A Generalized Coverage-Preserving Scheduling in WSNs: a Case Study in Structural Health Monitoring

Xuefeng Liu,Jiannong Cao
Dept. of Comp.
Hong Kong PolyU
{csxfliu,csjcao}@comp.polyu.edu.hk

Shaojie Tang
Dept. of Comp. Sci.
Temple University
tangshaojie@gmail.com

Peng Guo
Dept. of Comp.
Huazhong Univ. of Sci. and Tech.
guopeng@mail.hust.edu.cn

***Abstract*—Wireless sensor networks (WSNs) are generally used to monitor, in an area, certain phenomena which can be events or targets that users are interested. To extend the system lifetime, a widely used technique is 'Energy-Efficient Coverage-Preserving Scheduling(EECPS)', in which at any time, only part of the nodes are activated to fulfill the function. To determine which nodes should be activated at a certain time is the key for the EECPS and this problem has been studied extensively. Existing solutions are based on the assumption that each node has a fixed coverage area, and once the event/target occurs in this area, it can be detected by this sensor. However, this coverage model is not always valid. In some applications such as structural health monitoring (SHM) and volcano monitoring, to fulfill a required function always requires low level collaboration from multiple sensors. The coverage area for individual sensor node therefore cannot be defined explicitly since single sensor is not able to fulfill the function alone, even it is close to the event or target to be monitored. In this paper, using an example of SHM, we illustrate how to support EECPS in some special applications of WSNs. We re-define the 'coverage' and based on the new coverage model, two methods are proposed to partition the deployed sensor nodes into qualified cover sets such that the system lifetime can be maximized by letting these sets work by turns. The performance of the methods is demonstrated through extensive simulation and experiment.**

I. Introduction

Wireless sensor networks (WSNs) are widely used to detect events or targets that users are interested in a given area [1]. To extend system lifetime, a widely used technique is to activate only part of the sensors each time and put the others into sleep mode. Since the working sensors should still be able to cover the whole monitoring area, this technique is generally called 'energy-efficient coverage-preserving scheduling (EECPS)'.

Obviously, to determine which sensor nodes should be activated at a certain time is the key for the EECPS. Many methods have been proposed and they are closely associated with the concept of 'coverage'. Some methods are centralized [2][3][4] which allocate sensor nodes into the mutually exclusive or overlapping cover sets. Some are distributed [5][6][7] which determine a node's activity according to whether or not its sensing region has already been covered by its active neighbors. Both centralized and distributed methods assume that *each sensor node has a fixed coverage area which is defined explicitly and once the event/target occurs in this area, it can be detected by this node*. However, this assumption may not be valid in all applications. For some applications of WSNs, detecting the event/target of interest requires the low level collaboration from multiple sensors. For example, in structural health monitoring (SHM), wireless sensor nodes are deployed on a structure to detect possible structural damage on it. Detecting structural damage relies on the vibrational features of the structure, which are always extracted from data collected at multiple sensor nodes. As a result, a fixed coverage area for individual sensor node does not exist and the EECPS cannot be directly applied in WSN-based SHM.

In this paper, taking SHM as an example, we illustrate how to support EECPS in some specific applications of WSNs. We first define a generalized coverage model. Different from conventional coverage model which is a geographic area defined for individual sensors, this generalized coverage model does not need to have the coverage area of individual nodes, but only rely on a function to determine whether a set of sensor nodes is able to fulfill the required monitoring task in a given area. Under this generalized coverage model, we then propose two methods to partition the deployed nodes into qualified cover sets, which can work in turn to extend the system lifetime. The performance of the partition methods is demonstrated through extensive simulations on a bridge model.

Our contribution in this paper is as follows:

1) We found that traditional coverage model are not suitable for some applications of WSNs and correspondingly, the EECPS cannot be directly applied.
2) Taking an example of SHM, we re-define the 'coverage' and based on this generalized coverage model, two methods are proposed to support EECPS in SHM.
3) The proposed methods can be easily generalized to other application areas besides SHM such as volcano monitoring in which the sensing area of individual nodes cannot be defined explicitly.

II. Related Works

In many traditional applications of WSNs, such as battlefield surveillance, environmental and habitat monitoring, sensor

978-14799-3360-0/14/$31.00 ©2014 IEEE

nodes are deployed in an area to detect possible events or targets that are interested to application users. Usually, the number of deployed sensor nodes is larger than required to cover the area or targets. To extend the system lifetime, these sensor nodes can be divided into multiple sets, and they are activated to work in turn. This technique is generally called 'energy-efficient coverage-preserving scheduling' (EECPS).

In EECPS, the methods to determine which sensors should be activated in a round can be largely divided as centralized and distributed. Centralized methods can be found in [2] and [3] which allocate sensor nodes into the maximum number of mutually exclusive cover sets. A improved version was proposed in [4], where sensors are allowed to participate in multiple sets. The basic idea of the distributed methods is mainly as follows: by exchanging information with the active neighbors, a sensor node knows whether or not its sensing region has already been covered by its active neighbors and will then be activated or go to sleep accordingly. Examples can be found in [5][6][7].

However, both centralized and distributed methods mentioned above would fail in some applications of WSNs where detection events or targets requires collaboration from multiple sensor nodes. Since individual sensor node is not able to fulfill the task alone, a definite coverage area for individual sensor node does not exist.

In this paper, using an example of SHM, we illustrate how to support EECPS in more applications of WSNs. We redefine the 'coverage' which is based on a sensor set instead of individual sensors. Then based on this new coverage model, two methods are proposed to partition the deployed sensor nodes into qualified cover sets such that the system lifetime is maximized.

III. BACKGROUND

In this section, we introduce some preliminaries in SHM and how event SHM (i.e. structural damage) is detected.

A. Natural frequency and mode shape

Every structure has tendency to oscillate with much larger amplitude at some frequencies than others. These frequencies are called **natural frequencies**. When a structure is vibrating under one natural frequency, the corresponding vibration pattern it exhibits is called the **mode shape** of this natural frequency. For a structure with n-degrees of freedom (DOFs), its natural frequency set and mode shapes are denoted as:

$$\mathbf{f} = [f^1, f^2, ..f^n]', \mathbf{\Phi} = [\mathbf{\Psi}^1, \mathbf{\Psi}^2, \cdots, \mathbf{\Psi}^n] \quad (1)$$

where f^k $(k = 1..., n)$ is the k^{th} natural frequency, $\mathbf{\Psi}^k(k = 1, ..., n)$ is the mode shape corresponding to f^k. $\mathbf{\Psi}^k$ is a n-by-1 vector and each element corresponds to a specific i^{th} DOF. f^k and $\mathbf{\Psi}^k$ are also called **modal parameters** of the k^{th} **mode** of a structure. As an example, Fig. 1 shows the first 3 modal parameters of a typical cantilevered beam.

Modal parameters are internal properties of structure and by examining the changes in the modal parameters, damage on a structure can be identified [8]. However, natural frequencies

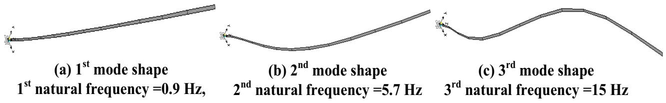

Fig. 1: The first 3 modal parameters of a cantilever beam

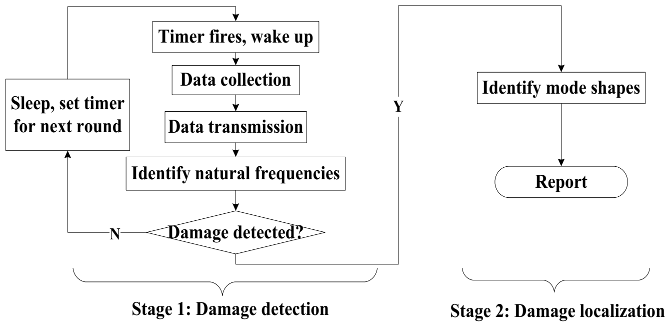

Fig. 2: The multi-tired strategy in SHM

and mode shapes play different roles in damage identification. It can be seen from Eq. 1 that the natural frequencies do not contain any spatial information. This means that by examining the changes of natural frequencies, only the existence of structural damage can be obtained. On the other hand, mode shape $\mathbf{\Psi}^k$ has an element corresponding to each DOF and thus contains spatial information. Mode shapes and their derivatives have been proven to be very effective features to locate structural damage [8].

A multi-tired strategy shown in Fig. 2 is adopted by WSNs to detect and locate structural damage. In this approach, every period of time, deployed nodes collect a number of data and transmit them to the sink where the natural frequencies are identified to detect the existence of damage. Once damage is detected, mode shapes are identified to give damage locations.

In this paper, both natural frequencies and mode shapes are identified using the classic eigen-system realization algorithm (ERA) [9] which will be described in the next section. Interestingly, we will show that the *identification of natural frequencies does not requires data from all the sensor nodes and therefore can be energy efficient*. This justifies why natural frequency instead of mode shape is used at the first stage.

In this multi-tired strategy, damage detection is of the most importance since most of the time, a system implementing this strategy will be only running at this stage. Therefore, damage detection will be the focus of this paper.

B. The identification of natural frequencies using the ERA

In this section, we briefly introduce how the natural frequencies are identified in the ERA and the associated cost.

From measured responses from deployed sensor nodes, the ERA first calculates two Hankel matrices and then implement a series of matrix computations such as singular value decomposition (SVD), and eigen decomposition on these two matrices to obtain an estimate of natural frequencies $\hat{\mathbf{f}}$ [9]. Note that to detect structural damage, only the first few frequencies needs to be identified. The number of frequencies to be identified, denoted as p_{mod}, is determined by civil analysts.

Traditionally, to identify natural frequencies, data from all the deployed sensor nodes are used in the ERA [9]. However, this approach is not suitable for a WSN. Interestingly, in the ERA, accurately identifying natural frequencies does not require data from all the deployed sensor nodes. *Data from some properly selected sensor nodes can also obtain very accurate natural frequencies*. Briefly speaking, the accuracy of $\hat{\mathbf{f}}$ identified from a sensor set S is determined by the measurement noise, the finite element model (FEM) of the structure, and the number as well as the locations of nodes in S. Furthermore, according to [10], under the assumptions that (1) all the sensor nodes in S have the same measurement noise, and (2) $|S| \geq p_{mod}$ where $|S|$ is the number of sensor nodes in S, the accuracy of identified $\hat{\mathbf{f}}$ is determined by the condition number of S:

$$cond(S) = \|\mathbf{\Phi}_S\| \cdot \left\|\mathbf{\Phi}_S^{-1}\right\| \quad (2)$$

where $\|.\|$ is the Euclidean norm, and $\mathbf{\Phi}_S$ is the structrue's mode shape matrix $\mathbf{\Phi}$ in Eq. 1 retaining only the rows corresponding to the DOFs of S. The larger the $cond(S)$, the less accurate of identified natural frequencies will be from S. $\mathbf{\Phi}_S$ can be obtained by the FEM of the structure. In this paper, we assume that the measurement noise is the same for all the sensor nodes and p_{mod} is a pre-determined value. Therefore, when identifying natural frequencies, we only need to choose a sensor set S with $|S| \geq p_{mod}$ and the $cond(S)$ is smaller enough so that the identified natural frequencies are accurate enough to detect a certain level of damage on the structure. Moreover, there may exist many sensor sets in V which are able to satisfy this requirement. If these sets work successively, system lifetime can be significantly extended. This is exactly the idea EECPS.

The energy cost of each node in a WSN when implementing ERA is formulated as follows. Assume the WSN is denoted as $G = (V, E)$ where V being the sensor nodes and E being the wireless links among V. For a sensor node $v_i \in V$, its total energy consumption in the ERA, denoted as $cost(v_i, V)$, has different form according to its 'status' in V. If v_i is the sink node, the $cost(v_i, V)$ can be represented as:

$$cost(v_i, V) = N \cdot e_S + (|V| - 1)N \cdot e_R + e_{ERA}(|V|) \quad (3)$$

where N is the total amount of time history record sampled in each sensor, e_S and e_R are the energy for sampling and receiving one data, respectively. $|V|$ is the number of nodes in V, e_{ERA} is the energy consumed in the ERA.

If v_i is not a sink node, the $cost(v_i, V)$ only includes data sampling and wireless communication parts:

$$cost(v_i, V) = N \cdot e_S + \sum_{\substack{\forall p_a, \exists j>0, \\ p_a[j]=v_i}} N \cdot e_R + \sum_{\substack{\forall p_a \exists j, \\ p_a[j]=v_i}} N \cdot e_T \quad (4)$$

In Eq. 4, we assume a routing algorithm is given and let the path from a node in V to the sink node be $p_a = h_0 h_1, \cdots, h_k$. The $p_a[i] = h_i$ is the i^{th} hop sensor on path p_a and the e_T is the energy cost for transmitting one data.

Eq. 3 and Eq. 4 can be easily extended to the condition when a sensor set $S \subseteq V$ is adopted and they will be used when formulating the problem in the next section.

IV. Problem formulation

In the first stage of the multi-tired strategy shown in Fig. 2, if the EECPS can be applied, system lifetime can be significantly increased. However, existing approach to partition the sensor nodes into cover sets cannot be directly used here since individual node in SHM is not able to detect structural damage, and we cannot define the coverage area for each sensor node.

In SHM, the detection of structural damage is directly connected with the accuracy of identified natural frequencies. Moreover, according to Eq. 2, given a structure, and a certain level of damage to be detected, a p_{mod}, we can determine a threshold γ for $cond(S)$ which determines whether the sensor set S is capable of detecting given damage on a structure. We call this SHM coverage:

Definition 1 (SHM coverage)**.** *A sensor set S is able to 'SHM cover' the whole area of the structure iff $|S| \geq p_{mod}$ and $cond(S) \leq \gamma$.*

It can be seen that we use a coverage model which is different from traditional one. Traditional coverage model is a geographic area defined for individual sensors, and this new coverage model does not need to have the coverage area of individual nodes, but rely on a function shown above which is only able to determine whether a set of sensor nodes is able to cover the whole monitoring area.

Also should be noted is that the $cond(S)$ cannot be determined by accumulating the condition number of each sensor node in S. Therefore, whether a sensor set S is a SHM cover set cannot be determined by considering sensors in S individually.

After the definition of SHM coverage, the objective becomes ***to divide the deployed sensor nodes into multiple SHM cover sets and when they are activated in turn, the total number of rounds they work is maximized.***

Considering that relaying large amount of sampled data through multiple hops consumes energy as well as wireless bandwidth, we require that each SHM cover set is a single-hop cluster. The cluster head (CH) in each SHM cover set needs to collect the data from others and implement the ERA to identify natural frequencies. Under this constraint, Eq. 4 is simplified to

$$cost(v_i, V) = N \cdot e_S + N \cdot e_T \quad (5)$$

but Eq. 3 is kept unchanged.

Now we give the formal problem definition below:

Maximum SHM Cover Set Problem (MSSP):

Given:

- a WSN $G = (V, E)$
- p_{mod}, γ, and $cond(S)$ which determines whether a set $S \subseteq V$ is a SHM cover set.
- $cost(v_i, S)$, which is the energy consumption of each node v_i when a sensor set $S \subseteq V$ is chosen to implement one round of damage detection. If $v_i \in S$, $cost(v_i, S)$ is taking in the form of Eq. 3 or Eq. 5 according to the role it plays in S. If $v_i \notin S$, $cost(v_i, S) = 0$.
- er_i, the remaining energy of each sensor node v_i.

The problem is to find a family of SHM cover sets $S_1, \cdots, S_p$ ($S_j \subseteq V, j = 1, \cdots, p$) with the corresponding number of rounds $r_1, \cdots, r_p$ allocated for these sets such that $r_1 + r_2 + \cdots + r_p$ is maximized subject to the following constraints:

- $\forall j = 1, \cdots, p$, $|S_j| \geq p_{mod}$ and $cond(S_j) \leq \gamma$
- $\forall j = 1, \cdots, p$, let the sub-graph for S_j in $G(V, E)$ be $G(S_j, E_j)$, where $E_j \subseteq E$. Then $\exists v \in S_j$, such that $\forall u \in S_j (u \neq v)$ there is an edge $a_{vu} \in E_j$ between v and u.
- $\forall v_i \in V$, $\sum_{j=1}^{p} cost(v_i, S_j) r_j \leq er_i$

Remarks:

- Inspired by [4], instead of dividing V into disjoint sets, we allow every node to be part of more than one set, and allow the sets to work for different number of rounds.
- The first constraint guarantees each set is a SHM cover set. The second constraint is to ensure only the single-hop SHM cover sets are generated. The last constraint guarantees that the total energy consumed for each node v_i across all SHM cover sets is no larger than er_i.

We prove that the MSSP is NP-hard by proving that the decision version of the problem is NP complete which is defined as: ***given a threshold*** k ***, does there exist a collection of sensor sets*** $C = \{S_1, S_2, \cdots, S_p\}$ ***and the corresponding*** $r_1, \cdots r_p$***, which satisfy all the constraints above and*** $r_1 + r_2 + \cdots + r_p$ ***is equal of larger than*** k***?***

Proof: It is easy to prove this problem is NP. We then show that it is NP-complete by reducing the set packing problem [11] to it. The set packing problem is defined as:

Given: A universe $U = \{\bar{s}_1, \bar{s}_2, \cdots, \bar{s}_m\}$, a collection of subsets: $\bar{C} = \{\bar{S}_1, \bar{S}_2, \cdots, \bar{S}_p\}$ with $\bar{S}_j \subseteq U$ for $j = 1, \cdots, p$, and a number k.

Find: if there exist k subsets in $\bar{C}$ which are pairwise disjoint (in other words, no two of them intersect).

To reduce the set packing problem to the MSSP, we construct a network $G = (V, E)$, mode shape matrix $\mathbf{\Phi}$, the number of natural frequencies to be identified p_{mod}, cost function $cost(v_i, S)$, threshold γ, and remaining energy er_i from the inputs of set packing problem in the following way:

1. For each $\bar{S}_j \in \bar{C}$, establish a local network $\bar{G}_j = (\bar{V}_j, \bar{E}_j)$ with a hub-spoke architecture. $\bar{V}_j$ includes one CH, denoted as $\bar{v}_0^j$ and m cluster members $\{\bar{v}_1^j, \bar{v}_2^j, \cdots, \bar{v}_m^j\}$. Each $\bar{v}_i^j, i = 1, \cdots, m$ corresponds to an element $\bar{s}_i \in U$. The mode shape vector corresponding to the CH $\bar{v}_0^j$ is a 1-by-m row vector with only the first element to be '1': $[1, 0, \cdots, 0]$, and the mode shape vector for the cluster member $\bar{v}_i^j$ is $[0, \cdots, 1, 0, \cdots]$, with the i^{th} element to be '1'. This setting applies for all the $\bar{S}_j \in \bar{C}$. It can be seen that after this stage, a total of p equivalent local star networks are generated.

2. For each $\bar{s}_k \in \bar{S}_i \cap \bar{S}_j$ ($\bar{S}_i, \bar{S}_j \in \bar{C}$ and $\bar{S}_i \cap \bar{S}_j \neq \emptyset$), the corresponding sensor nodes $\bar{v}_k^j$ and $\bar{v}_k^i$ in the local networks $\bar{G}_i$ and $\bar{G}_j$ will be merged. After this stage, we obtained a WSN $G = (V, E)$ which includes m single-hop clusters. Each cluster has exactly $m+1$ sensor nodes and some clusters may overlap with each other.

3. By adjusting e_T and e_R in Eq. 3 and Eq. 5, we make that given a single-hop network $\bar{G}_j = (\bar{V}_j, \bar{E}_j)$ with $|\bar{V}_j| = m+1$, the $cost(v, \bar{V}_j)$ ($v \in \bar{V}_j$) is the same no matter whether v is a CH or not. Also, the remaining energy of each node in $\bar{G}_j$ is set to be $er = cost(v, \bar{V}_j)$. This setting imposes that every sensor node can only make one round of damage detection and hence the SHM cover sets established using this approach are disjoint. $p_{mod} = m$ and the threshold γ is set to be 1.

With this transformation, it can be easily proved that 1) Assume, without loss of generality, $\{\bar{S}_1, \bar{S}_2, \cdots \bar{S}_k\}$ is a solution to the set packing problem, then the local networks $\{\bar{G}_1, \bar{G}_2, \cdots \bar{G}_k\}$ are the SHM cover sets which satisfy the constraints in the MSSP, and the number of rounds of damage detection assigned are $r_1 = r_2 = \cdots = r_k = 1$. Therefore, the total number of rounds of damage detection is k. 2) Assume a $G = (V, E)$ is constructed from the set packing problem and we have SHM cover sets $\{\bar{G}_1, \bar{G}_2, \cdots\}$ with $\sum_{j=1} r_j = k$ to the MSSP problem, then the subsets $\{\bar{S}_1, \bar{S}_2, ...\bar{S}_k\}$ is a solution to the set packing problem, where $\bar{S}_j$ ($j = 1, \cdots k$) is the subset from which the local network $\bar{G}_j$ is established. The detailed proof is omitted for brevity. ■

V. Proposed Methods

In this section, we describe two methods to solve the MSSP, one heuristic method based on multi-dimensional knapsack problem and the other based on the genetic algorithm.

A. A BMKP based Algorithm for the MSSP

In this section, a heuristic method is proposed. This method is largely divided into two stages. First, a set of candidate single-hop SHM sets are enumerated. Then the problem is reduced to the bounded multi-dimensional knapsack problem.

Given $G = (V, E)$, we find out, for each $v_i \in V$, a sensor set S_{v_i} which includes v_i and all its one-hop neighbors. Based on S_{v_i}, a group of sensor sets, denoted as $\mathbb{S}_{v_i}$, is established

$$\mathbb{S}_{v_i} = \{S_{v_i}^1, S_{v_i}^2, ...\} \tag{6}$$

Sensor sets in $\mathbb{S}_{v_i}$ are constructed from nodes in S_{v_i} and must satisfy the following constraints:

1) $\forall S_{v_i}^k \in \mathbb{S}_{v_i}, v_i \in S_{v_i}^k$ and $S_{v_i}^k \subseteq S_{v_i}$
2) $\forall S_{v_i}^k \in \mathbb{S}_{v_i}, |S_{v_i}^k| \geq p_{mod}$ and $cond(S_{v_i}^k) \leq \gamma$
3) $\forall S_{v_i}^{k_1}, S_{v_i}^{k_2} \in \mathbb{S}_{v_i} (k_1 \neq k_2), S_{v_i}^{k_1} \nsubseteq S_{v_i}^{k_2}$

Briefly speaking, the first constraint requires that for each $S_{v_i}^k \in \mathbb{S}_{v_i}$, it must contain v_i and all the nodes in $S_{v_i}^k$ are chosen from v_i's one-hop neighbors. v_i will be functioning as the CH in $S_{v_i}^k$ if $S_{v_i}^k$ is selected to detect damage. The second constraint is to ensure that each $S_{v_i}^k$ is a SHM cover set. The third constraint is to remove some redundant candidates when solving the knapsack problem. For two sets $S_{v_i}^{k_1}$, $S_{v_i}^{k_2}$, both satisfying the first two constraints above and $S_{v_i}^{k_1} \subseteq S_{v_i}^{k_2}$, it can be easily proved that when a solution of the MSSP contains $S_{v_i}^{k_2}$, a better or at least equally good solution can be obtained by replacing $S_{v_i}^{k_2}$ with $S_{v_i}^{k_1}$.

For each node $v_i \in V$, we wish to enumerate all possible sensor sets that satisfy the constraints above. Theoretically, the number of possible sets in $\mathbb{S}_{v_i}$ could grow exponentially with respect to $|S_{v_i}|$. However, different from many applications of WSNs where thousands or tens of thousands cheap wireless motes can be deployed, the number of sensor nodes deployed on a civil structure are generally less than one hundred considering the cost of attached sensors and the application requirement. $|S_{v_i}|$ is thus further limited. Therefore, we can assume that $\mathbb{S}_{v_i}$ for any sensor node $v_i \in V$ can be enumerated in relatively short time. For a WSN $G = (V, E)$ where $|V| = m$, a collection of sensor sets which satisfy the constraints above is denoted as:

$$C_V = \mathbb{S}_{v_1} \bigcup \mathbb{S}_{v_2} \bigcup \cdots \mathbb{S}_{v_m} = \{S_1, S_2, \cdots, S_M\} \qquad (7)$$

where S_i is the i^{th} set in C_V and M is the total number of sets in C_V. Note that when determining the sensor sets for each $\mathbb{S}_{v_i}$, the energy constraint in the MSSP is not considered.

After all the possible SHM cover sets have been enumerated, the MSSP is transformed to the bounded multi-dimensional knapsack problem (BMKP), which is stated as:

$$\begin{aligned}\text{maximize } & \sum_{j=1}^{n} b_j x_j,\ x_j = \{0, 1, \cdots l_j\},\\ \text{subject to } & \sum_{j=1}^{n} w_{ij} x_j \le c_i, i = 1, \cdots, d\end{aligned}$$

where n is the number of object types, d is the number of dimensions that restraint the knapsack, b_j represents the benefit of the object j in the knapsack, x_j is a non-negative integer that indicates how many objects with type j have been stored in the knapsack ($x_j = \{1, \cdots, l_j\}$) or remains out ($x_j = 0$), c_i represents the i^{th} dimension's capacity of the knapsack, and w_{ij} represents the entries of the knapsack's constraints matrix.

The MSSP is written in the form of the BMKP as below:

$$\begin{aligned}\text{maximize } & \sum_{j=1}^{M} r_j,\ r_j = \{0, 1, \cdots l_j\}, l_j = \min_{v_i \in S_j} \lfloor \frac{er_i}{cost(v_i, S_j)} \rfloor\\ \text{subject to } & \sum_{j=1}^{M} cost(v_i, S_j) r_j \le er_i, i = 1, \cdots, m\end{aligned}$$

It can be seen that when the MSSP is expressed in the form of BMKP, each sensor set $S_j \in C_V$ represents a certain type of item in the BMKP. All sensor sets in C_V have the same benefit (i.e. $b_j = 1$ for all $j = 1, \cdots, M$). We wish to maximize the total profit of the selected sensor sets ($\sum_{j=1}^{M} r_j$) under the constraint that the energy of each sensor node across all the selected sensor sets will not exceed its threshold:

$$\sum_{j=1}^{M} cost(v_i, S_j) r_j \le er_i \qquad (8)$$

For a sensor set S_j, the maximum number it can be selected is bounded by the sensor node in S_j whose energy will be depleted first when implementing the ERA (i.e. $\min_{v_i \in S_j} \lfloor \frac{er_i}{cost(v_i, S_j)} \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function which maps a real number to the largest previous integer).

After this transformation, we are able to employ existing algorithms originally designed for the BMKP to tackle our problem. In particular, we first transform the BMKP into 01MKP based on [12], and then adopt the cross-entropy optimization [13] to solve the 01MKP.

B. The GA Method for the MSSP

In this section, we propose a Genetic Algorithm (GA) to solve the MSSP. We will show later that in some conditions, the GA method can achieve satisfactory results comparable with the heuristic method but using less computation time.

A basic GA includes: [14]:

Generate an initial population;
repeat
 Select parents from the population to produce offspring;
 Evaluate fitness of the children and replace some or all of the population by the offspring with good fitness;
until a satisfactory solution has been found.

The first step in designing a GA for the MSSP is to devise a suitable representation scheme to represent solution of MSSP. As before, we first find out S_{v_i} for each v_i which includes v_i and all its one-hop neighbors in $G = (V, E)$. Then we have a group of sensor sets:

$$\bar{C}_V = \{S_{v_1}, S_{v_2}, \cdots, S_{v_m}\} \qquad (9)$$

It should be noted that $\bar{C}_V$ only includes m sensor sets and is different from C_V defined in Eq. 7. All the sensor sets in $\bar{C}_V$ are aligned together as an array and we use a $\sum_{i=1}^{m} n_i$-bit binary string `str` to encode it, where m is the number of nodes in V and n_i is the number of nodes in S_{v_i}. In this representation, a value of 1 or 0 at the j^{th} bit of `str` implies that whether the j^{th} node in $\bar{C}_V$ is selected or not. Therefore, from each S_{v_i}, at most one sensor set is generated by selecting the nodes in S_{v_i} whose value in `str` are encoded as $'1'$. Furthermore, we require that in S_{v_i}, if v_i is not selected, no sensor set will be generated from S_{v_i}, even if some other nodes in S_{v_i} are selected according to the `str`. Using this procedure, a collection of sensor sets under `str` is generated:

$$\bar{C}_V^{str} = \{S_1^{str}, S_2^{str}, \cdots, S_p^{str}\} \qquad (10)$$

where $p \le m$ since at most m sets can be obtained. The sensor sets in $\bar{C}_V^{str}$ will be a candidate solution for the MSSP.

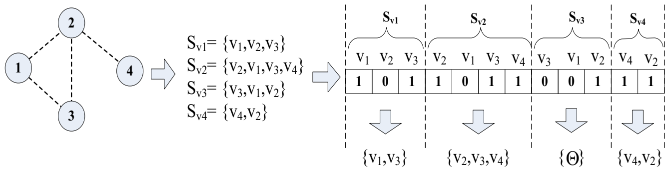

Fig. 3: GA example. Left: the topology of a WSN; Middle: The S_{v_i} of each node; Right: the encoding of $S_{v_1} \sim S_{v_4}$

We use an example shown in Fig. 3 to demonstrate the procedures above. According to the network topology shown in the left figure of Fig. 3, four S_{v_i}s are obtained. When they are aligned as an array, we have:

$$\bar{C}_V = \underbrace{\{v_1, v_2, v_3\}}_{S_{v_1}} \underbrace{\{v_2, v_1, v_3, v_4\}}_{S_{v_2}} \underbrace{\{v_3, v_1, v_2\}}_{S_{v_3}} \underbrace{\{v_4, v_2\}}_{S_{v_4}} \quad (11)$$

Assume a binary string `str` on the $\bar{C}_V$ is $'101101100111'$. According to this `str`, three sets $\{v_1, v_3\}$, $\{v_2, v_3, v_4\}$, and $\{v_4, v_2\}$ are respectively generated from S_{v_1}, S_{v_2}, and S_{v_4} and are shown in the right figure of Fig. 3. No sensor set came from S_{v_3} since its CH v_3 is 0 in `str`. Therefore, we have:

$$\bar{C}_V^{str} = \{\{v_1, v_3\}, \{v_2, v_3, v_4\}, \{v_4, v_2\}\} \quad (12)$$

So far, the encoding method described above only indicates which sensor sets will be chosen to work but does not provide the number of rounds assigned to them. To solve this problem, we modify the $\bar{C}_V$ in Eq. 9 and allow the repeated S_{v_i}:

$$\bar{C}_V = \{\underbrace{S_{v_1}, \cdots, S_{v_1}}_{\text{repeated } rep_1 \text{ times}}, \cdots \underbrace{S_{v_m}, \cdots, S_{v_m}}_{rep_m \text{ times}}\} \quad (13)$$

where rep_i ($i = 1, \cdots, m$) is the number of S_{v_i} repeated in $\bar{C}_V$. Theoretically, rep_i should be no less than the maximum number of rounds that any sensor sets generated from S_{v_i} (by mapping `str` on $\bar{C}_V$) can work to implement the ERA:

$$rep_i \geq \max_{\forall \bar{S}_{v_i} \subseteq S_{v_i}} \min_{\forall v_k \in \bar{S}_{v_i}} \lfloor \frac{er_{v_k}}{cost(v_k, \bar{S}_{v_i})} \rfloor \quad (14)$$

where er_{v_k} is the remaining energy of v_k, $\bar{S}_{v_i}$ is a subset of S_{v_i} but still using v_i as the CH. Given $\bar{S}_{v_i}$, $\min_{\forall v_k \in \bar{S}_{v_i}} \lfloor \frac{er_{v_k}}{cost(v_k, \bar{S}_{v_i})} \rfloor$ is the number of rounds that $\bar{S}_{v_i}$ is able to work. Since v_i is always included in S_{v_i} and the $cost(v_i, \bar{S}_{v_i})$ is a non-decreasing function with $|\bar{S}_{v_i}|$, a safe choice of rep_i is that

$$rep_i = \lfloor \frac{er_{v_i}}{cost(v_i, \bar{S}_{v_i} = v_i)} \rfloor \quad (15)$$

Note that $\bar{S}_{v_i}$ in Eq. 15 only includes v_i, and obviously, $\lfloor \frac{er_{v_i}}{cost(v_i, v_i)} \rfloor \geq \max_{\forall \bar{S}_{v_i} \subseteq S_{v_i}} \min_{\forall v_k \in \bar{S}_{v_i}} \lfloor \frac{er_{v_k}}{cost(v_k, \bar{S}_{v_i})} \rfloor$.

When encoding $\bar{C}_V$ in Eq. 13, smaller rep_i is always more favorable from computational point of view. The rep_i shown in Eq. 15 can be further decreased by considering the fact that $\bar{S}_{v_i}$ should be a SHM cover set (we will show soon that any `str` that generates $\bar{S}_{v_i}$ which is not a SHM cover set will be fixed). Obviously, by considering the constraint, the $\bar{S}_{v_i}$ in Eq. 15 should include more nodes and the corresponding rep_i will be further decreased. Here, we use a greedy method to find out the $\bar{S}_{v_i}$ in Eq. 15. Initially, $\bar{S}_{v_i} = v_i$. Then the v_i's neighbors are added one by one, each time the one which is able to minimize $cond(\bar{S}_{v_i})$ is added. The procedure iterates until $\bar{S}_{v_i}$ is a SHM cover set. The corresponding rep_i is then used in Eq. 13.

We note that the encoding method described above might represent an infeasible solution to the MSSP. A solution is infeasible when one or both of the following conditions occur:

$$\begin{aligned} &(1)\ \exists S_k^{str} \in \bar{C}_V^{str}, |S_k^{str}| < p_{mod} \text{ or } cond(S_k^{str}) > \gamma \\ &(2)\ \exists v_i \in V, \sum_{j=1}^{p} cost(v_i, S_j^{str}) > er_i \end{aligned} \quad (16)$$

To deal with infeasible solutions in GAs, we designed a repair operator to convert an infeasible solution into a feasible MSSP solution. This operator includes two stages and in each stage, one condition shown in Eq. 16 is considered. Given $\bar{C}_V = \{S_{v_1}, \cdots, S_{v_1}, \cdots, S_{v_m}, \cdots, S_{v_m}\}$, a binary string `str` and the corresponding group of sensor sets $\bar{C}_V^{str} = \{S_1^{str}, S_2^{str}, \cdots\ S_q^{str}\}$ by mapping `str` to $\bar{C}_V$, the repair operator first evaluates, one by one, all sensor sets in $\bar{C}_V^{str}$ to see whether they are SHM cover sets. For any S_i^{str} with $|S_i^{str}| < p_{mod}$ or $cond(S_i^{str}) > \gamma$, the operator finds out the sensor set in $\bar{C}_V$, denoted as S_{v_l}, from which the S_i^{str} is extracted. Then it randomly selects a $'0'$ from the segment of `str` corresponding to S_{v_l} and sets it to $'1'$. This procedure above corresponds to adding one neighbor of v_i which was not selected into S_i^{str}. The S_i^{str} is then updated and the $cond(S_i^{str})$ is re-evaluated. This procedure re-iterates until one of the two conditions is valid (1) $|S_i^{str}| \geq p_{mod}$ and $cond(S_i^{str}) < \gamma$ (2) $S_i^{str} = S_{v_l}$. The occurrence of the second condition indicates that even when all the one-hop neighbors of v_l have been used, S_i^{str} is still not a SHM cover set. If this is the case, the segment of `str` corresponding to S_{v_l} will be cleared and the S_i^{str} will be deleted from $\bar{C}_V^{str}$.

After the repair operator goes through all the sensor sets in $\bar{C}_V^{str}$, the remaining sensor sets in $\bar{C}_V^{str}$ are all SHM cover sets. The repair operator then seeks the most energy consuming node across the sensor sets in $\bar{C}_V^{str}$ and see whether its energy consumption exceeds the energy threshold. If it is true, the operator will delete, from $\bar{C}_V^{str}$, one sensor set in which the energy consumption of this node is highest in $\bar{C}_V^{str}$. This procedure re-iterates until the energy consumption constraint is satisfied for all the sensor nodes.

The detailed procedure of this two-stage repair operator is illustrated in Algorithm 1. Algorithm 1 is guaranteed to always produce a feasible solution for the MSSP, irrespective of the initial binary string `str`.

The fitness function of a `str` updated from Algorithm 1 is the number of sensor sets in the corresponding $\bar{C}_V^{str}$. The

Algorithm 1 Repair operator for the GA

Input: $\bar{C}_V$, `str`, and $\bar{C}_V^{str} = \{S_1^{str}, S_2^{str}, \cdots, S_q^{str}\}$

for i = 1 to q **do**
  Find out the sensor set in $\bar{C}_V$ from which S_i^{str} is extracted. This set is denoted as S_{v_l}
  while ($|S_i^{str}| < p_{mod}$ or $cond(S_i^{str}) > \gamma$) and $S_I^{str} \subset S_{v_l}$ **do**
    Randomly select a $'0'$ from the segment of `str` corresponding to S_{v_l} and set it to $'1'$
    Update S_i^{str}, `str` and $\bar{C}_V^{str}$
  end while
  if $|S_i^{str}| < p_{mod}$ or $cond(S_i^{str}) > \gamma$ **then**
    Delete S_i^{str} from $\bar{C}_V^{str}$
    Clear the segment of `str` corresponding to S_{v_l}
  end if
end for
Find out the node with the highest energy consumption across all sets in $\bar{C}_V^{str}$. This node is denoted as v_{max}.
while $\sum_{S_i^{str} \in \bar{C}_V^{str}} cost(v_{max}, S_i^{str}) > er_{v_{max}}$ **do**
  Delete the sensor set in which the energy consumption of v_{max} is highest among all the sensor set in $\bar{C}_V^{str}$
  Update $\bar{C}_V^{str}$ and `str`
  Find out the v_{max} in the updated $\bar{C}_V^{str}$.
end while

Output: The updated `str` and $\bar{C}_V^{str}$

larger the number of sensor sets in $\bar{C}_V^{str}$, the higher the fitness of the corresponding `str`.

Having decided on a representation, the repair operator and the fitness function, now we can start the evolution process. We first generate, at random, a population of binary strings (i.e. the genes), and then feed this population into the repair operator. From the output of the repair operator, each updated gene is evaluated by the fitness function. The the fitter genes will be used for mating and crossover to create the next generation of genes. A gene having the maximum fitness value among a population is called elite and carried through unchanged to the next generation. The iteration stops if there is no improvement in the maximum fitness value for ten consecutive generations.

VI. Validation of Proposed Method

A. Simulation

To test the effectiveness of the proposed methods, a simulated suspension bridge shown in Fig. 4(a) is generated by SAP2000.

A total of 60 wireless sensor nodes are used to measure the vibration at the transverse direction (z direction in Fig.4(a)) of the deck of the bridge. The locations of these nodes are shown in Fig. 4(b). These 60 locations are selected using domain knowledge from civil engineering.

The theoretical first 4 modes of the structure are illustrated in Fig. 4 (c). The mode shapes will be used to calculate the condition number of different subsets in the proposed methods and the natural frequencies are used as reference for validation

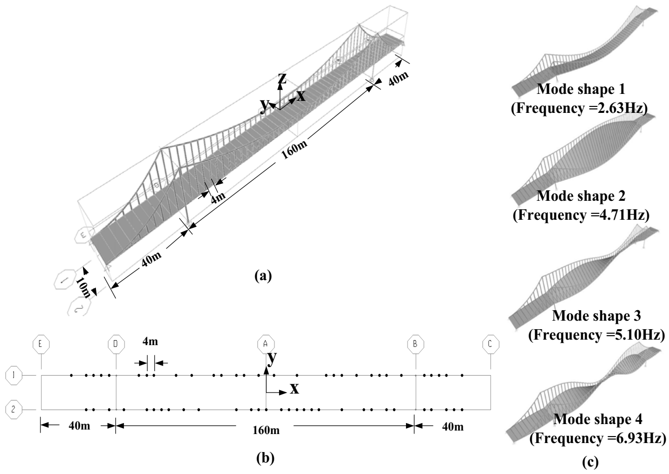

Fig. 4: The simulated bridge (a) The dimensions of the suspension bridge (3D), (b) the sensor locations (X-Y plane) (c) The first four theoretical modes of the bridge

N	er (mAh)	e_S (mAh)	e_R (mAh)	e_T (mAh)	e_{ERA} (mAh)
20480	700	1.1e-4	5e-4	5e-4	$0.0417(0.4\|V\|^2 + 1.2\|V\| - 3.6$

TABLE I: Parameters used in the simulation

of the accuracy of SHM cover sets. The parameters used in the simulation are listed in Table I. Some of the parameters, such as e_S, e_R, e_T and e_{ERA}, are obtained by some real tests on our own designed SHM motes which will be described in detail in the next section. In this simulation, it is assumed that all the sensor nodes use the same battery with $er = 700$mAh. We require that the first 4 natural frequencies need to be identified and the condition number of each SHM cover set should be less than $\gamma = 1000$ so that the identified natural frequencies can be accurate enough to detect a certain level of damage on this bridge. Note that γ is different for different structures, and choosing $\gamma = 1000$ will be justified later in this section.

Given the parameters shown in Table I, the bridge model and the locations of these 60 nodes shown in Fig. 4, the total number of rounds that the system can work is dependent on the network topology. Intuitively, a dense network in which each node has a large number of neighbors can lead to a system with long lifetime. Here, we use the total number of links in the network to measure the network density. In the simulation, we fix the parameters shown in Table I but change the network density by adjusting the communication range of each node C_r.

Fig. 5 illustrates the results using the heuristic methods when C_r = 12 m (the corresponding number of links is 142). From Fig. 5 (a), it can be seen that the heuristic method generates 19 sensor sets, and the number of working rounds assigned to each sensor set range from 1 to 20. The total system lifetime becomes 190. Also, although not shown in the figure, the number of sensor nodes in each sensor set ranges from 4 to 5. From Fig. 5 (b) and (c), we can see that all the

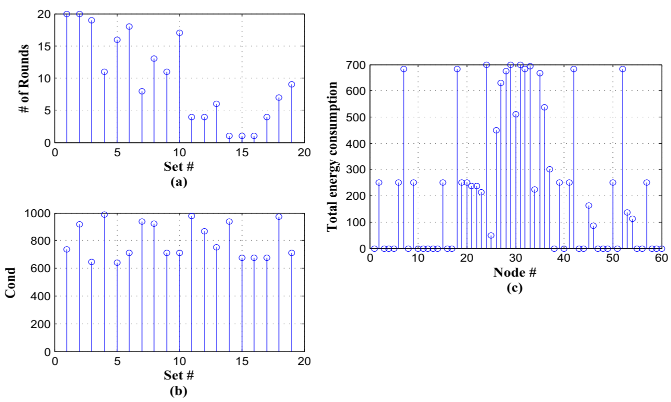

Fig. 5: Results from the heuristic method when C_r = 12m (a) The SHM cover sets and the number of rounds assigned (b) The condition number of each SHM cover set (c) The energy consumed of each sensor node across all SHM cover sets

sensor sets are SHM cover sets (condition number $\leq$ 1000), and the energy consumed of each node is below 700 mAh. An interesting thing that can found in Fig. 5 (c) is that some nodes are not selected in any SHM cover sets. This is the result of both network topology and node location. Note that these sensor nodes are only 'redundant' at the first stage of the multi-tired strategy but still useful at the second stage when the mode shapes are identified.

With the increase of the network density, the system lifetime can be expected to increase since more candidate sensor sets can be generated. As an illustration, Fig. 6 shows the results when C_r = 19 m. In this condition, 246 sensor sets are generated and the system lifetime is 393. Also can be seen from Fig. 6 (c) is that compared with Fig. 5 (c), there is no un-selected sensor nodes. A dense WSN obviously can increase the utility of individual nodes when they form into sensor sets to detect structural damage. As a summary, Fig. 7(a) illustrates the total number of rounds that the system can work when the number of links in the network is increasing from 142 $\rightarrow$ 175 $\rightarrow$ 238 $\rightarrow$ 260 (the corresponding C_r are 12 m 14 m, 16 m and 19 m, respectively).

For comparison, Fig. 7 (a) also shows the number of rounds that the system can work using the SHM cover sets from the GA method. It can be seen that compared with the heuristic method, the system lifetime of the GA method is slightly lower. However, the advantage of the GA over the heuristic method lies in the computation time. For the heuristic method, the computation time increases significantly with the increase of the C_r (i.e. network density). This is illustrated in Fig. 7 (b). The reason can be attributed to the fact that in the worst case, the number of candidate sensor sets in $\mathbb{S}_{v_i}$ of Eq. 6 is with the level of $O(n!)$ where n is the number of nodes in S_{v_i}. Consequently, when using the cross-entropy optimization to solve the transformed Knapsack problem, the computational complexity increases dramatically with the increase of the network density (i.e. the number of links).

However for the GA method, the length of the binary string

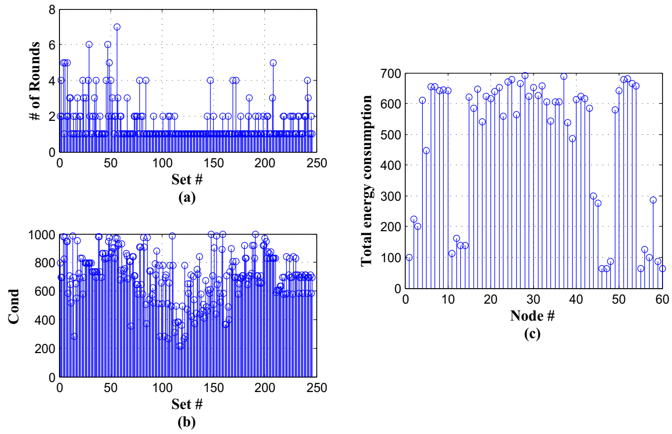

Fig. 6: Results from the heuristic method when C_r = 19m

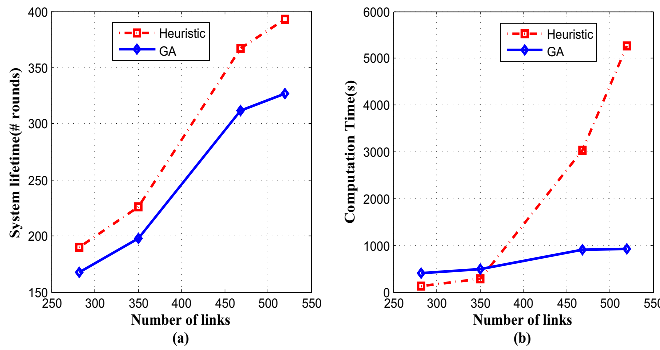

Fig. 7: Compare the results from the heuristic and the GA method in terms of (a) lifetime, and (b) the computation time

str, which highly affects the complexity of the GA algorithm, is only $O(m^2)$ where m is the number of the nodes in the network. Hence the advantage of the GA over the heuristic one in terms of computation time is more obvious in a dense WSN with large number of nodes. This is demonstrated in Fig. 7 (b), where the computation time of the two methods are compared in different network density. Fig. 7 (b) shows that the computation time of the GA becomes lower than the heuristic one when the number of links exceeds 350, and this gap becomes more obvious with the increase of network density.

Now we discuss the importance of controlling the condition number of each sensor set. Fig. 8 (a) shows the condition when the deployed 60 sensors are divided quite arbitrarily into 12 disjoint sensor sets, each containing exact 5 nodes, which is the upper bound of the number of nodes in the SHM cover sets identified in Fig. 5. It can be seen from Fig. 8 (b) that without specific control on the generated sensor sets, the condition number of a sensor set can easily exceed the required threshold. As an example, only #2 has condition number below 1000 and the condition numbers #1,#3,#11 even exceed 10^{15}.

To complete simulation part, we demonstrate how the condition number will affect the accuracy of identified natural

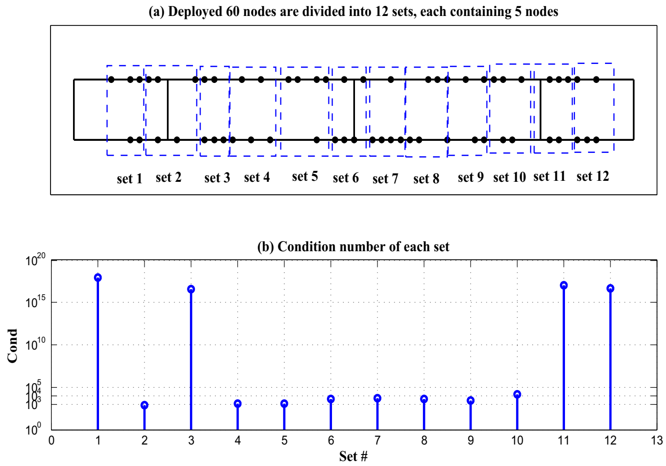

Fig. 8: (a) The deployed 60 sensors are divided into 12 disjoint sensor sets (b) the condition number of each sensor set

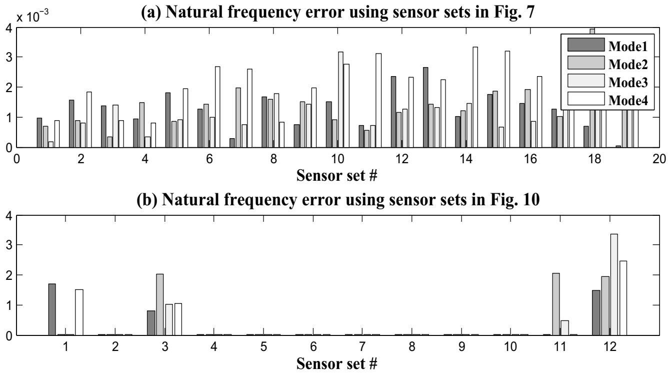

Fig. 9: The identified natural frequency error using the sensor sets in (a) Fig. 5 and in (b) Fig. 8

frequencies. Impulse responses of the simulated suspension bridge are generated at these 60 sensor nodes. The response time series were sampled at 200 Hz. Noise was added to the sensor data at each sample as a zero-mean Gaussian sequence with variance σ^2. In this simulation, σ^2 is chosen such that the ratio of the σ to the root-mean-square sensor output averaged over all the 60 sensors is 15%.

The error of the i^{th} identified natural frequencies, denoted as $\tilde{f}(i)$, is calculated as

$$\tilde{f}(i) = \left|\hat{f}^i - f^i\right| / f^i \tag{17}$$

where $\hat{f}^i$ is the identified i^{th} natural frequency and the f^i is the true one.

Fig. 9 (a) illustrates the identified natural frequency error using the SHM cover sets obtained from the heuristic method shown in Fig. 5. The natural frequency error of all the SHM cover sets in all of the four modes are below $5e-3$. It can be seen that even with the relatively high noise-to-signal ratio (15% in this case), the natural frequencies of the bridge are very accurately identified using data from each cover set.

For comparison, the natural frequency error using the sensor sets in Fig. 8 are presented in Fig. 9 (b). It can be seen that the natural frequencies identified from sensor sets with large condition number are not accurate enough to detect damage.

VII. Conclusion

In this paper, we re-define the 'coverage' and based on this generalized coverage model, two methods are proposed to support EECPS in SHM. SHM is not the only application where coverage area of individual sensor nodes cannot be defined explicitly. Besides SHM and Volcano monitoring, another application can be found in social management since the performance of a team cannot be determined by adding that of the individual's together. The partition methods proposed in this paper can be applied to these applications. We only need to have a function which is able to evaluate whether a given set of resources is able to fulfill a pre-defined task (as Eq. 2).

VIII. Acknowledgement

This research is financially supported in part under Hong Kong RGC Joint Research Schemes 3-ZG1F, the NSF of China with Grant 61332004, 61272053 and Hong Kong Scholars Program.

References

[1] A. Mainwaring and et al., "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM, 2002, pp. 88–97.

[2] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 472–476.

[3] S. Gao and et al., "Sensor scheduling for k-coverage in wireless sensor networks," *Mobile Ad-hoc and Sensor Networks*, pp. 268–280, 2006.

[4] M. Cardei, M. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *INFOCOM 2005. Proceedings IEEE*, vol. 3. IEEE, 2005, pp. 1976–1984.

[5] X. Wang and et al., "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st international conference on ENSS*. ACM, 2003, pp. 28–39.

[6] D. Tian and N. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of the 1st ACM workshop on WSNs and applications*, 2002, pp. 32–41.

[7] H. Zhang and J. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Urbana*, vol. 51, p. 61801.

[8] C. Farrar and K. Worden, "An introduction to SHM," *Philosophical Transactions of the Royal Society*, vol. 365, no. 1851, p. 303, 2007.

[9] J. Juang and R. Pappa, "An eigensystem realization algorithm for modal parameter identification and model reduction," *Journal of Guidance*, vol. 8, no. 5, pp. 620–627, 1985.

[10] L. Yao, W. Sethares, and D. Kammer, "Sensor placement for on-orbit modal identification via a genetic algorithm," *AIAA journal*, vol. 31, no. 10, pp. 1922–1928, 1993.

[11] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman & Co., 1979.

[12] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.

[13] Z. Botev and D. Kroese, "An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting," *Methodology and Computing in Applied Probability*, vol. 10, pp. 471–505, 2008.

[14] D. Goldberg, "Genetic algorithms in search, optimization, and machine learning," 1989.