doi: 10.1016/j.jnca.2018.04.005.

[15] I. Kasim Ibraheem, I. Department of Electrical EngineeringBaghdad UniversityBaghdad, and A. A.-H. Al-Hussainy, "A multi QoS genetic-based adaptive routing in wireless mesh networks with Pareto solutions," *Int. J. Comput. Netw. Inf. Secur.*, vol. 10, no. 9, pp. 1–9, Sep. 2018.

[16] S. Lalwani, S. Singhal, R. Kumar, and N. Gupta, "A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm," *Trans. Combinatorics*, vol. 2, no. 1, pp. 39–101, 2013.

[17] J. Jia, J. Chen, G. Chang, and Z. Tan, "Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm," *Comput. Math. Appl.*, vol. 57, nos. 11–12, pp. 1756–1766, Jun. 2009.

[18] S. Özdemir, B. A. Attea, and A. Khalil, "Multi-objective clustered-based routing with coverage control in wireless sensor networks," in *Soft Computing*. Berlin, Germany: Springer, 2012.

[19] H. Kumar and S. P. Yadav, "NSGA-II based fuzzy multi-objective reliability analysis," *Int. J. Syst. Assurance Eng. Manage.*, vol. 8, pp. 817–825, Oct. 2017, doi: 10.1007/s13198-017-0672-y.

[20] A. N. Kareem, O. B. Lynn, R. Ahmed, H. Ahmad, and M. Shujaa, "Lagged multi-objective jumping particle swarm optimization for wireless sensor network deployment," *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 2, pp. 423–433, 2019.

[21] S. M. Jameii, K. Faez, and M. Dehghan, "AMOF: Adaptive multi-objective optimization framework for coverage and topology control in heterogeneous wireless sensor networks," *Telecommun. Syst.*, vol. 61, pp. 515–530, Mar. 2015, doi: 10.1007/s11235-015-0009-6.

[22] A. Peiravi, H. R. Mashhadi, and S. Hamed Javadi, "An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm," *Int. J. Commun. Syst.*, vol. 26, no. 1, pp. 114–126, Jan. 2013, doi: 10.1002/dac.1336.

[23] E. B. dos Santos, E. R. Hruschka, Jr., and N. F. F. Ebecken, "Evolutionary algorithm using random multi-point crossover operator for learning Bayesian network structures," in *Proc. 9th Int. Conf. Mach. Learn. Appl.*, Dec. 2010, pp. 430–435, doi: 10.1109/ICMLA.2010.70.

[24] Y. Wang and X. Sun, "A many-objective optimization algorithm based on weight vector adjustment," *Comput. Intell. Neurosci.*, vol. 2018, Oct. 2018, Art. no. 4527968, doi: 10.1155/2018/4527968.

[25] J. Lanza-Gutiérrez, N. Caballé, J. Gómez-Pulido, B. Crawford, and R. Soto, "Toward a robust multi-objective Metaheuristic for solving the relay node placement problem in wireless sensor networks," *Sensors*, vol. 19, no. 3, p. 677, Feb. 2019, doi: 10.3390/s19030677.

[26] A. R. Ohida and H. El-Ocla, "Centralized routing protocol for detecting wormhole attacks in wireless sensor networks," *IEEE Access*, vol. 8, pp. 63270–63282, 2020, doi: 10.1109/ACCESS.2020.2983438.

[27] H. Alwan and A. Agarwal, "MQoSR: A multiobjective QoS routing protocol for wireless sensor networks," *Int. Scholarly Res. Notices*, vol. 2013, Jun. 2013, Art. no. 495803, doi: 10.1155/2013/495803.

[28] K. P. Ferentinos and T. A. Tsiligiridis, "Heuristic dynamic clustering in wireless sensor networks for environmental sensing," PYTHAGORAS-II, Eur. Social Fund Greek Nat. Resour., Tech. Rep., 2006.

[29] H. Zhou, T. Liang, C. Xu, and J. Xie, "Multiobjective coverage control strategy for energy-efficient wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 4, Apr. 2012, Art. no. 720734, doi: 10.1155/2012/720734.

[30] T. Friedrich, C. Horoba, and F. Neumann, "Multiplicative approximations and the hypervolume indicator," in *Proc. 11th Annu. Conf. Genetic Evol. Comput. (GECCO)*, Montreal, QC, Canada, 2009, pp. 571–578.

[31] Y. Caoa, B. J. Smuckerb, and T. J. Robinson, "On using the hypervolume indicator to compare Pareto fronts: Applications to multi-criteria optimal experimental design," Dept. Statist., Univ. Wyoming, Laramie, WY, USA, Tech. Rep. 82071, 2015.

[32] A. Metiaf and Q. Wu, "Wireless sensor network deployment optimization using reference-point-based non-dominated sorting approach (NSGA-III)," in *Proc. J. Phys., Conf.*, vol. 1284, 2019 Art. no. 012063, doi: 10.1088/1742-6596/1284/1/012063.

[33] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon, "Performance indicators in multiobjective optimization," *Eur. J. Oper. Res.*, vol. 292, no. 2, pp. 397–422, Jul. 2021.