ISSN 1726-5479

SENSORS & TRANSDUCERS

7 vol. 130 / 11

Sensor Networks and Wireless Sensor Networks

International Frequency Sensor Association Publishing

IFSA

Sensors & Transducers

Volume 130, Issue 7, July 2011 www.sensorsportal.com ISSN 1726-5479

Sensors & Transducers Journal (ISSN 1726-5479) is a peer review international journal published monthly online by International Frequency Sensor Association (IFSA).
Available in electronic and on CD.

Sensors & Transducers Journal

Contents

Volume 130
Issue 7
July 2011

www.sensorsportal.com

ISSN 1726-5479

Research Articles

Authors are encouraged to submit article in MS Word (doc) and Acrobat (pdf) formats by e-mail: editor@sensorsportal.com
Please visit journal's webpage with preparation instructions: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm

International Frequency Sensor Association (IFSA).

Sensors & Transducers Journal, Vol. 130, Issue 7, July 2011, pp. 61-76

Sensors & Transducers

ISSN 1726-5479

http://www.sensorsportal.com

Extending Coverage and Lifetime of *K*-coverage Wireless Sensor Networks Using Improved Harmony Search

Shohreh Ebrahimnezhad, Hoda Jalal Kamali and Mohsen Ebrahimi Moghaddam
Electrical and Computer Engineering Department
Shahid Beheshti University; G.C; Tehran, Iran
Tel.: +98-21-2990-2268, fax: +98-21-2397-9896
E-mail: sh_ebrahimnezhad@yahoo.com, h_kamali_87@yahoo.com, m_moghadam@sbu.ac.ir

Received: 1 June 2011 /Accepted: 22 July 2011 /Published: 29 July 2011

Abstract: *K*-coverage wireless sensor networks try to provide facilities such that each hotspot region is covered by at least *k* sensors. Because, the fundamental evaluation metrics of such networks are coverage and lifetime, proposing an approach that extends both of them simultaneously has a lot of interests. In this article, it is supposed that two kinds of nodes are available: static and mobile. The proposed method, at first, tries to balance energy among sensor nodes using Improved Harmony Search (IHS) algorithm in a *k*-coverage and connected wireless sensor network in order to achieve a sensor node deployment. Also, this method proposes a suitable place for a gateway node (Sink) that collects data from all sensors. Second, in order to prolong the network lifetime, some of the high energy-consuming mobile nodes are moved to the closest positions of low energy-consuming ones and vice versa after a while. This leads increasing the lifetime of network while connectivity and *k*-coverage are preserved. Through computer simulations, experimental results verified that the proposed IHS-based algorithm found better solution compared to some related methods. *Copyright © 2011 IFSA.*

Keywords: Wireless sensor networks, Improved harmony search, *K*-coverage, Lifetime, Mobile sensors.

1. Introduction

A wireless sensor network (WSN) consists of a large number of sensor nodes that are small in size and distributed in a region in uncontrolled and unorganized way [1]. These tiny nodes consist of sensing, data processing, communicating components, and leverage the idea of sensor networks [2]. They communicate with each other in a self organized manner and have limited resources (such as processing speed, storage capacity, energy, and communication bandwidth). Applications of wireless sensor networks include environmental and habitat monitoring, precision agriculture, surveillance, asset tracking, and healthcare [3]. Maximizing coverage is desirable in order to be able to monitor more regions and it is an essential requirement has been investigated in some literature reports [4, 5].

In addition, hotspot areas in a sensor network are more important regions than others and should be covered by more sensors. If all hotspot areas are covered by at least *k* sensors, then the network is called *k*-coverage. As a node sends, receives, or forwards packets; its energy is reduced accordingly, and once the energy level falls below a threshold, it shutdowns completely [6]. Battery-powered wireless sensor nodes are greatly constrained with regards to energy supply, so energy efficiency becomes a critical problem in WSNs [5].

Also, in a WSN, a Sink summarizes collected data and presents them to users or sends them to a remote host. As all nodes send data to the Sink, determining the position of Sink has great influence on the delivery latency and network efficiency [1].

There are several definitions for the lifetime of WSN in the literature, but the most common which is used as well in this paper is defined as the time till the node with minimum remained energy dies or the time to first failure and the network lifetime is equal to the minimum lifetime of all nodes [7, 8], [9-11]. In [12], the authors proved sensor scheduling for *k*-coverage is a NP-hard problem and [9] shows that maximizing lifetime is NP-hard. Also, in order to prolong the lifetime of a WSN, a placement should be found where nodes have smallest connecting paths to Sink node. What's more, energy balancing helps that less sensors are moved and therefore lifetime is extended.

In the present paper, a meta-heuristic algorithm is employed for extending coverage and lifetime that considers all of above points. The proposed method applies Improved Harmony Search (IHS) to find a near optimal solution for the coverage and lifetime of *k*-coverage WSN. This algorithm is an improvement of Harmony Search algorithm which can dynamically adjust the parameters of HS [13], [14]. It is worth mentioning that in [15], we had proposed an IHS-based algorithm which is used to find a node placement in a connected and *k*-coverage WSN which saves more energy while it can optimize covered area. This research gave more covered area compared to genetic algorithm based solutions and also original Harmony Search algorithm, moreover it could reduce total energy consumption but it did not investigate lifetime of the network. In this way, without attending energy balancing, after a while, some sensors are dead and the network would not work anymore while total remained energy is still good. So, in the present paper, we have focused on maximizing lifetime by defining the objective function of IHS to have a node deployment that is not only connected and *k*-coverage but also have a geometrically symmetric shape and gets optimal coverage and lifetime. This symmetric, circle-like shape lets us locate Sink at the center of the shape as a good position which have shortest path to all other nodes. Uniform distribution of nodes around the Sink helps load balancing among nodes. Furthermore, for further load balancing, the place of most consuming nodes is exchanged with less consuming ones. In other words, nodes that will shutdown early are replaced with close and high energy ones, after a time. These high and low energy sensors are selected as mobile ones. Also, the placement of whole network does not alter after the best placement is found and so connectivity and *k*-coverage are preserved.

Balancing energy, finding the suitable position for sink, using both static and mobile sensors and relocating mobile sensors using IHS-based algorithm are the novelties in this method to extend coverage and lifetime. Simulation results verified that the proposed IHS-based algorithm found better solution than original HS algorithm and GA-based methods in case of coverage.

The rest of the paper has been organized as follows. Section 2 introduces some related works. Section 3 defines extending coverage and lifetime problem in *k*-coverage wireless sensor networks and proposes an IHS-based method to balance energy among nodes for optimizing coverage and lifetime. The simulation results are reported in section 4 and the paper is concluded in section 5.

2. Related Works

Several research efforts have been done to improve scheduling of wireless sensor networks. For example, Huang et al. formulated *k*-coverage problem as a decision problem that its goal is to determine whether every point in the service area of the sensor network is covered by at least *k* sensors using polynomial-time algorithms [16]. S. Megerian et al. presented an optimal polynomial time worst and average case algorithm for coverage calculation that combined computational geometry and graph theory techniques (Voronoi diagram and graph search algorithms) [17]. Another research could optimize coverage in a connected and *k*-coverage wireless sensor network using genetic algorithms but it did not investigate network lifetime [18]. Olston *et al.* minimized the total communication cost between data sources and the data sink [19]. In another work, a novel sink selection scheme is presented which take transmission latency and sink forwarding latency into consideration [20]. However these researches could reduce total energy consumption but they do not attend to energy balancing so after a while, some sensors are dead and the network would not work anymore while total remained energy is still good. The effect of energy balancing in extending lifetime is stated in some works such as [9].

In [21], energy consumption is balanced by regulating communication frequency among sensor nodes. Also, Wang et al. proposed deploying redundant static nodes and the redundant nodes can be turned off initially and be turned on later when coverage holes appear upon sensor failures [22]. In [23], Mei et al. used mobile robots for sensor network repairing. In their proposed method, sensor nodes detect failures of their neighbors and report them to robots. Then robots move and replace failed nodes with functional ones. Another research proposed relocating redundant mobile sensor nodes to fill the coverage holes and using a cascading movement method to balance the energy cost and the repairing time [24]. In [25] a distributed particle swarm optimization is proposed to optimize energy consumption using mobile sensors. However, these works did not consider *k*-coverage criterion. Another work used a sensor deployment for a *k*-coverage wireless sensor network by a heuristic approach with the purpose of maximizing network lifetime but some parameters which are needed in a sensor scheduling did not consider in their method [26]. R. Katsuma et al. could extend WSN lifetime while maintaining *k*-coverage of the target field by using mobile sensors with GA algorithm but the coverage is not survived in this work [27].

3. IHS-Based Proposed Solution: Optimizing Coverage and Lifetime in *K*-coverage WSNs

In a *k*-coverage WSN, there are 3 items as prerequisites. The first one is that all sensor nodes must be located in the defined area. The second item is preserving connectivity between sensor nodes and the third one is that every hotspot region must be covered by at least *k* sensors. For formulating a WSN, we defined a 2D area with *length* × *width* dimension and the more important regions are considered as hotspot areas. Also, it is assumed that all of the sensor nodes are homogenous and they have identical sensory and communication ranges. In addition, Sink is a sensor with higher energy and its position is calculated so as its distance to other sensor nodes be low. The problem is coverage and lifetime improvement in such area regarding to the mentioned conditions.

For increasing the lifetime of wireless sensor network, balancing energy in initial placement is an important issue. When a WSN operates for a long time, batteries of some sensor nodes like the nodes close to the Sink will be exhausted and connection in the network will be broken. So by replacing mobile nodes lifetime is increased while *k*-coverage would be preserved. Also, if the energy difference between the sensor nodes is low, then the WSN can work longer. On the other hand, energy balancing helps that fewer sensors are moved and therefore lifetime would be extended. For example, in a large scale network that there are many sensor nodes, the density of nodes around Sink should be more than other places. One might argue that placing all nodes directly next to the Sink will eliminate the need for forwarding packets received from the farther nodes and therefore less energy will be consumed. To reply this, it should be reminded that we have to satisfy the other important constraint of the problem which is maximizing coverage so nodes should have minimum overlap with each other.

Here, Improved Harmony Search algorithm is proposed to have a *k*-coverage WSN with maximum coverage and lifetime.

3.1. Improved Harmony Search Contexts

IHS algorithm consists of the following five steps which are stated below and also are shown in Fig. 1.

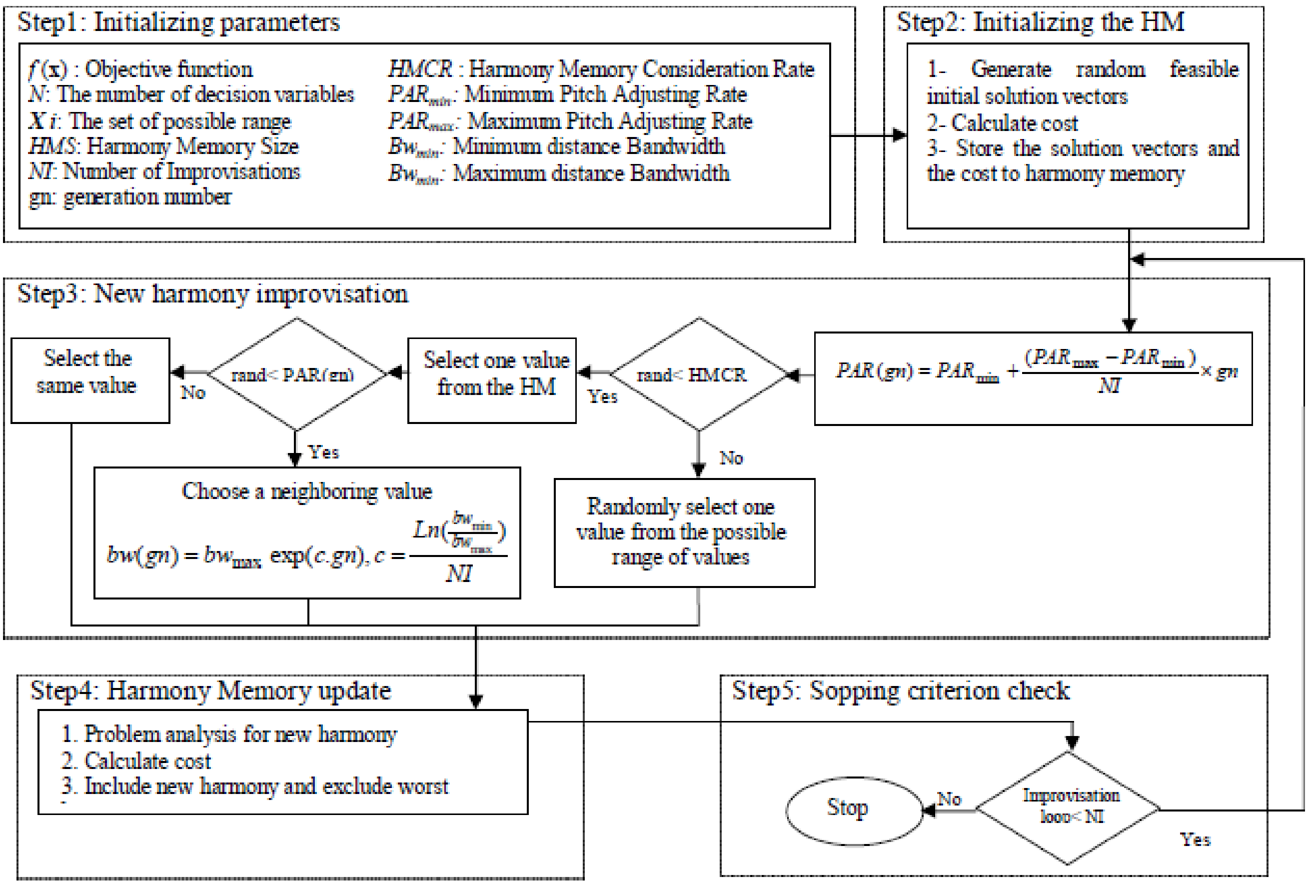


Fig. 1. Optimization procedure of the improved harmony search algorithm.

3.1.1. Step 1: Initializing the Problem and Algorithm Parameters

The first step is to specify the problem and initialize the parameter values. An optimization problem is defined as Minimizing (or Maximizing) *f*(*x*) such that $x_i \in X_i$, $i = 1,2,..., N$ where *f(x)* is a scalar objective function to be optimized; *N*, is the number of decision variables; *x* is a solution vector consisting of each decision variable x_i; X_i is the set of possible range of values for each decision variable that is ${}_L x_i \leq X_i \leq {}_U x_i$ as ${}_L x_i$ is the lower and ${}_U x_i$ is the upper limit for each decision variable. The parameters of the IHS are shown in Table 1.

Table 1. The parameters of the HIS.

Parameter	**Statement**
HMS	Harmony Memory Size
HMCR	Harmony Memory Consideration Rate
PAR(gn)	Pitch Adjusting Rate for each generation
$PAR_{\min}$	Minimum Pitch Adjusting Rate
$PAR_{\max}$	Maximum Pitch Adjusting Rate
NI	Number of Improvisations or stopping criterion
Gn	Generation number
bw(gn)	Distance bandwidth for each generation
$bw_{\min}$	Minimum bandwidth in the Harmony Memory
$bw_{\max}$	Maximum bandwidth in the Harmony Memory

3.1.2. Step 2: Initializing the Harmony Memory (HM)

In this step, the Harmony Memory (HM) matrix which is a storage where collects all the solution vectors (sets of decision variables), is filled with randomly generated solution vectors equal to HMS as illustrated in equation (1).

$$\left[\begin{array}{ccccc|c} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 & f(x^1) \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 & f(x^2) \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS-1} & x_N^{HMS-1} & f(x^{HMS-1}) \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} & f(x^{HMS}) \end{array}\right] \quad (1)$$

3.1.3. Step 3: New Harmony Improvisation

As Harmony Search algorithm imitates the improvisation process of musicians to find the perfect state of harmony, a new feasible harmony solution, $x' = (x'_1, x'_2, \ldots, x'_N)$ is generated based on three following operators: (1) Memory consideration, (2) pitch adjustment, and (3) random selection [28].

Analogy between Improvisation process in IHS or seeks to find optimum solution with the stated rules and music improvisation seeks to find musically pleasing harmony are explained as follows:

1. Memory consideration or choosing any one value from the Harmony Memory mimics playing any one note from existing memory.
2. Pitch adjustment or choosing a neighbor value of one value from the Harmony Memory mimics performing variations on an existing piece.
3. Random selection or choosing totally random value from the possible range of values mimics creating an entirely new piece.

IHS algorithm includes a number of optimization parameters such as the *HMCR* (Harmony Memory Considering Rate) that is the rate of considering one value from the previous values stored in the HM.

In each iteration, a random value between 0 and 1 is obtained and if it is more than *HMCR*, memory consideration should be done, otherwise the primary new harmony would be a random vector which is obtained from the possible range of values.

In the memory consideration, the value of the first decision variable (x') for the new harmony vector is chosen from any of the values that already exist in the specified HM range ($x_1'^{1} - x_1'^{\mathrm{HMS}}$). Values of the other decision variables $(x_2', x_3',..., x_N')$ are chosen in the same manner.

Another parameter is *PAR (gn)* which is the Rate of Pitch Adjustment for every component obtained by the memory consideration. It is used to determine whether the component should be pitch-adjusted or not.

Pitch adjusting decision is expressed as follows:

$$x_i' \leftarrow \begin{cases} x_i' \pm rand(0,1) \times bw(gn) & \text{with probability PAR(gn),} \\ x_i' & \text{with probability (1 - PAR(gn)).} \end{cases} \quad (2)$$

In this equation, rand (0, 1) is a uniformly distributed random number between 0 and 1, *gn* is generation number and, *bw (gn)* is an arbitrary distance bandwidth.

PAR (gn) and *bw (gn)* are required to be calculated for each iteration in IHS and are shown in equation (3) and (4). In the Improved Harmony Search, *PAR (gn)* is increased linearly while *bw (gn)* is decreased exponentially with generation number. The used parameters in the following equations have been defined in step 1.

$$PAR(gn) = PAR_{\min} + \frac{(PAR_{\max} - PAR_{\min})}{NI} \times gn \quad (3)$$

and

$$bw(gn) = bw_{\max}\ \exp(c.gn),\ c = \frac{Ln(\frac{bw_{\min}}{bw_{\max}})}{NI} \quad (4)$$

3.1.4. Step 4: Harmony Memory Update

In step 4, the IHS algorithm evaluates the new harmony solution by calculating its objective function value. If the value of this new harmony vector $x' = (x_1', x_2',..., x_N')$ is better than the worst value (x^i) in the harmony matrix then the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

3.1.5. Step 5: Stopping Criterion Check

In this step, the stopping criterion (maximum number of iterations=*NI*) which is the pre-selected maximum number of cycles is checked and if it is satisfied, computation is terminated. Otherwise, steps 3 and 4 are repeated.

At first, IHS algorithm tries to find a node placement where energy is balanced among the sensors. After NI iterations, there are some best results in Harmony Memory. Then, after a while, as a greedy method the high energy consumption nodes are relocated with the low ones which are close to them and with this way the best mobile nodes are found to increase the lifetime of network.

3.2. Applying IHS to Find Best Node Placement for Energy Balancing

In this section, all parameters are initialized. In addition to IHS parameters that were defined earlier, the extending coverage and lifetime problem involves some parameters that are listed in Table 2.

Table 2. The parameters of sensor nodes.

Parameter	**Statement**
s.energy	Initial energy amount of each node= 32400 J (two AA batteries) (by referring to [27])
U_{elec}	Power consumption coefficient for data processing= 50 nJ/bit (by referring to [27])
$\in_{amp}$	Power consumption coefficient for signal amplification= 100 pJ/bit/m2 (by referring to [27])
U_{move}	Power consumption coefficient for moving 8.267 J/m (by referring to [29])
E_{move}	8.267 J/m (by referring to [29])
packet_size	Size of data for sensed information=128 (by referring to [1])
sense_frequency	0.1 Hz; 0.1 per second (by referring to [1])

At first, one sensor is placed in each hotspot as a guideline and the other sensor nodes are distributed randomly in the defined area and in this way Harmony Memory is initialized according to step 2 of IHS algorithm. Every solution vector which is a node placement, has as many elements as the number of nodes and each element contains *x* and *y* coordinate of sensor node. In other words, each row of the HM makes a separate harmony and this matrix are crammed with *x* and *y* coordinates of sensor nodes plus the objective function and lifetime assigned to it. Next, connectivity and *k*-coverage are checked for each row of HM and the objective function values are calculated. Depth First Search (DFS) algorithm is used to check network connectivity for each harmony to find out whether there is a path between a specified sensor to other ones. For preserving connectivity between two sensors, their distances should be equal or less than communication radius and this value is calculated by Euclidean distance. In disconnected graphs, if a network has less connected components, and the distance between them is lower, it is a better graph. So we decided to define *connectivity_parameter* item consists of number of components and their distances as the objective function for the disconnected harmonies. The *connectivity_parameter* value is calculated according to equation (5) and is returned for disconnected harmonies as objective function.

$$connectivity\_parameter = -CMN \times CMD \quad (5)$$

Here CMN is number of components and CMD is an approximate total value of distances between components that is calculated by selecting two close sensors of each two components.

Also, the hotspot area is divided into uniform grids where each point in this area corresponds to a grid point and for all points it is checked if at least *k* sensors are placed. Placing nodes in a completely randomly manner will satisfy *k*-coverage in more iterations so it was decided to give a guideline to the algorithm such that each hotspot and initial Harmony Memory are filled with one node and such placements, respectively. Preference between connected and *k*-coverage harmonies is given based on the amount of coverage and lifetime. The optimization problem consists of maximizing coverage and lifetime so the objective function should reflect them as well. According to the definition of lifetime, lifetime is maximized when the energy consumption is balanced among nodes or if the difference between energy of nodes is minimized. It is obvious that distance between nodes has effect on their energy consumption. This brings to mind using the standard deviation of nodes distance (Std(d)) so if it is minimized then we could anticipate that their batteries will be exhausted at the same time so the lifetime will be prolonged. So an objective function is defined as follows:

$$ObjectiveFunction \leftarrow \begin{cases} connectivity\_parameter & \text{if the harmony is disconnected,} \\ -1 & \text{if the harmony is connected and it is not } k\text{-coverage,} \\ \dfrac{coverage}{(std(d))} & \text{if the harmony is connected and } k\text{-coverage.} \end{cases} \quad (6)$$

In order to calculate the covered area, the sensory area is divided into uniform grids where each point in this area corresponds to a grid point. For every point of this grid, it is checked if it is included in the sensing range of a node or not and then all detected points are summed up to approximate the coverage.

In addition, for each node or each decision variable of each harmony, the lowest cost path to the sink is calculated which will be used for calculating consumed energy of the node. In multi-hop routing, each node sends a message to the closest node on the way to the Sink node and this routing is less energy consuming than direct transmission. Dijkstra algorithm which can find minimum path between one single vertex and all the other vertexes on the connected graph is used to find the lowest cost path to transfer data from every node to the Sink. Then the lifetime of each sensor node is calculated according to equation (12) and the minimum lifetime of all nodes is considered as the network lifetime and is assigned to harmony row. The following formulas are required to calculate lifetime.

Powers $E_{Trans(x,\,d)}$, $E_{Recep(x)}$ are used to transmit *x* bit to *d* meters and receive *x* bit, regarding to the following equations, respectively [30]:

$$E_{\mathrm{Trans}(x,\,d)} = (U_{elec} \times x) + (\in_{amp} \times x \times d^2) \quad (7)$$

$$E_{\mathrm{Recep}\,(x)} = U_{elec} \times x \quad (8)$$

where U_{elec} and $\in_{amp}$ are constants representing the power required for information processing and the amplification power, respectively.

Also, E_{Sens} and E_{Listen} are required energy to sense the information which is packet_size [bit] data and listen to whether radio messages come or not for one second, respectively, as follows [27]:

$$E_{\mathrm{Sens}} = (U_{elec} \times packet\_size) + U_{\mathrm{Sens}} \quad (9)$$

$$E_{\mathrm{Listen}} = U_{\mathrm{Listen}} \quad (10)$$

The energy consumption of each sensor node per unit of time that is defined *Cper* is calculated by equation (11). Based on consumed energy of each node, its lifetime is calculated according to formula (12) and then minimum lifetime of all nodes is considered as the network lifetime.

$$Cper = (E_{\mathrm{Sens}} + E_{\mathrm{Recep}} + E_{\mathrm{Trans}(x,d)}) \times sens\_frequency + E_{\mathrm{Listen}} \quad (11)$$

$$nodeLifetime = s.energy/Cper \tag{12}$$

Here, *nodeLifetime* is the lifetime of each sensor node.

Then a new harmony is made according to step 4 and HM is updated. If objective function of the new harmony is better than the worst value of the Harmony Memory then the values of this row are replaced with new harmony. For example, if the new harmony is a connected and *k*-covered network then if the quotient of coverage divided by Std(d) of this harmony is more than the worst value in HM then this new harmony is replaced with the worst one in Harmony Memory as it is expressed in equation (6). Then termination criterion is checked. After NI iterations, the IHS algorithm will stop and a symmetric node deployment is achieved. In this step, the location of Sink could be found. As this symmetric shape is circle-like then by placing sink at the center of this circle, nodes will have equal distance from it and the longest path from all other nodes to it will be the smallest. By using Dijkstra algorithm, the node whose longest path to all other ones is less than others, is selected as Sink.

3.3. Moving Nodes to Improve Lifetime

By obtaining a symmetric placement of nodes after NI iterations, the network works as long as its lifetime is anticipated but in order to further prolong its lifetime we would change the place of most consuming nodes and less consuming nodes with each other, after a time. These nodes are selected as mobile nodes. It is stated in [31] that the lifetime of a WSN can be improved by deploying some mobile sensors around the Base Station (BS) and our simulation results show that most consuming nodes are always around the sink which are selected as mobile nodes.

Mobile nodes consume battery power not only for communication but also for movement. U_{Move} is a constant and Power $E_{\mathrm{Move}(d)}$ is required to move *d* [m] regarding to formula (13) [27].

$$E_{\mathrm{Move}(d)} = U_{Move} \times d \tag{13}$$

Then new lifetime is calculated as follow:

$$NewLF = CurrentTime + ((LEnergy - E_{\mathrm{Move}(d)})/Cper) \tag{14}$$

Here, *NewLF* is the new calculated lifetime, *CurrentTime* is current lifetime which is chosen to be half of the primary anticipated lifetime. In fact moving nodes is done far before the battery of any of the nodes is exhausted and *LEnergy* is the leftover energy of each sensor.

All required parameters for calculating are defined in 3.2. After relocating the mobile nodes, the new lifetime is placed in the assigned row of the HM.

4. Simulation Results

At first, for examining the proposed method for WSN optimization we considered 12 sensors in 400×400 sensory area and used IHS algorithm to extend coverage and lifetime. The parameters of the problem were initialized with the determined values in 3.1. *Rad* or radius of sensory range of each sensor is considered 50 distance units and *Dia* is defined as the diameter of a circle that it is twice of *Rad*. Also, we considered 1…3 hotspot areas in experiments. Each hotspot area is considered in form of a circle in experiments and its radius is equal to the sensory radius of sensors and its center was entered by the user while the value of *k* was variable between 1…3.

HMS, NI and *HMCR* were assumed 15, 300 and 0.95 respectively and *bw (gn)* and *PAR(gn)* are calculated in every iteration of IHS algorithm according to equation (3), (4).

Fig. 2 illustrates a schema of placement using IHS with 12 nodes determined as high and low consuming nodes where the farther nodes rout data back to Sink via a multi-hop method by the sensors near the Sink. Also a hotspot region is shown with a circle in this area.

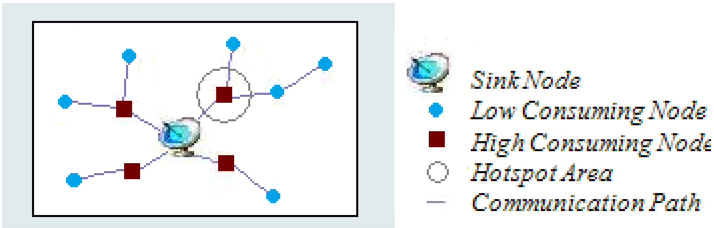
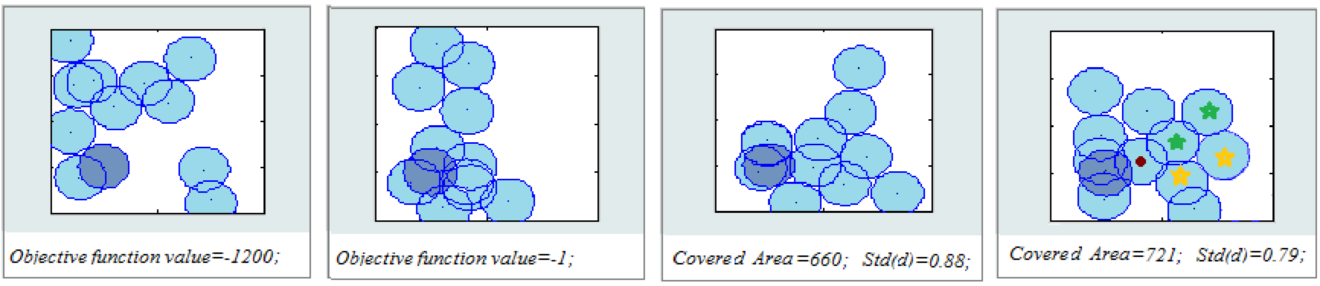


Fig. 2. Placement with 12 sensor nodes while high and low consuming nodes are determined.

Improvement of the algorithm in terms of coverage and lifetime, by increasing the number of iterations is shown in Fig. 3. In this figure, the results for k=2 are shown and the hotspot area is indicated as a gray circle. Fig. 3(a) illustrates that in 10^{th} iteration, a connected graph is not created, therefore the output schema can not provide connectivity and a negative value is reported for the objective function according to equation 6. As shown in Fig. 3(b), IHS algorithm could find a connected graph in 50th iteration but the hotspot is not k-coverage and -1 is reported as the function value. Also, Figs. 3(c) and 3(d) show that by increasing the number of iterations, Standard deviation of nodes distance among all nodes decreases and the algorithm tries to make a symmetric schema so the energy is balanced and lifetime is improved. Furthermore, coverage value is increased as well.



(a) Iteration=10 (b) Iteration=50 (c) Iteration=100 (d) Iteration=300

Fig. 3. Improvement in coverage and lifetime (by decreasing Std(d)) when Iteration Number increases; Sensory area=400×400; Sensory radius=50; Nodes number =12; Hotspot areas Number =1; and k-coverage degree=2.

In addition, to further balance energy consumption among nodes, nodes placed next to the sink (at distance less than *Dia*) could be exchanged with the nodes next to them (by path at distance between *Dia* and twice of *Dia*) from the Sink if we consider some of the nodes mobile. These nodes are indicated with a star and Sink is shown with red point in Fig. 3(d). As this figure shows the nodes with the orange stars should be exchanged with each other and nodes with the green stars should be exchanged with each other.

Fig. 4 illustrates two node placements obtained by different objective functions. In Fig. 4(a) the goal of minimizing sum of consumed energies of nodes is included in objective function while in Fig. 4(b) minimizing Standard deviation of consumed energy of all of the nodes is included in objective function, and the goal of maximizing coverage is also included in both cases. As Fig. 4(a) shows there are just 2 nodes next to the Sink which alone should relay the information received from nodes that are by path at distance between *Dia* and twice of *Dia* from the Sink and they have to relay the information received from the nodes by path at distance between twice of *Dia* and three times of *Dia* and so on. On the other hand, in Fig. 4(b) the task of relaying other farther nodes to the sink is done by 6 nodes which are next to the sink so the load is balanced among them. Sink is shown by a star and the neighbors of it are illustrated with pink points in Figs. 4(a) and 4(b). By using the previously stated objective function as it is expressed in equation (6), nodes not only tend to be placed in a symmetric circle-like shape that leads to load balancing but also they have minimum overlap with each other that leads to maximizing coverage.

(a) Objective Function is minimizing sum of nodes energies.

(b) Objective Function is minimizing sum of nodes energies.

Fig. 4. Effect of objective function on node placement schema; Sensory area=400×400; Nodes number =12; Sensory Radius=50 and Generation number=500.

As stated before, another contribution of this paper is determining best position for Sink. Imagine a case where there are 3 hotspots located as far as possible in the sensing field. This case is shown in Fig. 5. Hotspots are shown as pink circles and one of them which is the primary selected Sink as well, is shown as a gray point. After applying the algorithm, the following placement of nodes is achieved. The algorithm tries to connect all 3 hotspots while covers the area as much as possible. Then the place of Sink is found in 192 and 265 coordinate which are shown as a red point. It is almost at the center of the nodes deployment as it might be anticipated.

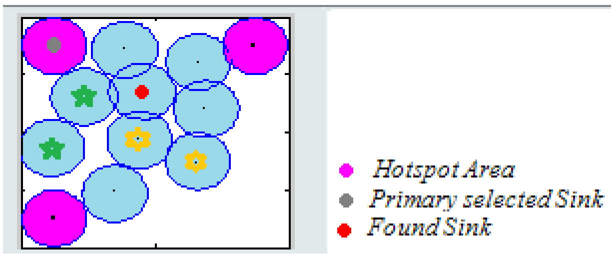


Fig. 5. Finding best position for Sink; Sensory area=400×400; Sensory Radius=50; Number of sensors=12; Number of hotspot area=3 and *k*-coverage degree=1.

After achieving a placement inside a circle-like shape (we also call it a symmetric placement, symmetric related to the Sink) and after a time that the network has worked and the nodes next to the sink have done the task of relaying other nodes packets to the Sink, it is the time to exchange their place with the nodes next to them and placed farther from the Sink (by path at distance between *Dia* and twice of *Dia*). Imagine node *i* which is placed at distance less than *Dia* from the sink is neighbor to 3 other nodes and relays the packets of these 3 nodes to the sink. Among these 3 nodes the one whose consumed energy is a lot less than that of *i* is chosen for exchanging with node *i*. The simulations show that best results are achieved when most energy consuming node is exchanged with less consuming one and these two nodes should be just beside each other because as equation (13) shows the movement energy is proportional to the distance, therefore, two nodes which are at most at distance *Dia* from each other have been selected. Also pairs of nodes that should be exchanged with each other are shown with green and orange stars.

Fig. 6 shows the effect of exchanging nodes on lifetime. As it shows lifetime is at least increased two times in all of the cases and in some cases it is increased 6 times (when primary lifetime is 1325 and it reaches 7622 hours after exchanging two pairs of nodes). According to experimental results, maximum lifetime is obtained when two pairs of nodes are exchanged. In Fig. 6, axis X shows the primary calculated lifetime which is obtained before exchanging nodes, red line shows the calculated lifetime after exchanging the place of one pair of nodes and blue line shows the calculated lifetime after exchanging two pairs of nodes and this is performed at the middle of primary anticipated lifetime. For example, if lifetime is calculated to be 2077 hours then after 1038 hours nodes are moved.

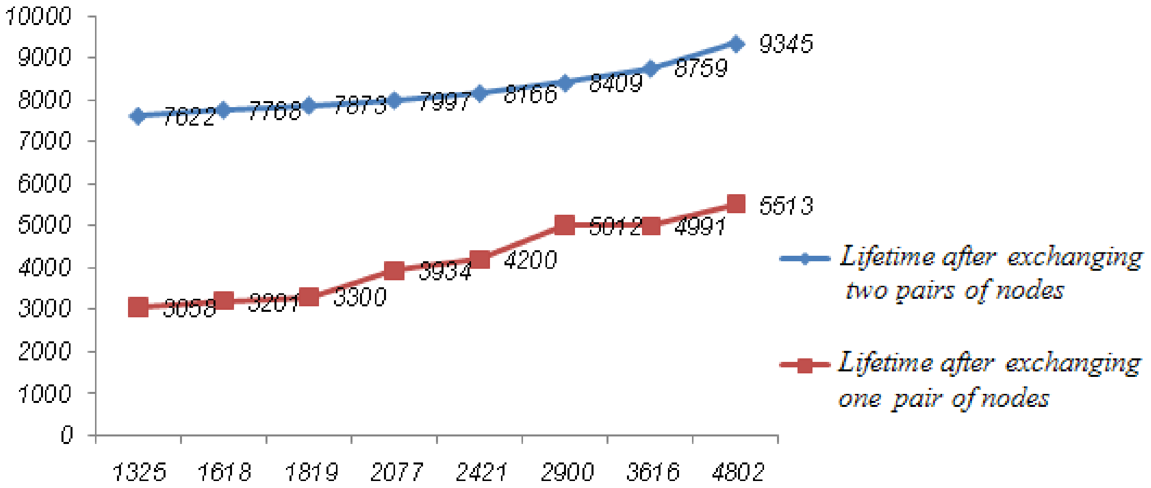


Fig. 6. The effect of exchanging nodes on increasing lifetime.

Experimental results show that best improvement in lifetime is achieved when two pairs of nodes are exchanged with two others. It means that almost 30 percent of nodes should be mobile, a result which is consistent with the results of R. Kastuma et al. [27]. They showed that when 25 percent of nodes are mobile best results are achieved.

In another experiment, the performance of the proposed method in terms of lifetime is indicated in the case of k=1 and when node number has been increased. In this experiment, a symmetric shape was obtained as well. This is shown in Fig. 7 for the case when number of nodes is 50. In the bellow placement, the most distance between nodes and Sink is in a path three times *of Dia*. In this situation, nodes at distance less than *Dia* from sink should be exchanged with nodes in a path at distance between *Dia* and twice of *Dia*. Also nodes in a path at distance between *Dia* and twice of *Dia* from sink should be exchanged with the nodes that have a path at distance between twice of *Dia* and three times of *Dia* from sink in order to improve lifetime.

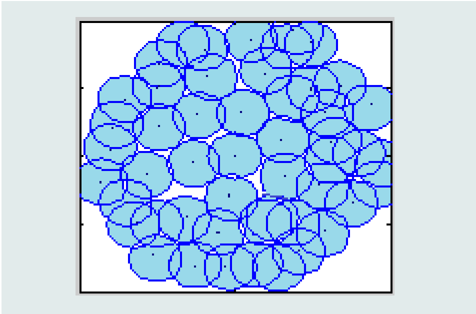


Fig. 7. Placement with many sensor nodes and obtaining a symmetric schema; Sensory area=400×400; Sensory Radius=35; Number of sensor nodes=50 and *k*-coverage degree = 1.

Improvement in lifetime is shown in Fig. 8 for different number of nodes. The algorithm tries to exchange as much as nodes while the lifetime is improved. Experimental results show that when nodes neighbor to the sink are exchanged by nodes next to them, most improvements in lifetime is achieved. Also, exchanging further nodes does not improve lifetime a lot.

In the following diagram the red line shows initial lifetime achieved before exchanging nodes and green line shows lifetime after exchanging the place of most consuming nodes with less ones which are next to them. Most consuming nodes are among nodes next to the Sink because they should do the task of relaying other nodes information to the sink.

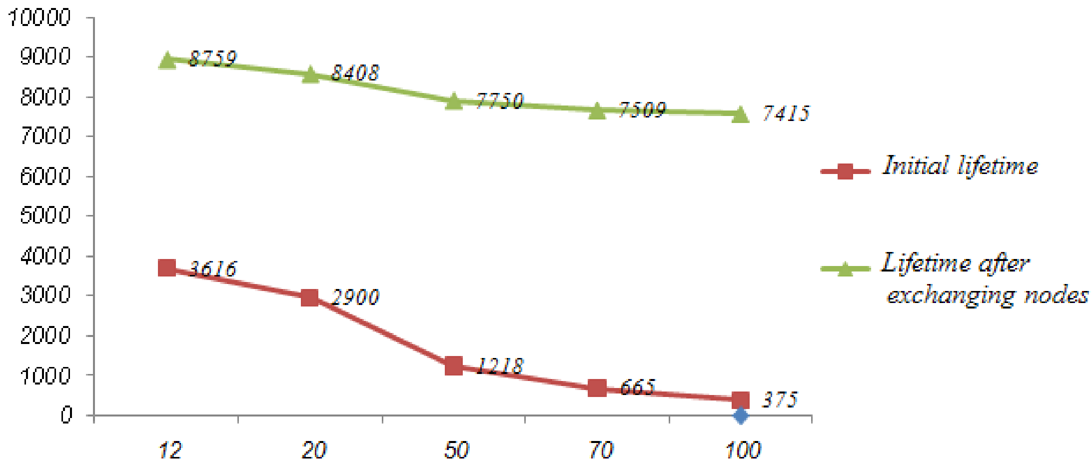
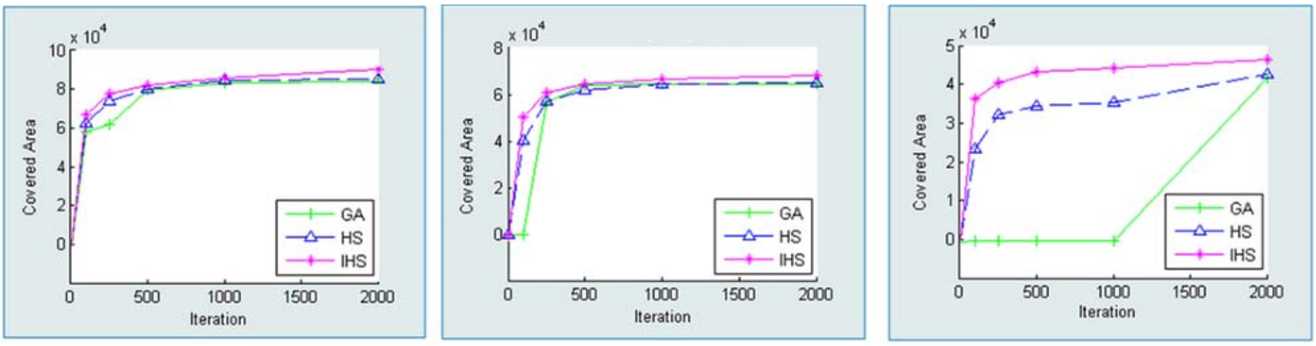


Fig. 8. Improvement in lifetime for different number of nodes.

In order to evaluate the performance of Improved Harmony Search in term of coverage, we have compared the coverage target field of the best result of proposed algorithm in various values of k with the results of presented method by T. Emre Kalayc et al. when *sensor_number*, *Rad*, and *NI*, were considered 12, 50, 2000 and the value of *k* was variable between 1..3 [18]. In addition, we implemented original HS in the same platform when *HMCR*=0.90 and *PAR*=0.3 in classic HS algorithm. Comparison among the results obtained by GA, the original Harmony Search method, and Improved Harmony Search has been reported in the Fig. 9 [18]. As this figure shows IHS algorithm finds a better optimal solution compared to GA and HS algorithms in same number of generations and its performance is more visible when it can make a connected network in less iteration when *k* is increased.



(a) *k*-coverage degree =1 (b) *k*-coverage degree =2 (c) *k*-coverage degree =3

Fig. 9. Comparison between the results obtained by the IHS, HS and GA.

Experimental results confirmed that the proposed IHS-based method by balancing energy and relocating mobile nodes provides an optimal sensor network distribution in case of lifetime while the coverage of network is better than two other approaches which are GA and HS algorithms [18].

5. Conclusions

In this paper, the lifetime and coverage of a *k*-coverage WSN is improved using Improved Harmony Search algorithm. This is done by defining a good objective function that tries to improve coverage in each iteration of the algorithm and also distribute nodes uniformly around the Sink to balance the traffic load between them. Also best position for Sink is found in the center of nodes deployment as a place where longest path to all other nodes is smallest than any other place. Although it is attempted to balance traffic load between nodes, still some nodes such as ones which are closer to the Sink bear more traffic than those which are farther, because they should relay others information to the Sink as well. Therefore, true load balancing requires that high energy consuming nodes be replaced with less energy consuming ones. Some of these nodes are selected as mobile ones. Simulations show that best improvement in lifetime is achieved when task of moving nodes is done just once; in the middle of primary calculated lifetime. The algorithm tries to find the sensor nodes that have low energy and will be dead earlier and relocate each of them with a mobile node with high energy while the distance between every two selected nodes is low.

Through computer simulations, it has been confirmed that the proposed method which is based on Improved Harmony Search algorithm and uses both static and mobile sensors found a distribution with better coverage compared to GA and HS based solutions. Furthermore, by balancing energy consumption among sensor nodes and proper selection of mobile nodes and replacing high remained energy nodes with low ones after a while, lifetime is increased two times and in some cases, for example when number of nodes is 100, up to 20 times.

References

[1]. Kamimura, J., Wakamiya, N., Murata, M. Energy-Efficient Clustering Method for Data Gathering in Sensor Networks, in *Proc. of the Annual International Conference on Broadband Networks*, 2004.

[2]. Akyildiz, I. F., Sankarasubramaniam, Y., Su, W., Cayirci, E. Wireless sensor networks: A survey, *Computer Networks*, Vol. 38, Issue 4, March 2002, pp. 393-422.

[3]. Culler, D., Estrin, D., Srivastava, M. Overview of sensor networks, *Computer*, Vol. 37, Issue 4, 2004, pp. 41-49.
[4]. Huang, C. F., Tseng, Y. C. A survey of solutions to the coverage problems in wireless sensor networks, *Journal of Internet Technology*, Vol. 6, No, 1, 2005, pp. 1-8.
[5]. Dhillon, S. S., Chakrabarty, K. Sensor placement for effective coverage and surveillance in distributed sensor networks, *IEEE Wireless Communications and Networking*, New Orleans, LA, USA, Vol. 3, 2003, pp. 1609-1614.
[6]. Chang, J. H., Leandros, T., Maximum System Lifetime Routing in Ad-Hoc Networks, A Critical Study. *IEEE/ACM Transactions on Networking*, 2004, Vol. 12, Issue 4.
[7]. Dong, Q. Maximizing system lifetime in wireless sensor networks, in *Proceedings of the 4t^h International Symposium on Information Processing in Sensor Networks (IPSN '05)*, 2005, NJ., USA.
[8]. Schumacher, A., Pekka, A., Thorn, T., Haanp, H. Lifetime Maximization in Wireless Sensor Networks by Distributed Binary Search, in *Proceedings of the 5th European Conference on Wireless Sensor Networks (EWSN'08)*, 2008.
[9]. Park, P., Sahni, S. Maximum Lifetime Routing In Wireless Sensor Networks, *IEEE Trans. Computers*, 54, 9, 2005, pp. 1081-1090.
[10]. Kang, I., Poovendran, R. Maximizing Static Network Lifetime of Wireless Broadcast Adhoc Networks, *Mobile Networks and Applications Journal*, Vol. 10, Issue 6, 2005.
[11]. Das, A. K., El-Sharkawi, M., Marks, R. J., Arabshahi, P., Gray, A., Maximization of Time-to-first-failure for Multicasting in Wireless Networks: Optimal Solution, in *Proceedings of the IEEE Milcom*, Vol. 3, 2004, pp. 1358-1363.
[12]. Zhou, Z., Das, S., Gupta, H. Connected k-coverage problem in sensor networks, in *Proceedings of the International Conference on Computer Communications and Networks*, Chicago, IL, 2004, pp. 373 - 378.
[13]. Geem, Z. W., Kim, J. H., Loganathan, G. V. A New Heuristic Optimization Algorithm: Harmony Search, *Simulation*, Vol. 76, No. 2, 2001, pp. 60-68.
[14]. Mahdavi, M., Fesanghary, M., Damangir, E. An improved harmony search algorithm for solving optimization problems, *Applied Mathematics and Computation*, Vol. 188, 2007, pp. 1567–1579.
[15]. Ebrahim Nezhad, Sh., Jalal Kamali, H, Ebrahimi Moghaddam, M. Solving k-coverage problem in Wireless Sensor Network Using Improved Harmony Search, in *Proc. of the IEEE, International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA)*, Vol. 3, 2010, pp. 49 - 55.
[16]. Huang, C., Tseng, Y. The coverage problem in a wireless sensor network, in *Proc. of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA '03)*, 2003, pp. 115–121.
[17]. Megerian, S., Koushanfar, F., Potkonjak, M., Srivastava, M. B. Worst and best-case coverage in sensor networks, *IEEE Transactions on Mobile Computing*, Vol. 4, Issue 1, 2005, pp. 84-92.
[18]. Kalaycı, T. E., Yıldırım, K. S., U˘gur, A. Maximizing Coverage in a Connected and K-Covered Wireless Sensor Network Using Genetic Algorithms, *International Journal of Applied Mathematics and Informatics*, Issue 3, Vol. 1, 2007, pp.123-130.
[19]. Olston, C., Jiang, J., Widom, J. Adaptive filters for continuous queries over distributed data streams, in *Proceedings of the ACM SIGMOD International Conference on Management of Data (ACM SIGMOD'03)*, June 2003, pp. 563–574.
[20]. Luo, D., Zuo, D., Yang, X. An Optimal Sink Selection Scheme for Multi-sink Wireless Sensor Networks, in *Proc. of the International Conference on Computer Science and Information Technology,* 2008, pp. 544-548.
[21]. Tang, X., Xu, J. Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks, in *Proc. of the International Conference on Computer Communications. Proceedings (INFOCOM' 2006)*, Barcelona, Spain, 2006, pp. 1–12.
[22]. Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., Gill, C. Integrated coverage and connectivity configuration in wireless sensor networks, *in Proc. of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, 2003, pp. 28-39.
[23]. Das, S., Mei, Y., Xian, C., Charlie Hu, Y., Lu, Y. Repairing Sensor Network Using Mobile Robots, in *Proc. of the IEEE Workshop on Wireless Ad hoc and Sensor Networks (WWASN' 2006)*, 2006.
[24]. Wang, G., Cao, G., La Porta, T., Zhang, W. Sensor Relocation in Mobile Sensor Networks, in *Proc. of the IEEE Annual Joint Conference on Computer and Communications Societies (INFOCOM 2005)*, Vol. 4, 2005, pp. 2302–2312.
[25]. Wang, X., Ma, J., Wang, S., Bi, D. Distributed Particle Swarm Optimization and Simulated Annealing for Energy-efficient Coverage in Wireless Sensor Networks, *Sensors*, Vol. 7, 2007, pp. 628-648.
[26]. Gao, S., Vu, C. T., Li, Y. Sensor Scheduling for k-Coverage in Wireless Sensor Networks, in *Proc. of the*

2nd International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2006), Hong Kong, China, 2006.

[27]. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., Ito, M. Extending k-Coverage Lifetime of Wireless Sensor Networks Using Mobile Sensor Nodes, in *Proc. of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Marrakech, Morocco, 2009, pp. 48-54.

[28]. Lee, K. S., Geem, Z. W. A new meta-heuristic algorithm for continues engineering optimization: harmony search theory and practice, *Computer Method in Applied Mechanics and Engineering*, Vol. 194, Issue 36 38, 2004, pp. 3902–3933.

[29]. Sudhakar, T. D. Supply restoration in distribution networks using Dijkstra's algorithm, in *Proc. of the IEEE International Conference on Power System Technology*, Vol. 1, 2004, pp. 640-645.

[30]. Heinzelman, W. R., Chandrakasan, A., Balakrishnan, H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks, in *Proc. of the 33rd IEEE Hawaii Int'l. Conf. on System Sciences (HICSS 2000)*, Washington, DC, USA, Vol. 8, 2000, pp. 3005-3014.

[31]. Jun, J.H,. Xie, B., Agrawal, D. P. Wireless Mobile Sensor Networks: Protocols and Mobility Strategies, in *Guide to Wireless Sensor Networks, Computer Communications and Networks*, 2009, pp. 607-634.
