A Pso-Based Maintenance Strategy in Wireless Sensor Networks

Long Cheng, Yan Wang, Chengdong Wu & Quancheng Han


To link to this article: http://dx.doi.org/10.1080/10798587.2014.919716

Published online: 01 Jul 2014.

Submit your article to this journal

Article views: 88

View related articles

View Crossmark data
A PSO-BASED MAINTENANCE STRATEGY IN WIRELESS SENSOR NETWORKS

LONG CHENG1, YAN WANG1, CHENGDONG WU2, QUANCHENG HAN2
1Department of Computer and Communication Engineering, Northeastern University, Qinhuangdao, P.R. China
2Information Science and Engineering College, Northeastern University, Shenyang, P.R. China

ABSTRACT—This paper presents a new wireless sensor networks maintenance strategy using particle swarm optimization. This strategy considers three factors: Coverage rate, node energy consumption and node residual energy. The advantage of this method is to prolong the maintenance period and reduce the computational complexity. We first construct the networks health indicator to determine the locations of redeployed nodes. Therefore, the maintenance problem is formulated into cost optimization problem, and the linearly decreasing weight particle swarm optimization is employed to reduce the computational complexity. Simulation results show that the proposed particle swarm optimization based maintenance strategy (PSOMS) outperforms the random and uniform redeploy strategy with longer repair period.

Key Words: wireless sensor networks; networks maintenance; particle swarm optimization; coverage hole

1. INTRODUCTION

Wireless sensor networks (WSNs) have various applications in health surveillance, battle field and environmental monitoring. The WSNs comprises hundreds of sensor nodes with microprocessor, radio, and a limited amount of storage [1]. The tiny nodes are deployed in a particular area to measure certain phenomenon. The main objective of WSNs is to sense the environment and collect the information to base station. In order to guarantee the quality of service, it is important to maintain the density of nodes. We also deploy many redundant nodes in the networks. However, no matter how many redundant sensors are deployed, they can only tolerate failures up to a certain lifetime level [2]. The failed nodes may lead to the loss of information and networks disconnected. In the worst case, the networks may get partitioned into several disjoint blocks. So, it is necessary to design appropriate maintenance scheme to keep desired quality of deployment.

Many works have been proposed to maintenance the networks. The mobility assisted and the robot assisted maintenance schemes are the two main directions. As a mobility assisted maintenance scheme, the authors use the mobile nodes to redeploy the networks. Thus, all or parts of the nodes are assumed to have mobility. A novel coverage conscious connectivity restoration algorithm [3] is proposed. This is a distributed method. The failure recovery task lies with only neighbors of the failed nodes. These neighbors coordinate among themselves and agree on their role in the recovery. Each node involved in the recovery will move to the position of the failed node to restore connectivity and coverage in that area and then go back to its original position after serving for some time. The neighbors of the failed node take turns. Energy-balance repair schemes are proposed to extend the maintenance period [4–5], but these methods focus on mobile sensor networks. Some or all sensor nodes are assumed to have mobility in mobile sensor networks.
networks. Although sensor mobility has exploited to boost the performance of WSNs, mobility is an expensive feature because the mobile nodes need to have motor and GPS modules. So mobility is not feasible for large-scale networks.

In robot assisted maintenance strategy, the robot serves as the maintainer of the networks. The robot assisted strategy is simple and the cost of implementation is much cheaper than mobility assisted strategy. Therefore, this strategy requires relatively low hardware configuration. Corke et al. [6] developed a connectivity test algorithm that allows UAV to obtain the connectivity graph, and then the UAV compares the connectivity graph against the desired topology embedding using graph isomorphism algorithms. The UAV can collect the topology information and determine how many separation regions there are. Finally, the controller obtains the position of the UAV from GPS and navigates the UAV to the desired way-points to repair the networks. Suzuki et al. [7] proposed an autonomous deployment and restoration using mobile robots in WSNs. The mobile robot carries nodes and deploys them into the environment while measuring received signal strength indicator values to ensure communication. The robot restores the networks by deploying additional or alternate nodes to the field when the networks are disrupted. Mei et al. [8] use a small number of mobile robots to replace failed sensors. This scheme sets up the guardian-guardian relationship among sensor nodes in initialization step, and then three robot coordination algorithms are proposed to repair the networks. This scheme needs the nodes communication with each other frequently in order to sustain the guardian-guardian relationship, so this scheme will consume more energy. Falcon et al. [9] model the robot-assisted coverage problem as one-commodity traveling salesman problem. The above methods are focus on coverage hole, but when networks run a long time, the residual energy is different in nodes due to the channel collision. Since nodes with less residual energy will deplete their energy rapidly, the networks will generate a new hole. The maintenance operation will implement in a very short period of time, so it will decrease the maintenance period. The above works have focus on coverage maintenance, but most articles neglect the residual energy in each node. Some nodes will die soon since their residual energy is less, so the maintenance operation will implement in a very short period of time. It will increase the maintenance cost.

Here, a particle swarm optimization based maintenance strategy (PSOMS) is proposed in this paper. It is a simple and efficient approach to deal with the networks maintenance for robot assisted strategy. We consider the coverage rate, node consumption energy and node residual energy in the maintenance strategy, and the networks maintenance problem is formulated into cost optimization problem. In order to reduce the computational complexity, we employ the linearly decreasing weight particle swarm optimization.

The rest of the paper is organized as follows: In Section 2 we describe the system model. In Section 3 we will introduce our maintenance strategy. Some simulation results will present in Section 4. The conclusions are given in Section 5.

2. SYSTEM ARCHITECTURE DESCRIPTION

We consider the WSNs that are randomly distributed into a 2-dimensional field for long-term surveillance. The system architecture is shown in Figure 1; the system consists of a networks maintenance robot, a base station, and the WSNs composed by groups of sensors. The sensor networks are based on a ZigBee communication, and all the communication is complied with the ZigBee protocol (IEEE 802.15.4).

We make the following assumptions in this study: 1) The location of node is known by itself. 2) The node reports its data to base station with fixed period through multi-hop communication, and the data consists of node_id, node’s location, sensing data, residual energy, so the base station knows the factors of node. 3) The location of replacement nodes (candidate nodes) are determined by base station. When the base station determines where to replace new nodes, the robot will go to those determined locations and deploy new nodes. 4) The maintenance operation runs round by round. In each round, the base station
computes the area of the coverage hole. When the base station discovers the area of coverage hole takes up 10% of the sensory field, the robot will repair the networks. The robot located at the base station should go back to the base station when the repairs are finished. The maintenance period is defined as: 
\[ MP = T_{i+1} - T_i \]
where \( T_{i+1} \) is the maintenance time at round \( i + 1 \), and \( T_i \) is the maintenance time at round \( i \).

For ZigBee networks, there are three types of networks device: The end nodes, the router nodes and the coordinator node. The functions of these nodes are described as following [10–11]:

End node: It can only collect data values from sensors, insert the values into packets and transfers the data of itself to the parent node. It has no routing capability, and it relies on its parents to transmit or route its packet.

Router node: The function of the router is to find the best route to the destination over to transfer a message, in addition to performing the sensing function.

Coordinator node: There is only one coordinator in the networks. It is the controller of the networks and it is responsible for initiating the networks setup, and it also stores information about the networks, including a list of neighbors and routers. The role of networks coordinator is typically given to the sink node.

As shown in Figure 2, the networks maintenance robot carries STM32F103 board, ZigBee node, CCD camera module, GPS module and mechanical arm. This robot is a six-wheel drive and independent
suspension vehicle, so it can adopt the complex environment. The main processor is STM32F103 with the ARM kernel. The real-time operating system uC/OS-II is adopted in the software platform. The STM32F103 board is the main control board. It controls the mechanical arm to pick up the new nodes and deploy them on the determined locations. The GPS module provides the location information for robot. However, when an obstacle exists between in satellite and GPS receiver in harsh environment, the GPS measurement accuracy may significantly degrade. The robot could communicate with the beacon nodes in the sensor networks to improve the location accuracy, and the CCD camera module can record the maintenance process.

3. SIGNAL MODEL

3.1 Sensor Coverage model

The most common models in WSNs are binary detection model and Elfes’s model [12]. The binary detection model assumes that an event occurs within the sensing radius of a sensor would be detected with probability 1, while an event outside this circle of influence would not be detected with probability 0. The binary model is commonly adopted for its analytical simplicity, but it is based on unrealistic assumption of perfect coverage for sensors. The Elfes’s is more realistic than binary model. Both of the above models do not incorporate the false characteristics.

We introduce the NP detection model according to Neyman-Pearson criterion in this section. The Neyman-Pearson (NP) model [13] incorporates both the signal characteristics and false alarm rate in the model. According to the Neyman-Pearson criterion, the detection probability $P_D$ is given by

$$P_D = 1 - \Phi \left( \Phi^{-1}(1 - \alpha) - \frac{\beta}{\sigma} \sqrt{\frac{1}{R_{ij}^2}} \right)$$  \hspace{1cm} (1)$$

Where $\alpha$ is the false alarm probability $\alpha = P_F$, $R_{ij}$ is the distance between target $i$ and sensor $j$, $\beta$ is a scalar. $\sigma$ is the standard variance of noise. $\gamma$ is the path loss coefficient. $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function, i.e.,

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{y^2}{2} \right) dy$$

A target may be detected by many sensors within the sensing radius, so the detection probability of target $i$ is

$$\tilde{P}_i = 1 - \prod_{j=1}^{K} (1 - P_{ij})$$  \hspace{1cm} (2)$$

Where $K$ is the number of nodes which fall in the circle with its center at the target $i$ and of radius $r_s$ ($r_s$ is the sensing range of a sensor node). $P_{ij}$ is the detection probability of target $i$ by sensor $j$. Obviously, if the number of deployed nodes is large enough, the detection probability will be 1, i.e. $K \to \infty, \tilde{P}_i = 1$.

3.2 Sensor Energy Consumption Model

In this section, we present an analytical model for deriving the energy consumed by a node. This model takes into retransmission times for frames and successful transmission probability. The amount of energy
consumed by a node for the transmission of a data packet is given by [14]

\[
E_{tx} = \frac{E_{t0} + (\beta/\eta_{amp})d^2}{1 - l_f} b_d
\]  

(3)

Where \(E_{t0}\) is the energy consumed by non-transmitted device, which include frequency synthesizer, mixers, and filters. \(\eta_{amp}\) is the amplifier of transmitter. \(\beta\) is the constant of amplifier. \(l_f\) is the loss rate of query and reply frames. \(b_d\) is the length of data frames. \(d\) is the communication range.

The energy consumption which transmits and receives a data packet in one-hop is

\[
E_{1\text{hop}} = \frac{E_{rx}}{1 - l_f} b_c + \frac{E_{t0} + (\beta/\eta_{amp})d^2}{1 - l_f} b_d
\]  

(4)

We assume that \(E_{rx}\) is the amount of energy consumed for transmitting a single bit by receiver. \(b_c\) is the length of control frames such RTS, CTS and ACK.

In this paper we assume that the networks using CSMA based media access control protocol. The transmission probability of success in \(r\)-th slot can be expressed as:

\[
P_{\text{success}} = N \cdot P_r \left[ 1 - \sum_{i=1}^{r} P_i^{\text{success}} \right]^{N - 1}
\]  

(5)

Where \(P_i^{\text{success}}\) denotes the transmission probability of \(i\)-th slot being selected, \(N\) denotes the number of contenders. \(P_r\) is the transmission probability of the node in \(r\)-th time slot.

So the expectation of energy consumption in CSMA networks can be expressed as:

\[
E_{\text{CSMA}} = E_{1\text{hop}} + \left(1/P_{\text{success}} - 1\right)E_{\text{busy}}
\]  

(6)

Where \(E_{\text{busy}}\) is the energy consumption during backoff period.

4. MAINTENANCE STRATEGY

The goal of the maintenance strategy is to determine the locations of redeployed nodes. The sensor field is made up of grid points, e.g. a 200 × 100 field with 40 × 20 cells and a grid resolution \(k = 5\). We assume that the size of the sensor field is \(n \times m\). There are \(h\) sensors in the networks. The node set is \(NS = [node_1, node_2, \ldots, node_h]\). The field is divided into \(n \times m\) grid points (grid resolution \(k = 1\)), so we can obtain a total of \(nm\) grid points. The networks health indicator is given by:

\[
\text{AveCost} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} C_{ij} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\bar{P}_{ij} + E_{ij}\right)
\]  

(7)

\(\bar{P}_{ij} = 1 - \Pi_{k=1}^{K} (1 - P_{ijk})\) \((d_{ijk} \leq r_s, K \leq h)\). \(\bar{P}_{ij}\) is the combined detection probability of grid point \((i,j)\). \(P_{ijk}\) is the detection probability of grid point \((i,j)\) by \(k\)-th node. \(d_{ijk}\) is the distance between grid point \((i,j)\) and \(k\)-th node. \(r_s\) is the sensing range of a sensor node.

\(E_{ij} = \sum_{k=1}^{K} w_{ijk} (1 - E_{ij}^{\text{CSMA}})\) \((d_{ijk} \leq r_s)\). \(E_{ij}\) is responsible for energy distribution. \(w_{ijk} = (1/E_{ij}^{\text{res}})/\sum_{k=1}^{K} (1/E_{ij}^{\text{res}})\) is the weight of each node. \(E_{ij}^{\text{CSMA}}\) is the expectation of energy consumption between grid point \((i,j)\) and \(k\)-th node. \(E_{ij}^{\text{res}}\) is the residual energy in \(k\)-th node. The value of \(E_{ij}\) is low if the grid point \((i,j)\) around it have low residual energy.

We attempt to find the locations, which can maximum the average cost as the locations of candidate nodes. A straightforward method is exhaustive search. However, the computational cost is extremely high. For example, let there be \(n \times m\) grid points (grid resolution \(k = 1\)) to be searched. Then the total number of searching points will be equal to \(n \times m\). When the field is divided into finer grids (grid resolution \(k = 0.5\)), the number of searching points will be equal to \(2n \times 2m\). While the computation complexity may be
feasible for the desktop computer, it is likely to be excessive for low power sensor nodes with limited computing capabilities.

Particle swarm optimization (PSO) is a population-based heuristic optimization technique. It has been widely used in parameter estimation and localization algorithm [15]. The rapidity growing computation complexity can be mitigated with the use of PSO. Unfortunately, conventional PSO suffers from the premature convergence problem, especially in complex problems since the interactive information among particles in PSO is too simple to encourage a global search [16]. In this paper, we employ the linearly decreasing weight particle swarm optimization (LinWPSO) method [17] to select the locations of candidate nodes.

We assume that a swarm consists of 20 particles \( S = \{X_1, \ldots, X_{20}\} \). The search space is 2 dimensional vector, and then the particle \( i \) of the swarm can be represented by 2 dimensional vector \( X_i = (x_{i1}, x_{i2}) \), and the position of the particle represents a location of candidate node. In search process, the particle tunes its current position toward the global optimum. At step \( t \), the velocity of particle \( i \) and its new position will be assigned according to the following two equations:

\[
v_i^t = \omega v_i^{t-1} + c_1 \xi (pbest_i^{t-1} - X_i^{t-1}) + c_2 \eta (gbest^{t-1} - X_i^{t-1})
\]

\[X_i^t = X_i^{t-1} + v_i^t\]  

Where, \( v_i^t \) stands for the velocity of particle \( i \) at step \( t \). \( X_i^t \) is the position of the \( i \)-th particle. \( \omega \) is the inertia weight to balance exploitation and exploration, where \( \omega_{\text{max}} = 0.9, \omega_{\text{min}} = 0.4 \). \( pbest_i \) represents the best location in the search space ever visited by particle \( i \) and \( gbest \) is the best location discovered so far. \( c_1 \) and \( c_2 \) are two acceleration constants, where \( c_1 = c_2 = 2 \). \( \xi \) and \( \eta \) are two uniform random numbers in \([0,1]\).

The inertia weight for \( t \) iteration is used to balance the global and local search abilities of PSO, and it plays an important role in PSO. The large inertia weight enables global search and a small inertia weight facilitates local search. So we employ the linearly decreasing weight particle swarm optimization to improve the search ability of the PSO. The inertia weight for \( t \) iteration is given by:

\[
\omega = \omega_{\text{max}} - \frac{t(\omega_{\text{max}} - \omega_{\text{min}})}{t_{\text{max}}}
\]

The maximum number of iterations \( t_{\text{max}} = 80 \). The pseudocode of the proposed strategy is shown as follows:

1. Initialization parameters: deployde_num = 0, node_limit = 70, cost_threshold = 1.97.
2. The field is divided into grid. The number of grid points is \( nm \).
3. While deployde_num < node_limit and ave_ cost < cost_threshold
4. While \( t < t_{\text{max}} \)
5. Initial population of particles with random positions and velocities.
6. For each particle \( i = 1, \ldots, 20 \)
7. Add the particle \( i \) as a virtual node into the node set.
8. Compute the average cost value in the grid according new node set: \( \text{vir}_\text{ave}_\text{cost} \).
9. If AveCost(X_i) > AveCost (pbest) then pbest = X_i,
10. If AveCost (X_i) > AveCost (gbest) then gbest = X_i.
11. Remove the virtual node from the node set.
12. End for
13. Update the velocity and position of particle \( i \) using Equation (8) and (9) respectively.
14. Update the inertia weight for \( t \) iteration using Equation (10).
15. Increment \( t \).
16. End while
17. The \( g_{\text{best}} \) is the location of candidate node. Add the candidate node into the node set.
18. Increment \( \text{deployed}_{\text{num}} \).
19. End while

5. SIMULATION

In this section, some simulation results will be presented to show the performance of our proposed particle swarm optimization based maintenance strategy (PSOMS). In this simulation, 120 randomly nodes are deployed in a region of size 100 m \( \times \) 100 m. The residual energy of each node is different from 0 to 3000 Joules. When the area of coverage hole takes up 10% of the sensory field, the robot will repair the networks. The values of the parameters in the coverage and energy consumption model are shown in Table I.

We perform the comparative performance analysis of the proposed strategy and the existing schemes (random and uniform redeploy strategy). Figure 3 shows the relation between number of candidate nodes and indicator values. All the values increase with the number of candidate nodes increasing, and the LinwPSO outperforms the exhaustive search method (grid resolution \( k = 1 \)). For exhaustive method, we assume that the grid resolution \( k = 1 \), then the number of search points will be equal to 10,000. By using the LinwPSO method, the number of search points will be equal to 1600. So the computational complexity has reduced 84%.

Figure 4 shows the relation between coverage rate and the number of deployed nodes. We can see that the coverage rate of all algorithms increase with the number of candidate nodes increasing. The PSOMS has the highest coverage rate. And PSOMS has much higher coverage rate than random and uniform redeploy algorithms, about 7.46% and 5.48% respectively.

In Figure 5 we study the impact of the number of candidate nodes on the maintenance period. It can be observed that the maintenance period increases as the number of redeployed nodes increase, and the PSOMS presents the highest maintenance period. The proposed algorithm can achieve significantly longer maintenance period than random and uniform algorithms, about 106.86% and 60.14% respectively.

The computational complexity of LinWPSO can be represented as \( O(t_{\text{max}} \cdot NP) \). The computational complexity will linearly increase with the increase of the number of iteration and particles, so the the number of iteration and particles are key parameters for LinWPSO. Figure 6 shows the evaluation for the convergence rate by changing the parameters. Obviously, the convergence of LinWPSO algorithm occurs when the number of iteration is 60, but the LinWPSO shows poor performance when the number of particles is 10, and the LinWPSO shows similar performance when the number of particles is 20 and 30.

Figure 7 shows the proposed algorithm outperform the random and uniform redeploy algorithm when the grid resolution is relatively lower. However, the maintenance period of PSOMS decreases with the grid resolution increase.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_{\text{in}} )</td>
<td>3.63( \mu )J/bit</td>
<td>( E_{\text{rx}} )</td>
<td>11.13( \mu )J/bit</td>
</tr>
<tr>
<td>( l_r )</td>
<td>0.3</td>
<td>( E_{\text{busy}} )</td>
<td>1.2</td>
</tr>
<tr>
<td>( P_{\text{success}} )</td>
<td>0.8</td>
<td>( b_\gamma )</td>
<td>19 bit</td>
</tr>
<tr>
<td>( b_d )</td>
<td>133bit</td>
<td>( \gamma )</td>
<td>2</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>0.05</td>
<td>( \beta )</td>
<td>50</td>
</tr>
<tr>
<td>( \sigma )</td>
<td>0.25</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>
Figure 3. Number of Deployed Nodes Versus Indicator Values.

Figure 4. Number of Deployed Nodes Versus Coverage Rate.

Figure 5. Number of Deployed Nodes Versus Maintenance Period.
6. CONCLUSION

In this paper, we proposed a particle swarm optimization based maintenance strategy (PSOMS). This strategy incorporated three factors (coverage rate, node residual energy and node consumption energy). We first introduced the sensor coverage model and energy consumption model, then established the maintenance cost function. The linearly decreasing weight particle swarm optimization was employed to find optimal estimation. Simulation results showed that the PSOMS could achieve relatively higher coverage rate while achieving much longer maintenance period than random and uniform redeploy algorithms, and the computational complexity was reduced in comparison with exhaustive method.

ACKNOWLEDGEMENTS

This work was supported in part by "National Natural Science Foundation of China" [grant number 61403068, 61203216, 61273078], "Fundamental Research Fund for the Central Universities of China" [grant number N130323004, N130323002],
REFERENCES


NOTES ON CONTRIBUTORS

L. Cheng received B.S. degree in Automatic Control from Qingdao University of Science and Technology, Qingdao, China, in 2008, M.S. degree in Pattern Recognition and Intelligent Systems from Northeastern University, Shenyang, China, in 2010. He received Ph.D. degree in Pattern Recognition and Intelligent system from Northeastern University, Shenyang, China, in 2013. His research interests include wireless sensor networks and distributed system. He is a student member of the IEEE and IEICE.
Y. Wang received M.S. and Ph.D. degree from Northeastern University, Shenyang, China, in 2010 and 2013 respectively. Her research interests include routing and localization algorithm in wireless sensor network.

C. D. Wu received M.S. degree in Automatic Control from Tsinghua University, Beijing, China, in 1988. He received Ph.D. degree in Automatic Control from Northeastern University, Shenyang, China, in 1994. From 1995 to 1997, he worked as a postdoctoral researcher in University of Wales, U.K. From 2001 to 2002, he worked as senior visiting professor in University of Bedfordshire, U.K. Now, he is a full professor in Northeastern University. His current research interests include wireless sensor networks and robot.

Q. C. Han received B.S. degree in Automatic Control from Dezhou University, Dezhou, China, in 2010. He received M.S. degree in Pattern Recognition and Intelligent Systems from Northeastern University, China, in 2012, Shenyang, China. His research interests include robot, embedded system, wireless sensor networks.