Lifetime Enhancement of Dynamic Heterogeneous Wireless Sensor Networks with Energy-Harvesting Sensors

Chun-Cheng Lin · Yung-Chiao Chen · Jiann-Liang Chen · Der-Jiunn Deng* · Shang-Bin Wang · Shun-Yu Jhong

Abstract Lifetime enhancement has been the major constraint of developing wireless sensor networks (WSNs). Most of previous related works separately considered dynamics and heterogeneity of WSNs, and did not consider energy-harvesting (EH) sensors, which can absorb natural power (e.g., solar and wind power) to extend lifetime of sensor devices. Therefore, this work investigates the problem of extending the lifetime of dynamic heterogeneous WSNs with EH sensors to enhancing the total WSN lifetime. This problem can be characterized as finding the maximal number of covers each of which is a part of all sensors so that all targets can be monitored by these sensors. Since the case for static WSNs has been shown to be NP-complete, the concerned problem is also NP-complete. Hence, this work first models this problem mathematically, and then proposes a novel harmony search algorithm with multiple populations and local search (HSAML) for this problem with dynamics, heterogeneity, and EH sensors. By simulation, the network lifetime, stability, and executing time of the proposed algorithm are analyzed. From

---

C.-C. Lin · S.-B. Wang · S.-Y. Jhong
Department of Industrial Engineering and Management,
National Chiao Tung University, Hsinchu 300, Taiwan
e-mails: cclin321@nctu.edu.tw, s31305911@hotmail.com

Y.-C. Chen · J.-L. Chen
Department of Electrical Engineering
National Taiwan University of Science and Technology, Taipei 106, Taiwan
e-mails: Lchen@mail.ntust.edu.tw

D.-J. Deng
Department of Computer Science and Information Engineering,
National Changhua University of Education, Changhua 500, Taiwan
e-mail: djdeng@cc.ncue.edu.tw

* D.-J. Deng is the corresponding author of this paper (E-mail: djdeng@cc.ncue.edu.tw).

experimental results, the proposed HSAML performs better than the conventional algorithm in terms of average network lifetime for larger-scale problems (i.e., when the number of common and EH sensors is small). In addition, the results confirm that adding EH sensors really helps extend the total WSN lifetime.

Keywords Heterogeneous wireless sensor network, energy-harvesting sensor, harmony search algorithm, dynamic optimization

1. Introduction

Wireless sensor network (WSN) consists of a large number of sensors, in which the sensing function of sensor devices is used to detect targets and environments, e.g., weather temperature, and human heartbeat pulse; the transmitting function of sensors is used in wireless communications among sensors, so that the collected messages are transmitted to base stations or monitoring centers, which can analyze the current condition and further provide responses. With WSNs, human daily life is promoted to a more convenient level. WSN has advantage of small size and low price, and has lots of wide applications, including environmental monitoring [1], [2], and healthcare monitoring [3], [4].

However, WSN still has some drawbacks required to be improved. Since most WSN applications are deployed in mountains or wilderness, it would not be convenient to replace batteries to extend lifetime of sensors. Hence, it has been challenging to extend lifetime of WSNs under limited device power. From the related literature [5], [6], some works divided all sensors into multiple groups (each of which is called a *cover*) so that each cover includes a minimal number of sensors to monitor all targets. Then, consider one of those covers. Sensors in this cover are switched on to monitor all targets until some target cannot be monitored, while the other sensors are switched off or use out their own power. Then, repeat the same procedure until all covers have been considered. Hence, lifetime of this WSN may be extended if more covers can be found.

On the previous approaches, the work in [7] has shown the problem of finding the maximal number of covers to be reduced to the set k-cover problem, which is NP-complete [8], and hence, the work in [7] proposed a polynomial-time heuristic algorithm to resolve the problem. However, the probability of find optimal solutions using their proposed algorithm is low, and it is not guaranteed to find optimal solutions. Hence, the work in [9] proposed a genetic algorithm (GA) for the problem, in which dynamic WSNs are

additionally considered.

On the previous problem settings, most previous works focused on the frameworks of homogeneous and static WSNs [7], [10], [11]. In homogeneous WSNs, all sensors are of equal sensing range and transmission range. However, with advance in technologies, practical WSNs are heterogeneous, e.g., the work in [12] showed that adjustment of sensing range and transmission range of sensors can effectively extend the WSN lifetime. On the other hand, static WSNs is not practical. The work in [9] considered a dynamic WSN framework in which each sensor may change its own state to *active* or *asleep* due to some emergency or power saving, e.g., early damage of this sensor due to violent environmental change, and going asleep for saving power. The work in [13], [14] proposed a sleep scheduling scheme to control states of sensors dynamically, to extend the total WSN lifetime.

Most of previous works focused on effectively making use of remaining power of sensors to extend the WSN lifetime. Recent works have tended to investigate how to increase the remaining power of sensors. One of those novel techniques is energy-harvesting (EH) sensors, which can transform the natural power in environments (e.g., solar power and wind power) into electricity power stored in sensor batteries for future use. As long as the environment has no much violent change (e.g., torrential rains), an infinite WSN lifetime could be achieved [15]. However, it is impossible to replace all sensors by EH sensors using the current technology [16]. Hence, the most appropriate approach is to apply a hybrid WSN framework consisting of common sensors and EH sensors. By doing so, the installation cost can be reduced; and the total WSN lifetime can be extended.

In light of the above, to meet real situations, this work considers the lifetime extension problem in the WSN framework with dynamics, heterogeneity, and EH sensors. Dynamics means that each common or EH sensor has four states: active, asleep, malfunctioned, and dead; and heterogeneity means that the WSN framework considers the mixed optimization of common and EH sensors, in which the coverage of each sensor is different. Since the lifetime extension problem for static WSNs has been shown to be NP-complete, there is no deterministically polynomial-time algorithm for the problem. Hence, it is suitable to apply metaheuristic algorithms to resolve this problem. However, there is no metaheuristic algorithm that dominates all metaheuristic algorithms [17]. Hence, it is of interest to investigate an improved metaheuristic algorithm for the concerned problem. Since the harmony search algorithm (HSA) has been

shown to perform better than conventional metaheuristic algorithms, and the multi-population metaheuristic algorithm is a popular approach for dynamic problems, this work proposes a harmony search algorithm with multiple populations and local search (HSAML) for the problem.

The main contributions of this paper are as follows:

- Considering dynamic heterogeneous WSNs to meet more practice: This work considers a dynamic heterogeneous WSN with EH sensors, in which each common or EH sensor has four states: active, asleep, malfunctioned, and dead. The advantages of the heterogeneity include saving costs and extending the WSN lifetime. Note that there have been a lot of previous works that regards the extension of conventional networks with EH sensors as a main contribution, e.g., [5], [6], [16].
- Proposing an approach suitable for the concerned problem: This work extends the approach proposed in [18] with multiple populations and local search to solve the concerned problem, in which the solution representation includes encoding EH sensors.

The rest of this work is organized as follows. Section 2 gives literature review. Section 3 gives the system framework. Section 4 proposes the proposed HSAML, and explains its difference from previous algorithms. Section 5 gives experimental results and some discussions. Section 6 concludes this work.

2. Literature review

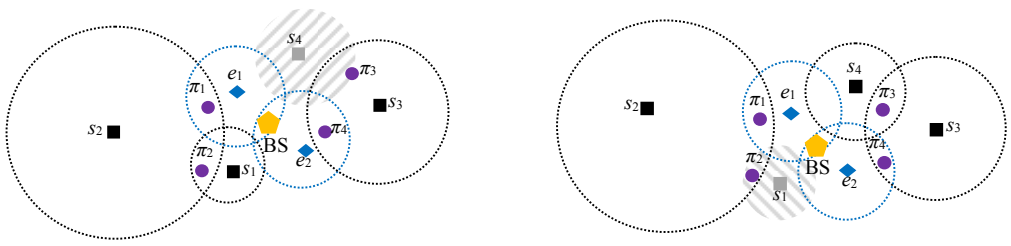
The lifetime extension problem of static WSNs can be reduced to the set k-cover problem [7], which is NP-complete. Hence, the concerned dynamic problem is NP-complete. Most of previous related works considered that if more covers that satisfy the required conditions can be found, then each cover of sensors are active in turn while other covers of sensors fall asleep, so that the total WSN lifetime can be extended. Such a type of problems is called the sleep scheduling problem for WSNs, and has been discussed widely. However, the problem is NP-complete, and hence, designing metaheuristic algorithms for the problem received much attention. The work [10] proposed a metaheuristic algorithm for the problem, but it took much computing time. The work in [9], [11] further applied a genetic algorithm (GA) for the problem, and showed that GA performs better in the same benchmark problems. The problem in [19] applied a multi-population HSA for dynamic underwater acoustic sensor networks.

Another main feature of this work is to include EH sensors. The power sources of EH sensors include

solar power [20], wind power and hydroelectric power [21]. Those natural powers can help increase the remaining battery power of sensor devices. As long as the environment does not have violent change, EH sensors can achieve almost infinite lifetime [15], [16], and further extend the whole WSN lifetime. For example, the work in [22] proposed a power management algorithm to effectively control the power consumption problem in EH sensor networks. The work in [23] considered the problem of harvesting-based wireless sensor devices with battery degradation, created a Markov chain model for the problem, and applied a linear programming method to solve this problem. The work in [16] applied a clustering algorithm to resolve the lifetime extension problem for WSNs with EH sensors.

3. System Framework

This work investigates a dynamic heterogeneous WSN with EH sensors. Through appropriately controlling the states of common and EH sensors, the lifetime of the WSN can be extended. Consider a dynamic and heterogeneous WSN with EH sensors consisting of n common sensors s_1, ..., s_n and m EH sensors e_1, …, e_m to monitor p targets π_1, ..., π_p. Note that the so-called "common" sensors mean those without EH sensors. The communications are centralized by a base station (BS), which can collect messages from all sensors. An instance when $n = 4$, $m = 2$, and $p = 4$ is illustrated in Fig. 1(a). Consider a time axis with multiple *key times*, at each of which the environment to be monitored by a WSN is changed, e.g., change exists between Figs. 1(a) and 1(b). Suppose that positions of all sensors are fixed over the whole time horizon. The work in [24] showed that when $R_C^i \geq 2R_S^i$ (where R_C^i and R_S^i are the transmission range and the sensing range of the i-th sensor, respectively), sensor s_i can not only detect (monitor) targets, but also transmit messages to the BS. Hence, this work supposes that $R_C^i \geq 2R_S^i$.



(a) At the τ-th key time

(b) At the (τ + 1)-th key time

Fig. 1. Illustration of a dynamic heterogeneous WSN with EH sensors.

Dynamics of the concerned WSN is explained as follows. Each common or EH sensor has four states: *active*, *asleep*, *malfunctioned*, and *dead*. A sensor can monitor targets only when the sensor is active. A sensor that is not active would go *asleep* to avoid power consumption. Suppose that only an active sensor consumes its battery power; while an asleep sensor consumes very little power, which can be neglected. Hence, when an active sensor consumes all its own battery power after a period of time, or the power collected by an EH sensor is not enough to compensate its power consumption, the sensor cannot be used to monitor targets, and hence is *dead*. Additionally, a sensor may be malfunctioned temporarily due to some unpredictable reasons, but could be recovered via repairing of the sensor processor and is able to be set to be active or asleep again. Note that an active or asleep sensor is also said to survive, i.e., it can monitor targets now; while a malfunctioned or dead sensor is also said to fail, i.e., it cannot monitor targets now. To simplify the problem, it is supposed that each EH sensor is not dead, i.e., each EH sensor only has three states: *active*, *asleep*, and *malfunctioned*.

At different times, each sensor could have different modes, so that active sensors at different times are different. Therefore, *key times* are used to resolve this problem as follows. The first key time is the initial time (see Fig. 2), at which information of all sensors are collected. With the information, we find a *cover*, which is a subset of all active sensors whose coverages cover all targets. Then, we apply a *sleep schedule* in which sensors in this cover are switched to be active; while others fall asleep. This sleep schedule keeps being applied until some target cannot be covered due to some failed sensor. Then, we enter the 2nd key time. Similarly, information of all sensors is updated at the 2nd key time; and the 2nd sleep schedule is found to be applied until some target cannot be covered. Similarly, the other key times can be defined. And, we decide a sleep schedule at each key time until we cannot find any cover to cover all targets.

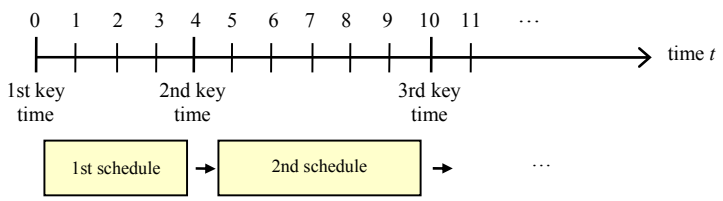


Fig. 2. Illustration of relationship of schedules and key times.

In what follows, the mathematical model for the problem concerned in this work is created by extending the model in [9] with dynamics, heterogeneity, and EH sensors. Consider a WSN with n common sensors $s_1, s_2, \ldots, s_n$ and m EH sensors $e_1, e_2, \ldots, e_m$ to monitor p targets $\pi_1, \pi_2, \ldots, \pi_p$. Since finding the maximal number of covers is supposed to imply a high possibility of extending the WSN lifetime in most of the previous works, the objective of the concerned problem is to find a maximal number K_τ of covers at the τ-th key time as follows:

$$\text{Maximize } K_\tau \tag{1}$$

Before introducing constraints of the concerned problem, the notations used in the problem are listed in Table 1. With those notation, set of survival sensors at the τ-th key time (i.e., A_τ) is computed as follows:

$$A_\tau = (S_{\tau\text{-}1} \cup E_{\tau\text{-}1} \cup R_\tau) \setminus (D_\tau \cup F_\tau) \tag{2}$$

Table 1. Notation used in the problem.

Notation	Content
S	Set of common sensors, $S = \{s_1, s_2, \ldots, s_n\}$.
E	Set of EH sensors, $E = \{e_1, e_2, \ldots, e_m\}$.
T	Set of targets, $T = \{\pi_1, \pi_2, \ldots, \pi_p\}$
S_τ	Set of survival common sensors at the τ-th key time.
E_τ	Set of survival EH sensors at the τ-th key time.
A_τ	Set of all survival sensors at the τ-th key time.
F_τ	Set of malfunctioned sensors at the τ-th key time.
D_τ	Set of dead sensors at the τ-th key time.
R_τ	Set of recovered sensors at the τ-th key time.
P_f	Probability that a sensor is malfunctioned.
P_r	Probability that a sensor is recovered.
C_τ	Set of covers at the τ-th key time, $C_\tau = \{C_1^\tau, C_2^\tau, \ldots, C_{K_\tau}^\tau\}$.

A_τ is the set of all survival sensors at the τ-th key time, which is the union of the set of survival common sensors at the $(\tau - 1)$-th key time ($S_{\tau\text{-}1}$), the set of survival EH sensors at the $(\tau - 1)$-th key time ($E_{\tau\text{-}1}$), and the set of recovered sensors at the τ-th key time (R_τ) excluding the set of dead sensors at the τ-th key time (D_τ) and the set of malfunctioned sensors at the τ-th key time (F_τ).

Define C_τ as a subset of A_τ whose sensing coverages cover all targets in T as follows:

$$\pi_j \in C_\tau \subseteq A_\tau, \forall\ \pi_j \in T \tag{3}$$

Then, the above definition implies the following three constraints of the concerned problem at the τ-th key time:

$$C_1^\tau, C_2^\tau, ..., C_{K_\tau}^\tau \subseteq A_\tau \tag{4}$$

$$C_i^\tau \bigcap C_j^\tau = \varnothing, \forall i, j \in \{1, 2, \ldots, K_\tau\}, i \neq j \tag{5}$$

$$\pi_j \in C_i^\tau, \forall \pi_j \in T, \forall i \in \{1, 2, \ldots, K_\tau\} \tag{6}$$

Constraints (4) and (5) enforce that each pair of covers are two disjoint subsets of A_τ; or, one or both are empty. Constraints (6) enforces that all targets are covered by each cover. The maximal number K_τ of covers at the τ-th key time can be obtained under Constraints (4) – (6).

Take Fig. 1(a) as an example. Set of EH sensors is $E = \{e_1, e_2\}$, which are never dead in our problem setting. At the τ-th key time, set of survival sensors is $S_\tau = \{s_1, s_2, s_3\}$. Hence, set of survival sensors is $A_\tau = S_\tau \cup E = \{s_1, s_2, s_3, e_1, e_2\}$; set of malfunctioned sensors $F_\tau = \{s_4\}$. Under this configuration, we can find a cover $C_1^\tau = \{e_1, s_1, s_3\}$ that monitor all targets. At the (τ + 1)-th key time (see Fig. 1(b)), consider that common sensor s_1 is dead (i.e., $D_\tau = \{s_1\}$); and common sensor s_4 is recovered (i.e., $R_{\tau+1} = \{s_4\}$). Set of survival sensors at the (τ + 1)-th key time is $A_{\tau+1} = S_{\tau+1} \cup E = \{s_2, s_3, s_4, e_1, e_2\}$. Under this configuration, we can find a cover $C_1^{\tau+1} = \{s_2, s_4, e_2\}$. Note that Fig. 1 is a small-scale WSN, so only one cover is found. In other instances, more covers may be found.

After finding all covers $C_1^\tau, C_2^\tau, ..., C_{K_\tau}^\tau$ at the τ-th key time, a random cover of them is selected. Then, the WSN applies the schedule at the τ-th key time in which sensors in the selected cover are active, while other sensors go asleep. Once when some target cannot be monitored by the selected cover due to some failed sensor, the WSN enters the next (τ+1)-th key time, finds the maximal number of covers at the (τ+1)-th key time, and uses one of those covers to decide the sleep schedule at the (τ+1)-th key time. Repeat the

above procedure until no cover can be found.

Compared to our approach that randomly selects a cover of sensors to be active, the optimal cover of sensors selected to be active could be found to extend the total WSN lifetime in the long run. However, it is hard and time-consuming to find the optimal cover of sensors because it involves the cover selection and the states of all sensors in many later key times. Therefore, it would be of interest to investigate how to find the optimal cover in the future.

Note that, inheriting from the NP-completeness of finding the maximal number of covers, the concerned problem is also NP-complete. To resolve the concerned problem, this work proposes a metaheuristic algorithm in the rest of this work.

4. The Proposed Harmony Search Algorithm with Multiple Populations and Local Search

Harmony search algorithm (HSA) [25] is a metaheuristic algorithm by simulating that a number of musicians cooperatively play a harmony impromptu. First, HSA includes a solution encoding scheme, in which each candidate solution (called *harmony*) for the concerned problem is encoded as a string of decision variables (called *notes*) for the problem. And, HSA defines a *fitness* function to evaluate performance of a harmony, which is related to the objective function of the concerned problem. Then, HSA has a repetitive loop to improve a harmony memory (*HM*) matrix that stores a number of harmonies, which keeps *hms* best harmonies found so far. Each iteration of this loop improves the *HM* matrix with a probability *HMCR* to select a historical note from *HM* matrix, and a probability *PAR* to adjust a note within a range [-*bw*, *bw*]. The loop ends until the maximal number of iteration (*NI*) is achieved.

In general, the most popular metaheuristic algorithm is genetic algorithm (GA). The work in [11] showed that for the set k-cover problems, HSA performs better than GA. Additionally, the work in [25] showed that HSA generally performs better than GA in lots of benchmark datasets. Hence, this work proposes an improved version of HSA for the concerned problem. As the multi-population version of HSA is a general solution for dynamic problems, this work improves our previous algorithm in [19] to propose a novel HSA with multiple populations and local search (HSAML). Note that the difference of the proposed HSAML from [19] is to include a novel local search scheme.

The HSAML for the τ-th key time is given in Algorithm 1, and the main steps of the HSAML are listed

as follows:

Step 1. Initialization: Since survival sensors may be different at different key times, the first step of HSAML is to update information of survival sensors at the τ-th key time. Then, according to the information, generate and evaluate *hms* random harmonies, which are stored in the *HM* matrix. Additionally, the iteration number η is initialized to 1.

Step 2. Multiple populations: HSAML considers multiple populations, i.e., it divides the whole population (*HM* matrix) into δ sub-populations of equal sizes (denoted by $sub\text{-}HM_1$, $sub\text{-}HM_2$, …, $sub\text{-}HM_\delta$).

Step 3. Adjustment of each $sub\text{-}HM_i$: Consider each $sub\text{-}HM_i$. Execute the following two solution operators:

1) Global search: Check if a random number from [0, 1] is less than *HMCR*. If true, we apply two times of tournament selection operator to select two harmonies x^{new1} and x^{new2} from $sub\text{-}HM_i$. Then, we conduct a uniform crossover operator on x^{new1} and x^{new2}, and replace x^{new1} and x^{new2} by the resultant two harmonies of the crossover operator.

2) Local search: When the above step holds (i.e., harmonies x^{new1} and x^{new2} are generated), this local search operator is executed as follows. Check if a random number from [0, 1] is less than $PAR(\eta)$, which depends on iteration number η. If true, then we improve the worse one of x^{new1} and x^{new2} by a local search operator which will be detailed later, in which parameter $MT(\eta)$ is applied.

Step 4. Updating $sub\text{-}HM_i$: Let x^{new} be the one of x^{new1} and x^{new2} with better fitness. If x^{new} has better fitness than the worst harmony in $sub\text{-}HM_i$, x^{new} replaces it.

Step 5. Loop check: Increase the iteration number. If the maximal iteration number *NI* is achieved, go back to Step 3.

Step 6. Solution decoding: The best harmony among all $sub\text{-}HM_i$'s is decoded as the final solution (i.e., number and content of covers). If number of covers is nonzero, then one of the covers is selected as the output; otherwise, "No solution" is outputted.

Algorithm 1 HSAML (the τ-th key time)

```
Update set of survival sensors at the τ-th key time
Initialize the parent harmony memory HM and other parameters
Divide HM into sub-HM_1, sub-HM_2, …, sub-HM_δ
Initialize the current iteration number η = 1
while η ≤ the maximal iteration number NI do
    for each sub-HM_i do
        if rand(0, 1) < HMCR then
            Use two times of tournament selection to choose two harmonies x^new1 and
            x^new2 from sub-HM_i, conduct a uniform crossover operator on x^new1 and
            x^new2, and replace x^new1 and x^new2 by the resultant two harmonies of the
            crossover operator
            if rand(0, 1) < PAR(η) then
                Choose the one of x^new1 and x^new2 with worse fitness value, according
                MT(η) on x_i^new and x_j^new for local search
            else
                Let x^new be the one of x^new1 and x^new2 with better fitness value
            end if
        else
            Randomly generate a feasible harmony as x^new
        end if
        If x^new is better than the worst harmony in sub-HM_i, x^new replaces it
    end for
    η = η + 1
end while
Decode the best harmony among all sub-HM_i's as a solution of the concerned problem, i.e., number
of covers
If number of covers is nonzero, randomly choose one of the covers as the output; otherwise, output
"No solution after the τ-th key time"
```

4.1 Solution encoding and fitness evaluation

On solution encoding, the set A_τ of all survival sensors (including common and EH sensors) at the τ-th key time is collected first. According to the information, if a target falls within the sensing coverage of a sensor, a link between the target and the sensor is created. By doing so, we create a bipartite graph $G_\tau = (A_\tau, T, L)$ between sensors and targets, in which A_τ and T are the sets of nodes respectively representing survival sensors and targets; L is the set of links between A_τ and T. For example, Fig. 3 is the bipartite graph corresponding to the WSN configuration in Fig. 1(a). Hence, a solution (a cover of sensors) is a matching of the bipartite graph.

Notations of the sensors in A_τ are changed as $s_{1'}, s_{2'}, \ldots, s_{|A_\tau|'}$. Then, with the bipartite graph created above, a harmony in the proposed algorithm is encoded as a permutation of IDs of all survival sensors in A_τ, denoted by $\langle p_1, p_2, \ldots, p_{|A_\tau|} \rangle$, in which each $p_i \in \{s_{1'}, s_{2'}, \ldots, s_{|A_\tau|'}\}$ and $p_i \neq p_j$ for $i \neq j$.

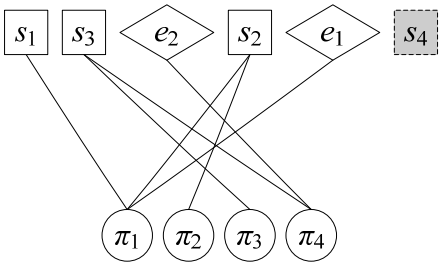


Fig. 3. The bipartite graph between survival sensors and targets corresponding to Fig. 1(a).

On fitness evaluation (solution encoding), the algorithm of evaluating the fitness value of a harmony $\langle p_1, p_2, \ldots, p_{|A_\tau|} \rangle$ is detailed as follows:

Step 1. Initialize parameters. Let parameter k record the number of covers; and C_k^τ record set of sensors in the k-th cover at the τ-th key time. Let $k = 1$ and $C_k^\tau = \varnothing$.

Step 2. For each $i = 1, \ldots, |A_\tau|$, let $C_k^\tau = C_k^\tau \cup \{p_i\}$, and if C_k^τ is a cover according to bipartite graph G_τ, then increase k by one.

Step 3. If C_k^τ is a cover, then output $k \cdot |T|$; otherwise, Output $(k - 1) \cdot |T|$ + number of the targets monitored by sensors in C_k^τ.

For example, consider the bipartite graph in Fig. 3. Fitness of the corresponding harmony is 1 + 2 + 0 + 1 + 1 = 5, which is contributed by s_1, s_3, e_2, s_2, and e_1 sequentially. The time complexity of the algorithm is analyzed as follows.

Lemma 1. The fitness of a harmony can be computed in $O(n + m)$ time.

Proof. In the above algorithm, Steps 1 and 3 are done in $O(1)$ time. Step 2 counts all values of iteration i, and hence can be done in $O(|A_\tau|) = O(n + m)$ time.

□

4.2 Dynamic parameter scheme

The classical HSA applies fixed values for parameters *HMCR* and *PAR*. The work in [18] discovered that the number of iterations of HSA affects the solution search ability of the algorithm, if the *PAR* value increases and the *bw* value decreases as iteration number increases. Hence, this work applies the design for

PAR parameter in [18] as follows:

$$PAR(\eta) = par_{\min} + \frac{par_{\max} - par_{\min}}{NI} \cdot \eta \tag{7}$$

where $par_{\min}$ and $par_{\max}$ are the minimal and maximal values, respectively; *NI* is the maximal iteration number; and η is the current iteration number.

This work considers a parameter *MT* (mutation times) in local search, in which the MT value decreases as iteration number increases, because the mutation change at latter stage is supposed to be smaller. The *MT* value is computed as follows:

$$MT(\eta) = \left\lceil | A_\tau | \cdot mr_{\max} (\frac{mr_{\min}}{mr_{\max}})^{\frac{\eta}{NI}} \right\rceil \tag{8}$$

where $mr_{\min}$ and $mr_{\max}$ are the minimal and maximal ratios, respectively.

4.3 Adjusting harmonies

Different from the classical HSA designed for solutions with continuous values, the proposed HSAML is designed for solutions with discrete values and hence requires a further design. Hence, this section gives details on two operators of adjusting harmonies.

1) Global search by tournament selection and uniform crossover

Tournament selection is a basic operator in genetic algorithm. The work in [26], [27] showed that HSA with tournament selection can search better solutions. The operator is to consider two harmonies from the *HM* matrix to compete. The one of the two harmonies with better fitness is the resultant harmonies of this operator. By two times of tournament selection, two harmonies x^{new1} and x^{new2} are selected.

Then, a uniform crossover on the two harmonies x^{new1} and x^{new2} is conducted. The work in [28] showed that uniform crossover is helpful in searching solutions. The uniform crossover used in this work is detailed as follows. First, randomly generate a mask, which is a binary string of length equal to the harmony length. Without loss of generality, consider harmony x^{new1}; the case for x^{new2} can be done similarly. Based on the information of this mask from the left to the right, this operator generates a duplicate harmony $x^{new1'}$ where the notes at positions of 0's of the mask are copied from x^{new1}. The notes at positions of 1's of the mask are

determined as follows. Generate a duplicate harmony $x^{new2'}$ whose content is the same with x^{new2}. Remove the notes in $x^{new2'}$ that have appeared in $x^{new1'}$. Copy the remaining notes in $x^{new2'}$ in the same order to fill out the notes in $x^{new1'}$ at positions of 1's of the mask.

The time complexity of the global search is analyzed as follows.

Lemma 2. If the number of the harmonies concerned in the global search is denoted by ρ, then the global search can be computed in $O(n + m)$ time.

Proof. The tournament selection operation randomly selects two of the ρ harmonies in $O(1)$ time, and remains the better one of the two selected harmonies in $O(1)$ time. Hence, two harmonies x^{new1} and x^{new2} are selected in $O(1)$ time.

Next, the uniform crossover operation generates a random mask of length $n + m$ in $O(n + m)$ time. With the help of duplicate harmonies $x^{new1'}$ and $x^{new2'}$, the crossover operation can be done in $O(n + m)$ time.

□

2) Local search

After the global search above, some cover sets decoded from a harmony may include no EH sensors. But, EH sensors can increase the probability of extending the lifetime of a cover. Hence, this issue can be addressed by the proposed local search operator. First, this local search operator selects and improves the worse one of the two harmonies x^{new1} and x^{new2} after uniform crossover. Then, this operator conducts $MT(\eta)$ times of mutations on this harmony, i.e., swapping notes of two random positions. Since $MT(\eta)$ is just a parameter and all operations in local search can be done in $O(1)$ time, it is obvious to have the following analysis on time complexity.

Lemma 3. The local search above can be computed in $O(1)$ time.

4.4 Algorithm complexity analysis

The complexity of the proposed HSAML is analyzed as follows.

Theorem 3. The proposed HSAML can be computed in $O(NI \cdot (hms + \delta \cdot (n + m)))$ time.

Proof. By Lemma 1 that computes the fitness of each harmony in $O(n + m)$ time, Step 1 of the proposed HSAML randomly initializes the *HM* matrix of size $hms \times (n + m)$ in $O(hms \cdot (n + m))$ time. Step 2 divides the *HM* matrix into δ *sub-HM* matrices in $O(\delta)$ time by designing the data structure for *sub-HM* matrices.

By Lemmas 2 and 3, the global search and the local search can be done in $O(n + m)$ and $O(1)$ time, respectively. Step 3 executes δ times of the global search and the local search, and hence can be done in $O(\delta \cdot (n + m))$ time. In Step 4, the worst harmony of each *sub-HM* can be found in $O(hms / \delta)$ time; it is replaced by x^{new} in $O(m + n)$ time; and δ times of these replacements can be done in $O(\delta(hms / \delta + m + n)) = O(hms + \delta(m + n))$ time.

Step 5 considers NI iterations of Steps 3 and 4. Hence, Step 5 can be done in $O(NI \cdot (\delta \cdot (n + m) + hms + \delta \cdot (m + n)) = O(NI \cdot (hms + \delta \cdot (n + m)))$ time. Step 6 can be done in $O(hms + (m + n))$ time.

□

Since NI, hms, and δ are parameters, the time complexity of the proposed HSAML corresponding to the input size is $O(n + m)$ time, which is linear in terms of the input size.

5. Experimental Results

This section gives implementation of the proposed HSAML and establishes a simulation environment to evaluate the performance of the proposed HSAML and the conventional HSA. First, a simulation for a dynamic WSN framework is established. Next, the experimental setting of the algorithm and simulation are introduced. Next, the experimental analyses under difference scenarios are given.

The flowchart of simulation on the HSAML is given in Fig. 4, and the main steps of the simulation are given as follows:

Step 1. Initialize the time t to 0, and let 0 be the 1-th key time. And, let τ record the serial number of the current key time, i.e., $\tau = 1$.

Step 2. Collect information of survival sensors (A_τ).

Step 3. Execute the HSAML algorithm at the τ-th key time detailed in the last section. If the output is "No solution", this simulation is terminated and output t as the lifetime of this WSN; otherwise (i.e., there is at least one cover to be found), execute the next step.

Step 4. Select one of the found covers. The sensors in this sensors are switched on (i.e., they consume power); while other sensors are switched off.

Step 5. Each active sensor consumes one unit of power. And increase one unit of time, i.e., $t = t + 1$.

Step 6. Simulate dynamic change of sensor states, in which each sensor has a probability P_f to fail and a probability P_r to be recovered.

Step 7. Collect information of survival sensors (A_τ).

Step 8. If all targets are covered by active sensors, then go to Step 5; otherwise, let $\tau = \tau + 1$ and go to Step 3.

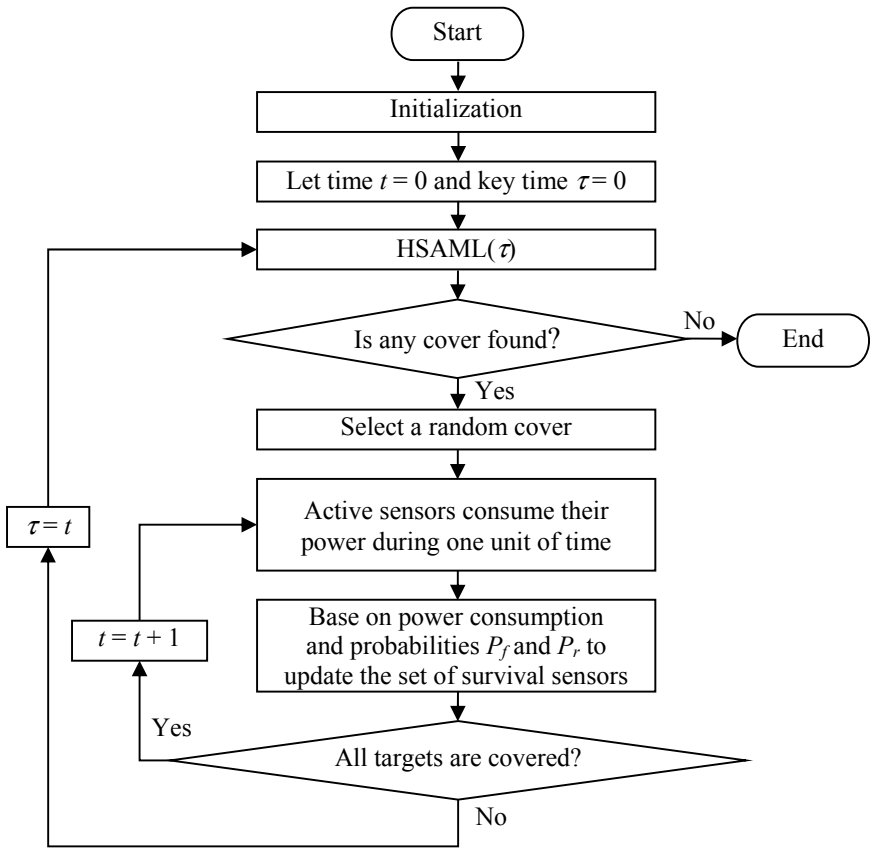


Fig. 4. Flowchart of the simulation.

5.1 Experimental setting

Part of parameters of the concerned problem are based on the model in [9]; and additional parameters for EH sensors and heterogeneity are based on practice. Those parameter settings are listed in Table 2. Note that most of the previous works (e.g., [29], [30]) did not set an upper bound for the energy storage of EH

sensors, nor does this work. In addition, with the same setting of some previous works (e.g., [31], [32]), this work assumes the power-harvesting rate of each EH sensor to be fixed.

No previous works had the same problem setting with the concerned problem. For instance, some works on extending the WSN lifetime focused on improving network structures of WSNs. The work in [33] adopted the method of layering in hierarchical routing to extend the WSN lifetime. The work in [6] considered the number and positions of sensor nodes to optimize the transmission trajectory to extend the WSN lifetime. The work in [5] proposed clustering schemes to suggest simple and static clustering strategies for a heterogeneous WSN with good harvesting efficiency to further extend the WSN lifetime. The work in [31] investigated the energy-harvesting wireless body area networks (EH-WBANs), and proposed a harvesting-rate oriented self-adaptive algorithm for enabling lifetime operation in EH-WBANs. However, the problem setting and environment of these previous works on extending the lifetime and the design of dynamics are not matched with ours, and hence these works cannot be compared experimental with this work. Therefore, we only focus on comparing the experimental results between the conventional HSA and the proposed HSAML.

Table 2. Parameter setting for the concerned problem.

Parameter	Value
The deployment area size	$50 \cdot 50$
Number of common sensors	150
Number of EH sensors	5
Number of targets	10
Fail probability	0.001
Recovery probability	0.001
Sensing coverage range of a common sensor	20, 30
Sensing coverage range of an EH sensor	10
Initial power of each common sensor	100 unit
Power consumption rate of each common sensor	1 unit/unit time
Power harvesting rate of each EH sensor	0.2 unit/unit time

For performance evaluation, the proposed HSAML is compared with the classical HSA, i.e., without multiple populations and local search. The algorithms are implemented in C++ programming language; and the simulation is conducted on a PC with Intel Core i5-3470 CUP and 8GB memory. Parameter settings for the algorithms are listed in Table 3, in which some parameters are referred to [19], including δ, hms, $HMCR$, PAR_{max}, PAR_{min}; and the others are set accordingly after lots of experimental trials.

Table 3. Parameter setting for the algorithms.

Parameter	Notation	Value
Number of *sub-HM*$_i$'s	δ	8
Size of the *HM* matrix	*hms*	40
Harmony moderation consideration rate	*HMCR*	0.95
Maximal *PAR* value	$PAR_{\max}$	0.99
Minimal *PAR* value	$PAR_{\min}$	0.45
Maximal *MT* value	$mr_{\max}$	0.40
Minimal *MT* value	$mr_{\min}$	0.10
Number of iterations	*NI*	30000
Size of tournament selection	*ts*	2

5.2 Experimental analysis under different scenarios

This section analyzes experimental results under different scenarios. Recall that $|S|$ denotes number of common sensors; and $|E|$ denotes number of EH sensors. Statistics of lifetime results under different combinations of $|S|$ and $|E|$ are shown in Table 4, which records the mean and standard deviation (StdDev) of the lifetime results of executing 20 times of HSA and HSAML. In all cases, our proposed HSAML performs better than HSA, and the performance difference becomes larger as the number of sensors increases. In comparing the cases with different numbers of EH sensors (i.e., $|E|$ = 0, 5, 10), more EH sensors have remarkable influence on lifetime extension.

Table 4. Statistics of lifetime results using two algorithms under different numbers of sensors.

$\|S\|$	$\|E\|$	HSA		HSAML	
		mean	StdDev	mean	StdDev
50	0	809.0	93.853	832.0	98.924
45	5	819.0	89.488	823.6	93.490
40	10	1749.2	154.849	1928.6	149.871
150	0	2270.4	194.491	2419.2	141.955
145	5	2551.6	121.685	2721.4	140.463
140	10	2904.8	142.112	3101.4	114.784
250	0	4021.2	163.062	4318.2	143.512
245	5	4467.8	125.382	4633.4	135.581
240	10	5105.8	148.724	5447.4	136.749

To observe algorithm convergence, Figs. 5, 6, and 7 gives the plots of fitness results versus number of iterations using two algorithms under different numbers of common sensors, EH sensors, and targets, respectively. In each figure, fitness fluctuates a lot because the WSN environment is changed dynamically at each 200 iterations in our experimental setting. From those figures, both the two algorithms can adapt to

the change, because they can recover to a high fitness level after a sudden fitness drop.

From Figs. 4 and 5, it is observed that a larger number of common sensors or EH sensors can increase fitness and further extend the WSN lifetime. Additionally, it is also observed that the proposed HSAML performs better than HSA, overall.

Remind that each sensor has four states. The simulation simplifies practical state of each sensor with two probabilities, i.e., each sensor has a probability P_f to fail and a probability P_r to be recovered. Figs. 8 and 9 gives the plots of fitness results versus number of iteration using two algorithms under different values of probabilities P_f and P_r, respectively. From Fig. 8, when P_f increases slightly (from 0.0005 to 001), the fitness performance becomes much worse. From Fig. 9, as P_r increases, the fitness performance increases. Overall, the proposed HSAML performs better than HSA. It is concluded that if quality of the hardware of sensor devices can be improved so that the fail probability becomes less, the lifetime could be extended.

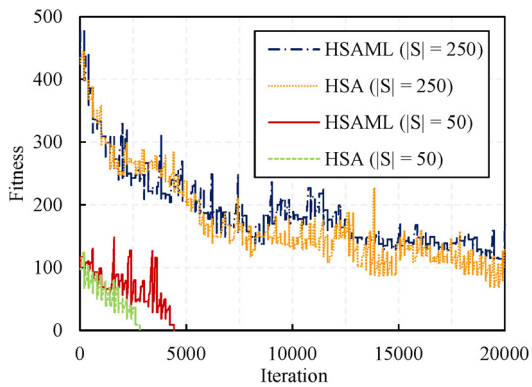


Fig. 5. Analysis on different numbers of common sensors.

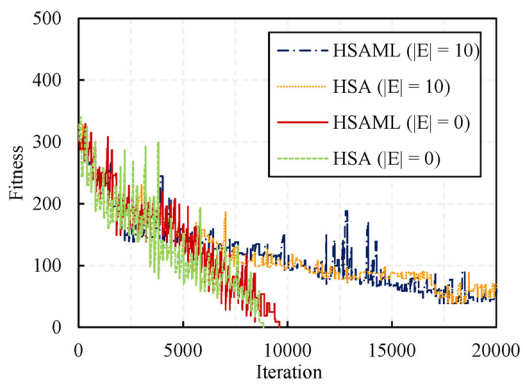


Fig. 6. Analysis on different numbers of EH sensors.

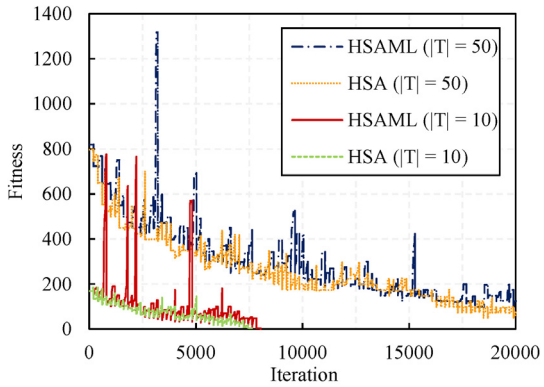


Fig. 7. Analysis on different numbers of targets.

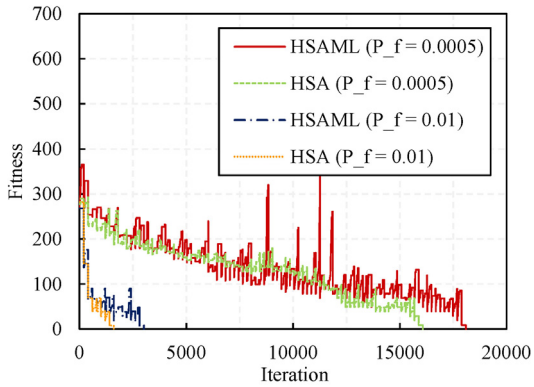


Fig. 8. Analysis on different fail probabilities.

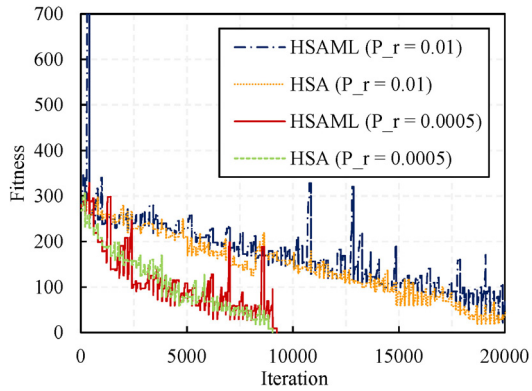


Fig. 9. Analysis under different recovery probabilities.

6. Conclusion

With advance in energy-harvesting (EH) sensors, this work has proposed a novel lifetime extension problem for dynamic heterogeneous WSNs with EH sensors. This problem is equivalent with finding the maximal number of sensor covers each of which can cover all targets. As this problem is NP-complete, this work has proposed a harmony search algorithm with multiple populations and local search (HSAML) for this problem. Simulation verifies outperformance of the proposed HSAML over classical HSA. In the future, more advanced techniques could be considered in the problem setting, including extension of new protocols and wireless charging. Additionally, implementation of the proposed algorithm on real sensor devices is of practical interest.

Acknowledgements

The authors thank the anonymous referees for comments that improved the content as well as the presentation of this paper. This work has been supported in part by MOST 104-2221-E-009-134-MY2.

References

[1] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. (2002) Wireless sensor networks: a survey. *Computer Networks* 38(4), pp. 393–422.

[2] Oliveira LM, Rodrigues JJ (2011) Wireless sensor networks: A survey on environmental monitoring. *Journal of Communications* 6(2), pp. 143–151.

[3] Ko J, Lu C, Srivastava M, Stankovic J, Terzis A, Welsh M (2010) Wireless sensor networks for healthcare. *Proceedings of the IEEE* 98(11), pp. 1947–1960.

[4] Aminian M, Naji HR (2013) A hospital healthcare monitoring system using wireless sensor networks. In: *Proceedings of Journal of Health & Medical Informatics (JHMI 2013)*, pp. 1–6.

[5] Awan S W, Saleem S (2016) Hierarchical clustering algorithms for heterogeneous energy harvesting wireless sensor networks. In: *Proceedings of* 2016 *International Symposium on Wireless Communication Systems (ISWCS 2016)*, IEEE Press, pp. 270–274.

[6] Yang C, Chin K W (2016) On nodes placement in energy harvesting wireless sensor networks for coverage and connectivity. *IEEE Transactions on Industrial Informatics*, in press.

[7] Slijepcevic S, Potkonjak M (2011) Power efficient organization of wireless sensor networks. In: *Proceedings of IEEE International Conference on Communications (ICC 2001)*, pp. 472–476, IEEE Press.

[8] Garey MR, Johnson DS (1979) *Computers and Intractability - A Guide to the Theory of NP-Completeness*, Freeman, San Francisco.

[9] Liao C, Ting C (2012) Extending the lifetime of dynamic wireless sensor networks by genetic algorithm. In: *Proceedings of IEEE World Congress on Computational Intelligence (WCCI 2012)*, pp. 1–8, IEEE Press.

[10] Cardei M, Du DZ (2005) Improving wireless sensor network lifetime through power aware organization. In: *Proceedings of IEEE Wireless and Mobile Computing, Networking and Communications (WiMob 2005)*, pp. 333–340, IEEE Press.

[11] Nezhad SE (2010) Solving k-coverage problem in wireless sensor networks using improved harmony search. In: *Proceedings of IEEE Broadband, Wireless Computing, Communication and Applications (BWCCA 2010)*, pp. 49–55, IEEE Press.

[12] Cardei M, Wu J, Lu M, Pervaiz M (2005) Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In: *Proceedings of IEEE Wireless and Mobile Computing, Networking and Communications (WiMob 2005)*, pp. 438–445, IEEE Press.

[13] Liu F, Tsui C, Zhang YJ (2010) Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks. *IEEE Transactions on Wireless Communications* 9(7), pp. 2258–2267.

[14] Zhao Y, Wu J, Li F, Lu S (2012) On maximizing the lifetime of wireless sensor networks using virtual backbone scheduling. *IEEE Transactions on Parallel and Distributed Systems* 23(8), pp. 1528–1535.

[15] Sudevalayam S, Kulkarni P (2010) Energy harvesting sensor nodes: Survey and implications. In: *Proceedings of IEEE Communications Surveys Tutorials (CST 2010)*, pp. 1–19, IEEE Press.

[16] Zhang P, Xiao G, Tan H (2013) Clustering algorithms for maximizing the lifetime of wireless sensor networks with energy-harvesting sensors. *Journal of Computer Networks* 57(4), pp. 2689–2704.

[17] Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1(1), pp. 67–82.

[18] Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation* 188(2), pp. 1567–1579.

[19] Lin CC, Deng DJ, Wang SB (2016) Extending the Lifetime of Dynamic Underwater Acoustic Sensor Networks Using Multi-Population Harmony Search Algorithm. *IEEE Sensors Journal* 16(11), pp. 4034–4042.

[20] Shaikh FK, Zeadally S (2016) Energy harvesting in wireless sensor networks: A comprehensive review. *Journal of Renewable and Sustainable Energy Reviews* 55, pp. 1041–1054.

[21] Azevedo JAR, Santos FES (2012) Energy harvesting from wind and water for autonomous wireless sensor nodes. *IET Circuits, Devices & Systems* 6(6), pp. 413–420.

[22] Kansal A, Hsu J, Zahedi S, Srivastava MB (2007) Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems* 6(4), pp. 1–32.

[23] Michelusi N, Badia L, Carli R, Corradini L, Zorzi M (2013) Energy management policies for harvesting-based wireless sensor devices with battery degradation. *IEEE Transactions on Communications* 61(12), pp. 4934–4947.

[24] Zhang H, Hou JC (2004) Maintaining sensing coverage and connectivity in large sensor networks. In: *Proceedings of International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, pp. 89–124.

[25] Geem ZW, Kim JH (2001) A new heuristic optimization algorithm: harmony search. *Simulation* 76(2), pp. 60–68.

[26] Castelli M, Silva S, Manzoni L, Vanneschi L (2014) Geometric selective harmony search. *Information Sciences* 279(20), pp. 468–482.

[27] Karimi M, Askarzadeh A, Rezazadeh A (2012) Using tournament selection approach to improve harmony search algorithm for modeling of proton exchange membrane fuel cell. *International Journal of Electrochemical Science* 7(7), pp. 6426–6435.

[28] Syswerda G (1980) Uniform crossover in genetic algorithms. In: *Proceedings of the 3rd International Conference on Genetic Algorithms (ICGA 3rd)*, pp. 2–9.

[29] Mehrabi A, Kim K (2016) General framework for network throughput maximization in sink-based energy harvesting wireless sensor networks. *IEEE Transactions on Mobile Computing*, in press.

[30] Mehrabi A, Kim K (2016) Optimal transmission period for improved sink-based data collection in energy harvesting wireless sensor networks. In: *Proccedings of IEEE International Conference on Communications (ICC 2016)*, pp. 1–6.

[31] Qi X, Wang K, Huang A (2015) A harvesting-rate oriented self-adaptive algorithm in energy-harvesting wireless body area networks. In: *Proceedings of IEEE 13th International Conference on Industrial Informatics (INDIN 2015)*, pp. 966–971.

[32] Kunikawa M, Yomo H, Abe K, Ito T (2015) A fair polling scheme for energy harvesting wireless sensor networks. In: *Proceedings of IEEE 81st Vehicular Technology Conference (VTC Spring 2015)*, pp. 1–5.

[33] Sedighimanesh A, Sedighimanesh M, Baqeri J (2015) Improving wireless sensor network lifetime using layering in hierarchical routing. In: *Proceedings of 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, IEEE Press, pp. 1145-1149.