Asynchronous Optimization Algorithms for the Enhancement of Lifetime in WSNs

Abdullah Alshahrani, Majed Abdouli, Mohamed Alqarni, Abdullah Alamri, and Ahmed S Alghamdi
Faculty of Computing and IT, University of Jeddah, KSA
Email: {asalshahrani2, mabdouli, Maaalqarni8, Amalamri, ahmedg}@uj.edu.sa

Abstract **—**Efficient utilization of sensor energy to prolong the network lifetime has been the focus of much research on WSNs. In order to save the overall energy of the system and to balance the load among sensors nodes we propose a Modified PSO sleep scheduling algorithm for WSNs. The Modified PSO algorithm includes mutation and crossover functions inspired in GA and searches for both saving-energy and high-coverage rate configurations. In addition, clustering routing protocol is needed to reduce the communication cost and evaluate the network lifetime. In this paper, Fuzzy C-Means (FCM) makes the clustering algorithm. Simulations we have carried out show that by allowing better configuration, the asynchronous scheme greatly increase the WSNs' lifetime.

Index Terms—WSN, Lifetime, modified PSO, fuzzy C-means

I. Introduction

The proliferation of wireless sensor networks (WSNs) requires new application domains for wireless ad hoc networks. WSN consists of sensor nodes that collect sensed data from the environment and communicate via wireless transceivers. The sensor nodes are typically expected to operate with batteries and are expected to be deployed largely in not easily accessible or hostile environments, such as forests or sea. Accordingly, it is not easy to replace the sensor nodes' batteries. Therefore, efficient utilization of sensor energy to prolong the network lifetime has been the focus of much research on WSNs. In order to save the overall energy of the system and to balance the load among nodes, several approaches have been proposed such as, the availability of node, sensors coverage, connectivity, sleep scheduling and clustering routing protocols. These approaches attempt to meet competing objectives. For example, maximization of sensors coverage conflicts with the sleep scheduling. To address different nature of optimization problems relating to WSN lifetime, multi-objective optimization (MOO) plays a key role. Indeed, abundant literature is available where MOO has been used to solve different optimization objectives relating to WSNs. This paper considers how to maximize the lifetime of WSN by studying the sleep scheduling, sensors coverage and clustering routing protocols. The random deployment of sensors can lead to redundant sensor nodes that share the same area. This means that when an event occurs, it can be detected and reported by more than one sensor. These redundant transmissions increase energy consumption. Hence, it is not necessary to activate all nodes simultaneously. The required number of active sensors should be reduced to develop a scheduling mechanism where small number of sensors (objective 1) are active while maintaining maximum coverage and connectivity (objective 2). These two objectives conflict which each other, i.e., great coverage rate increases the number of active nodes and vice versa. In order to obtain these two objectives, a multi-objective hybrid optimization is proposed in this work to enhance the network lifetime by asynchronous scheme of PSO and GA.

This paper proposes a new routing protocol to optimize network lifetime while keeping high-coverage area. It consists of using Fuzzy C-Means (FCM) algorithm to clusterize the network. Then each cluster rotates the cluster head role in order to protect low-energy nodes. Finally, we provide a configurable sleep scheduling based on Particle Swarm Optimization (PSO) that maximizes network lifetime alone, coverage area alone or even a weighted combination of both. The proposed method is hereafter called FCMMPSO (FCM with Modified PSO).

The remaining of this paper is organized as follows. In the next Section, we review the state-of-art related to this topic. Section 3 and 4 present the clustering and sleep scheduling phases respectively and give justification of methods and choices. The results of our simulation experiments are given in section 5. The latest section summarize our contributions and discuss the final remarks.

II. Related Work

There are many drawbacks when sensed data is sent directly from the nodes to the base station. One of them is that nodes farther from the base station get their battery depleted first due to the fact that transmission power is proportional to communication distance. Another issue is that they may spend unnecessary energy sending redundant information (e.g. when the same information sensed by two nodes that are next to each other). This simple setup is hereafter called Direct Communication (DC).

The Minimum Transmission Energy (MTE) routing protocol that opposes to Direct Communication was proposed in [1]. This protocol uses the Dijkstra's

Manuscript received July 30, 2017; revised November 14, 2017.
doi:10.12720/jcm.12.11.636-643

algorithm to calculate the shortest paths between the nodes and the base station. Instead of simply using the Euclidean distance, the path cost is the modeled transmission energy spend in that transmission. Once the shortest path graph is known, the protocol must inform every node of their best next hop, so they can forward the sensed message to the path with less energy expenditure. Depending on the situation, this protocol may quickly overload nodes that are next to the base station (e.g. when the base station is not at the center of the network and few nodes are next to it). As a side-effect, when the nodes next to the base station die, depending on the wireless range of the nodes, farther devices may become unreachable, incurring in a short network lifetime.

There are other solutions in the literature that are based on clustering. Clustering is a plethora of algorithms used for splitting a set of elements into clusters, based on some predefined criteria, and also for finding cluster centers for each split [2]. It is vastly used by statisticians and machine learning engineers as a tool to help with classification in unsupervised learning problems [3].

There are many variations of LEACH [4], [5], but the main idea is to choose the cluster heads depending on a predefined probability. Then the cluster heads advertise their locations to other nodes, which choose the cluster head with the strongest signal (possibly nearer). The cluster heads then act as a router, receiving the messages of those nodes and redirecting them to the base station. They can either redirect the messages on their entirety or filter some information and forward only non-redundant messages. The rotation of the cluster head role balances the energy consumption between nodes, improving the network lifetime.

Another clustering-based solutions are K-Means [6] and FCM [7]. K-Means is a well-known statistical learning technique that clusterize a space based on the distance. The FCM algorithm was first proposed by Dunnin 1973 [8] then improved by Bezdek in 1981 [9] to be used in cluster analysis, especially in pattern recognition. In WSN, it was used to find split the network into clusters based on the Euclidean distance. Fuzzy C-Means is very similar to K-Means in respect to the formation of clusters, but instead of assigning each node to a single cluster, it assigns a node to all clusters with a probability that it belongs to each one of them. Therefore, a node will be assigned to a higher probability of belonging to nearer clusters and to a lower probability of belonging to farther ones.

In FCM [7], once the clusters are formed, the authors propose to use a cluster head rotation mechanism where the first cluster head is the node nearer to the cluster centroids, and after each round the cluster head role is transferred to the node with the highest energy. This rotation, prevents that low-energy nodes assume this power-consuming role. As shown in [7], for a given scenario, this strategy was able to postpone the time that the first node die (surpassing both MTE and LEACH in this aspect).

Apart from clustering and routing protocols, sleep scheduling techniques also have been proposed to improve network lifetime of clustered networks. In [10], the authors present a scheme that assigns a probability of sleeping to every node, where nodes that are farther from the cluster head are more prone to sleep. In [11], the same authors improve their mechanism by proposing a solution where the energy consumption of each node is independent. In [12], a sleep schedule for heterogeneous networks is proposed. An alternative where nodes with more residual energy have more probability of being active is presented in [13].

In [14], an algorithm based on PSO [15] is used to find a sleep scheduling that maximizes network lifetime and network coverage. The authors proposed modifications that are inspired in Genetic Algorithm (GA) [16] in order to improve results. While the authors suppose that the network is clusterized, no particular clustering algorithm is described. The solution presented in Section 4 is inspired in this work, but also investigates the effects of using FCM as clustering and a cluster head rotation mechanism. Additionally, the modified version of PSO described in Section 4 presents differences to the one proposed in [14], including a reduced number of hyperparameters.

In [13], authors propose a hybrid solution that uses FCM to clusterize the network and the original binary PSO as sleep scheduling mechanism. Besides, authors suppose the presence of energy-harvesting nodes to serve as cluster heads, aiming at recharging the excess of energy spent by cluster heads. The results shown in [13] show that this hybrid technique has benefits over LEACH-C [5] and C-FCM [7]. The solution presented in Section 4 differs from the one proposed in [13] in two aspects: it supposes the use of a modified PSO algorithm and it considers that all nodes have the same capabilities, i.e. there are no energy-harvesting devices.

The main reasons to use modified PSO instead of the original PSO are inspired by [14]. One of them is that original PSO has some well-known issues, for instance, premature stagnation. Besides that, we found out that in practice, original PSO always find solutions that lead to shorter network lifetimes than modified PSO, as shown in Section 5.

In [17], NSGA-II [14] (Non-Dominated Sorting Algorithm, version 2) is used to optimize the scheduling of sleep slots. NSGA-II is a multi-objective optimization algorithm based on GA. In Section 5, we show that FCMMPSO results are comparable to ECCA, but usually with a better coverage rate. Besides, the proposed solution allows for the user to choose how much to weight each objective, while in ECCA (as it is), the algorithm cannot prioritize one objective in detriment for another.

III. Clustering Phase

Clustering is a plethora of algorithms used for splitting a set of elements into clusters, based on a predefined criteria, and also for finding cluster centers for each split. It is vastly used by statisticians and machine learning engineers as a tool to help with classification in unsupervised learning problems. In the case of Wireless Sensor Networks, clustering-based protocols can be used to group network nodes into clusters accordingly to the proximity criteria. It means that closer nodes tend to belong to the same cluster. Once the clusters are defined, the nodes nearest to the cluster centers are selected as "cluster heads".

The communication in this network respects the following hierarchy: ordinary nodes communicate only with their cluster head, and base station communicates only with the cluster heads. This hierarchy reduces the average energy dissipated during a peer-to-peer communication, due to the hypothesis that the average distance from ordinary nodes to cluster heads is usually shorter that the average distance to the base station. In cases where this hypothesis does not hold, the clustering technique described here may not be an adequate.

Several clustering-based protocols have been proposed, among them: MTE (Minimum Transmission Energy) [1], LEACH (Low-Energy Adaptive Clustering Hierarchy) [18], the use of K-Means and FCM (Fuzzy C-Means) [5]. Experiments [5] show a comparison of FCM with the following scenarios: direct communication, MTE, LEACH and K-Means. From the average energy dissipated perspective, the FCM approach is the one that scales better with the network diameter, presenting better results over the other techniques with networks with more than 150 meters of diameter. Another significant result is that FCM approach improve the lifetime of the sensor nodes, mainly when compared with the direct communication, MTE and LEACH approaches. Therefore, FCM guarantees a greater coverage of the network during the application lifetime. For more detailed results, please see [5]. In this section, only the FCM protocol is described since it is part of the proposed solution.

The FCM algorithm was first proposed by Dunnin 1973 [8], then improved by Bezdek in 1981 [1] to be used in cluster analysis, especially in pattern recognition. Then in [5], it was applied to routing protocols in order to improve the energy-efficiency in WSNs. Differently from other approaches, the FCM protocol does not always assign a sensor node to a single cluster. Instead, a sensor node has a degree of membership to each cluster, i.e. it has a probability of belonging to every cluster.

In comparison to fixed membership where a node belongs to a single cluster, Fuzzy node assignment can be seen as every node having an ordered list of more suitable clusters. This may be specially interesting when the nodes' distribution changes dynamically or when some nodes get inactive over time (due to battery depletion or to sleeping). In this cases, a node may join another cluster with a lower degree of membership. However, it was not found in the literature any work that exploit this.

A. Determination of the Optimal Number of Clusters

The first step required in clustering is to determine the optimal number of clusters. For this purpose, Eq. 1 [12] is used:

$$C=\frac{\sqrt{N}}{\sqrt{2*\pi}}\sqrt{\frac{Efs}{Emp}}\frac{M}{d_{toBS}^2} \qquad (1)$$

where:

- C is the number of clusters;
- N is the number of sensor nodes;
- Efs is the energy spent in the transmission of a single bit of data through free space, achieving an acceptable bit error rate;
- Emp is the energy spent in the transmission of a single bit of data through a multipath fading model, achieving an acceptable bit error rate;
- M is the network diameter;
- d_{toBS} is the average Euclidean distance from the cluster heads to the base station.

Notice also that Efs and Emp are dependent on the distance of transmission. Determination of the clusters's centroid and cluster initialization.

B. Determination of the Cluser's Centroid and Cluster Initialization

Once the number of clusters is defined, the FCM algorithm help to determine both the cluster centroids and the initial assignment of sensor nodes to clusters. For that purpose, the method minimizes the following objective function (Eq. 2):

$$J_m=\sum_{i=1}^{C}\sum_{j=1}^{N}u_{ij}^m d_{ij}^2, 1\le m<\infty \qquad (2)$$

where U_{ij} is the coefficient representing the degree of membership of the node i w.r.t. the cluster j, d is the Euclidean distance between the i^{th} node and the j^{th} cluster centroid, and m is a real parameter that represents the fuzziness of the clusters. The uij coefficients form a coefficient matrix U where i indexes the i^{th} row and j indexes the j^{th} column.

The coefficients of the U matrix are calculated as follows (Eq. 3):

$$u_{ij}=\frac{1}{\sum_{k=1}^{C}\left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{m-1}}} \qquad (3)$$

where d_{kj} is the Euclidean distance between the k^{th} sensor node and the jth cluster centroid. Eq. 3 shows that the greater the distance between a node and a centroid, the smaller the respective coefficient will be. The matrix U is initialized with random samples from a uniform distribution with values ranging from 0 to 1 (representing a probability).

The cluster centroids are iteratively calculated using Eq. 4:

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{4}$$

where x_i is the geolocation of the i^{th} node and c_j is the geolocation of the j^{th} cluster centroid.

The complete algorithm is show in **Algorithm 1**. Lines 1-3 show the initialization of the matrix U ($U_{(0)}$). Lines 6-11 represent an iteration, where for each cluster the centroid is iteratively calculated using Eq. 4 (line 9). The membership matrix is updated at each iteration using Eq. 3 (line 10). Finally, the FCM algorithm stops either when the error is below some threshold ε, or when the algorithm had run for a certain number of iterations.

Algorithm 1 FCM algorithm for cluster formation

```
1   for i = 0 to N
2     for j = 0 to C
3       u_ij^(0) ~ uniform(0,1)
4
5   k ← 0
6   repeat
7     k ← k + 1
8     for j = 0 to C
9       update cluster centroid c_j using Eq. 4
10    update U^(k) using Eq. 3
11  until ||U^(k) – U^(k-1)|| < ε
```

After completing, the algorithm has defined the degree of membership of nodes. Then, each node is initially set to belong to the cluster that it has the highest degree of membership.

C. Cluster Head Selection and Election

The base station initially selects the node nearest to each cluster centroid as the cluster head. However, the network hierarchy overloads cluster heads: they must relay messages from every cluster node to the base station. Thus, they expend energy faster than ordinary nodes. For this reason, the cluster head selection must be dynamic in order to prevent those nodes to extinguish their energy supplies and to balance the energy load.

Therefore, after the first round, the current cluster head elects another node to be the new cluster head. It choses the node with the highest residual energy (this information is sent to the cluster head in every packet) and nearest to the cluster centroid. This procedure is repeated every exchange of data, meaning that the cluster head role is reassigned periodically. It is also important to notice that the FCM algorithm is only used at the initial setup of the network. From that moment on, the partitions (clusters) are fixed while the cluster head changes dynamically.

The cluster head is responsible for allocating time slots when cluster members can transmit. This is implemented using a TDMA schedule. Regarding power consumption, cluster members activate their radio component only during their own slots, reducing the power consumption. Also, transmission power is optimized since the Euclidean distance between nodes and cluster head is minimized by the FCM algorithm. Additionally, energy is saved due to the fact that data aggregation and fusion is executed at the cluster head, which compresses the information sent to the base station.

IV. Sleep Scheduling

PSO [15] is a computational method that optimizes an objective function by searching the solution space simultaneously with multiple "probes", aiming to improve the proposed solution iteratively. It is inspired by the social behavior of living being herds, where each "probe" is called particle and it represents a candidate solution. Particles move through the solution space looking for a local optima with a certain velocity. At each iteration, each particle velocity is updated proportionally to the particle's own inertia (algorithm's parameter) but is also partially redirected to the local (particle's) best known position and to the global (swarm's) best known position. Therefore the particle movement (velocity) has three components: an inertial one, a personal one and a social one; simulating the herd's behavior.

PSO has been successfully used in several domains such as [6], [19]-[22]. Specifically, in the WSN applications, PSO was introduced by Kulkarni *et al*. [23], who presented PSO and showed now to use it for energy-saving purposes.

Algorithm 2 Modified PSO Algorithm

```
for each cluster do
  for i = 1 to P do
    p_i ← initialize_particle()
    l_i ← p_i
    if i = 1 or fitness(l_i) > fitness(g) then
      g ← l_i

  for it = 1 to M do
    for i = 1 to P do
      r_p, r_g ~ uniform(0.0, 1.0)
      p_i ← mutation(p_i, ω)
      if r_p < φ_p then
        p_i ← crossover(p_i, l_i)
      if r_g < φ_g then
        p_i ← crossover(p_i, g)

      if fitness(p_i) > fitness(l_i) then
        l_i ← p_i
      if fitness(p_i) > fitness(g) then
        g ← p_i
```

Algorithm 2 describes the modified PSO used in this paper. It is based on [14], which is inspired itself in the Genetic Algorithm. For each cluster, a different set of particles is generated and each set searches to maximize the sleep scheduling for that cluster independently. Each particle has N dimensions, where N is the number of nodes in that cluster. Each dimension is binary, meaning that it has only to points: 0 (for active nodes) and 1 (for nodes that will be put to sleep). Every particle dimension is randomly initialized with 0 or 1 with equal probabilities (0.5) (line 3). Then the i^{th} particle's best position l_i is set to the particle's initial configuration p_i (line 4). Then the swarm's best position g is set to the best initial position (lines 5-6). In order to decide which particle has the best configuration, the fitness function is used (described later on this section). P is the number of particles that simultaneously searches for the solution.

Lines 8-20 describe the search for the best sleep configuration. This procedure is repeated for a certain number of iterations (M), which is found out in practice. Then for each particle, three basic steps are repeated: mutation (line 11), crossover with the particle's best position (line 13) and crossover with the swarm's best position (line 15). In each step, particle and swarm's best positions are updated when the new particle reached better results in terms of the fitness function (lines 17-20).

In the modified PSO, the number of particles P keeps unchanged, meaning that every particle is mutated and suffers crossover but no additional particles are generated during this process.

The mutation and crossover functions are inspired in GA and are described in Algorithm 3 and Algorithm 4. Both functions differ from the functions described in [13], because our results shown that we achieved better learning (of particles' positions) with these changes.

The mutation function changes some of the dimensions of a particle (which is called in GA, individual). The function traverses the individual genes and statistically flips (Boolean negation) ω percent of the genes. The mutation allows for the particle to "move" in the search space, assuming new configurations. Therefore, larger ω leads to greater movements.

The crossover function takes two particles (individuals) as input and the resulting individual has statistically half of genes from one individual and half from the other. Crossing over allows the particle to be attracted towards its best position or towards the swarm's best position.

The fitness function that the modified PSO tries to maximize is described by Eq. 5:

$$fitness = \alpha * f_1 + \beta * f_2 \tag{5}$$

Algorithm 3 Mutation function

Function *mutation (individual, ω)*
  for *g = 1 to G* **do**
    r_1 *~ uniform(0.0, 1.0)*
    if r_1 < ω **then**
      flip(individual$_g$)
  return *individual*

Algorithm 4 Crossover function

Function *crossover (father, mother)*
  r_1 *~ uniform(0.0, 1.0)*
  for *g = 1 to G* **do**
    if $r_1 < 0.5$ **then**
      $child_g \leftarrow father_g$
    else
      $child_g \leftarrow mother_g$
  return *child*

It is the weighted sum of two terms, with parameters α and β. The first term is a function of the remaining energy at devices and it is different from the energy-related term used in [19] because we found out that in some situations using that function makes the modified PSO find that the best configuration is to put all nodes to sleep. In order to avoid that, instead of minimizing the sum of energies of all wake nodes, Eq. 12 maximizes the probability that nodes with above-average energy levels are kept awake, while nodes with below-average energy levels are put to sleep. This improves energy-balance inside the cluster. In Eq. 16, E_g is the energy of the g^{th} node, G is the number of nodes (genes in GA terminology), p_i is the i^{th} particle and E is the average energy in the alive nodes.

$$f_1 = \frac{\sum_g^G (1-p_{i,g}).(E_g-\bar{E}) - \sum_g^G min(E_g-\bar{E},0)}{\sum_g^G max\left((E_g-\bar{E}),0\right) - \sum_g^G min(E_g-\bar{E},0)} \tag{6}$$

The second term is the fitness function is related to the coverage and the overlapping areas (as defined in Section 3) and it is described in Eq. 13. $Regions_{exclusive}$ are the exclusive regions for a particular particle p, and $Coverage_{all}$ are the coverage area (see Section 3) for all alive nodes. Differently from [20], we only have one term related to coverage and overlapping areas instead of two, reducing the number of the algorithm's parameters. The value of f_2 may range from 0 (when awake nodes represented in the particle have no exclusive area) and 1 (when all awake nodes cover areas that no other node covers). Maximizing f_2 maximizes coverage area while reducing the overlapping at the same time. Therefore, the nodes that have more overlapping are more prone to be sleeping.

$$f_2 = \frac{Regions_{exclusive}(p)}{Coverage_{all}} \tag{7}$$

α and β should sum to 1 and are chosen depending on how much one values the network lifetime(term 1) and coverage area (term 2).

The modified PSO (Algorithm 2) has 3 hyperparameters ω, φ_g and φ_g. The learning strategy was implemented according to [20]:

$$\omega = \omega_{max} - (\omega_{max} - \omega_{min}) * it/M \tag{8}$$

$$\phi_p = 1 - 1 * it/M \tag{9}$$

$$\phi_g = 1 - \phi_1 \tag{10}$$

V. Results and Discussion

The settings common to all scenarios are described bellow (except when mentioned differently).

The 2-dimensional field where 300 nodes are distributed is 250x250 m^2 and the nodes are distributed using an independent uniform distribution for axis x and y. The base station is at (125, 125), i.e. the center of the field. The message length is 4000 bits plus a 150 bit header. The coverage radius for every node is 20 meters. The fuzzyness factor is equal to 2, and the number of clusters is usually calculated as 5. The initial energy at every node is 2 Joules.

Regarding the modified PSO settings, the fitness function α is set to 0.5 and β to 0.5 by default. The maximum number of iterations for modified PSO is 100 and the number of particles is set to 20.

In all simulations, except for the Direct Communication setup, we considered that the base station must regularly broadcast routing/clustering information and sleep scheduling in order to every node know how to proceed in the next round. This forces nodes to spend at least the energy to receive this information (considering a 4150 bit message). Some authors consider that the cluster head perform this functionality while other authors do not consider this step in the energy consumption.

It is also important to note that the results shown in this section do not consider the initial energy spend in case nodes need to send their position to the base station. For the settings described above, this expenditure is equal to 0.303 Joules net or 0.001 Joules per node (in average).

A. Scenario 1: Cluster Formation

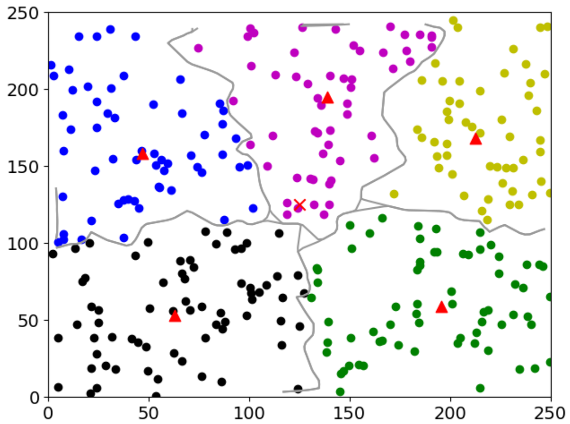

Fig. 1. FCM cluster formation.

In this scenario, we show the cluster formation generated by the FCM algorithm (Fig. 1). Nodes are represented as points and each color indicates a different cluster membership. Gray solid lines are the boundaries of each cluster. Red triangles represent the cluster centroids while the red cross represent the base station. As mentioned earlier, each node in FCM belong to all clusters with a certain probability. In this plot, we assigned nodes to the most probable cluster.

Nodes are represented as points and each color indicates a different cluster membership. Gray solid lines are the boundaries of each cluster. Red triangles represent the cluster centroids while the red cross represent the base station. As mentioned earlier, each node in FCM belong to all clusters with a certain probability. In this plot, we assigned nodes to the most probable cluster.

B. Scenario 2: Network Lifetime

In this scenario, DC, MTE, LEACH, FCM, FCM plus original PSO and FCMMPSO are analyzed from the network lifetime perspective. The results are shown in Fig. 2.

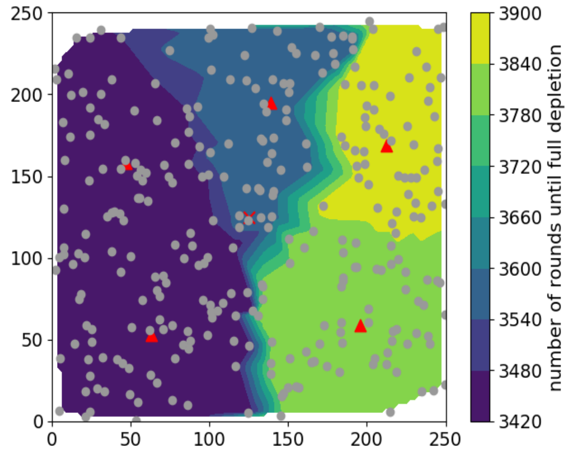

Fig. 2. Map with time of depletion for FCMMPSO.

As expected, using DC makes nodes farther from the base station deplete their batteries first, but nodes nearer to the base station outlive any other algorithm. When using LEACH, nodes tend to survive longer than DC ones. The cluster rotation mechanism used for LEACH randomly selects the next cluster head (as [9]). This mechanism does not take node's remaining energy into account. For this reason it performs poorly.

On the other hand, FCM rotates the cluster head role between higher-energy nodes, what improves the network lifetime. Clusters deplete their energies as a whole, causing step descents in the number of alive nodes.

Concerning MTE, nodes nearer to the base station forward messages of all other nodes that are farther from the base station. Therefore, these nodes are overloaded and are depleted quickly. When these nodes are off, the farther nodes have to communicate directly to the base station, and due to the effort of transmitting over larger distances, also get their batteries depleted.

As expected, FCM with original PSO perform better than other methods that rely only on clustering. This is straightforward since original PSO decides that some nodes should occasionally sleep, saving their batteries.

As shown, FCMMPSO outperforms all other strategies. It seems that the modified PSO was able to always find solutions that lead to improved network lifetime. It is important to note that FCMMPSO performed better in all simulations that we run, but are not included here for lack of space. Also, the settings used in these comparisons were exactly the same (including values for α and β).

In order to detail the results shown in Fig. 3, Table I shows the time when the first node depletes its battery and the time when 30% of the nodes die, respectively. Also we included results (for the same distribution/position of nodes) where the base station is decentralized at (125, -75), i.e. out of the map.

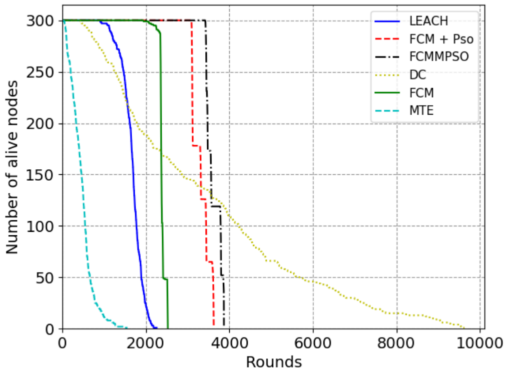

Fig. 3. Number of alive nodes vs time (in rounds).

Fig. 2 shows the map where the nodes are distributed. Red triangles represent the cluster centroids while the red cross represent the base station. Each area is colored according to the time (round) when the surrounding nodes get their batteries depleted. Dark areas represent nodes that die earlier, while light colors represent nodes that die later in the network lifetime. The base station is at the center of the map i.e. at (125, 125).

Since the base station is at the center of the map, those nodes are the last to die (they spend less energy communicating with the base station). As it can be seen, most of the clusters die almost at the same time (have the same color in the map). This means that the network lifetime lasts longer but also that all nodes die approximately together.

TABLE I: ROUND WHEN FIRST NODE DIE AND WHEN 30% OF NODES DIE (FOR DIFFERENT ALGORITHMS)

Algorithm	**Base station at:**			
	(125, 125)		*(125, -75)*	
	Number of depleted nodes:			
	1	*30%*	*1*	*30%*
DC	430	1685	11	97
MTE	29	312	28	141
LEACH	906	1585	459	1247
FCM	1943	2363	1500	1595
FCM+PSO	3088	3117	1872	1955
FCMMPSO	3422	3477	2033	2102

VI. CONCLUSION

This paper presented a new scheme to improve network lifetime while maximizing network coverage area. FCM is used for network clustering by splitting the network in small subsets where transmission power is kept low. Additionally, each cluster has a head rotation mechanism that seeks to prevent low-energy nodes from communicating directly to the base station, resulting at a better energy-balance. Better configurations of sleep schedules are found using the modified PSO algorithm that searches for both saving-energy and high-coverage rate configurations. Results show that, by correctly setting up PSO hyperparameters, the algorithm is able to learn better configurations and thus provide meet the specifications of applications with different goals.

REFERENCES

[1] M. Ettus, "System capacity, latency, and power consumption in multihop-routed SS-CDMA wireless networks," in *Proc. Radio and Wireless Conference (RAWCON)*, August 1998, pp. 55-58.

[2] C. Yu, W. Guo, and G. Chen, "Energy-balanced sleep scheduling based on particle swarm optimization in wireless sensor network," in *Proc. IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, Shanghai, 2012, pp. 1249-1255.

[3] C. J. Lin, C. H. Chen, and C. T. Lin, "A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 39, no. 1, pp. 55-68, January 2009.

[4] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, pp. 32-57, 1973.

[5] H. Dhawan and S. Waraich, "A comparative study on LEACH routing protocol and its variants in wireless sensor networks: A survey," *International Journal of Computer Applications*, vol. 95, no. 8, 2014.

[6] F. V. den Bergh, "An analysis of particle swarm optimizers," Ph.D Eng. thesis, Department of Computer Science, University of Pretoria, 2002.

[7] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney, "Balanced-energy sleep scheduling scheme for high density cluster-based sensor networks," *Elsevier Computer Communications Journal, Special Issue on ASWN*, vol. 28, no. 14, pp. 1631-1642, September 2005.

[8] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney, "Scheduling sleeping nodes in high density cluster-based sensor networks," *Mobile Networks and Applications*, vol. 10, no. 6, pp. 825-835, December 2005.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[10] M. R. Pearlman, J. Deng, B. Liang, and Z. J. Haas, "Elective participation in ad hoc networks based on energy consumption," in *Proc. IEEE Global Telecommunications Conference*, November 2002, pp. 26-31.

[11] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660-670, 2002.

[12] S. Chelbi, H. Dhahri, M. Abdouli, C. Duvallet, and R. Bouaziz, "A new hybrid routing protocol for wireless sensor networks," *International Journal of Ad Hoc and Ubituitous Computing*, 2016.

[13] L. Tan, Y. Gong, and G. Chen, "A balanced parallel clustering protocol for wireless sensor networks using K-means techniques, hskip 1em plus 0.5em minus 0.4em," in *Proc. Second International Conference on Sensor Technologies and Applications*, Cap Esterel, France, August 2008, pp. 25-31.

[14] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, "A hybrid particle swarm optimization for distribution state estimation," *IEEE Transactions Power System*, vol. 18, pp. 55-57, Nov. 2002.

[15] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum Press, 1981.
[16] J. Holland, *Adaptation in Natural and Artificial Systems*, Cambridge, MA: MIT Press, 1992.
[17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, Apr. 2002.
[18] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edition, In Springer Series in Statistics, Springer New York, 2009.
[19] J. Jia, J. Chena, G. Changa, and Z. Tan, "Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm," *Computers and Mathematics with Applications*, vol. 57, pp. 1756–1766, 2009.
[20] S. Sekhar, "A distance based sleep schedule algorithm for enhanced lifetime of heterogeneous wireless sensor networks," Master's thesis, University of Cincinnati, 2005.
[21] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless sensor networks," in *Proc. 33rd Annual Hawaii International Conference on System Sciences*, Hawaii, USA, Jan. 2000.
[22] D. C. Hoang, R. Kumar, and S. K. Panda, "Fuzzy C-Means clustering protocol for wireless sensor networks," in *Proc. IEEE International Symposium on Industrial Electronics*, Bari, 2010, pp. 3477-3482.
[23] W. L. Du and B. Li, "Multi-strategy ensemble particle swarm optimization for dynamic optimization," *Information Sciences*, vol. 78, no. 15, pp. 3096-3109, August 2008.

Abdullah Alshahrani is a PhD candidate at the Catholic University of America. He is also a lecture at University of Jeddah. He earned his master degree from La Trobe University, Melbourne Australia in 2010 in computer since and networking. His area of interests are computer networks, communication and security.

Majed Abdouli is an assistant professor at the FCIT, University of Jeddah, Saudi Arabia. He has graduated with a Ph.D in 2006 from Le Havre University, France that dealt with real-time data management and timeliness/consistency issues in distributed computer systems. Currently, his main research topics of interest are distributed multimedia applications, real-time data freshness, and lifetime in sensor networks.

Mohammed A. Alqarni is currently the vice dean of Faculty of Computing and Information Technology at University of Jeddah. He has graduated recently with a PhD from McMaster University, Canada. His main research topic was modeling concurrent systems. However, he is now more interested in the wide field of Networking

Abdullah Alamri holds his bachelor degree in computer science from King Khalid University, in 2007, and received master degree of Information Technology from School of Engineering & Mathematical Sciences, La Trobe University, Australia in 2009 and his Ph.D in computer science from RMIT University in 2014. He is currently assistant professor in the Faculty of Computing and Information Technology at the University of Jeddah.

Ahmed is assistant professor in the department of information system. He graduated from the Department of Electrical Engineering and Computer Science of the School of Engineering at The Catholic University of America in Washington, DC with an emphasis in network security. He earned his master of science in formation system from DePaul University in Chicago in 2010.