ISSN 1330-3651(Print), ISSN 1848-6339 (Online)
UDC/UDK [654.16:004.78]:681.586

MAXIMUM SET COVERS BASED ENERGY CONSERVATION SCHEME IN WIRELESS SENSOR NETWORKS

Lan Yao, Feng Zeng

Original scientific paper

In this paper, an efficient scheme is proposed for energy conservation in wireless sensor networks through solving the maximum set covers problem. First, a distributed mechanism is introduced for the sinks to find at most K paths to each sensor. Then, an algorithm named as MDP-MSC is presented for the maximum set covers problem. Making use of the collected path information in the first step, the proposed algorithm partitions all nodes into possibly maximum disjointed set covers, and the nodes in each set cover have all targets covered, ensuring the network connectivity. When constructing a set cover, the key idea of the proposed algorithm is to select a node joining into the set if it has the minimum distance to the nodes which are already in the set. Simulation is done and compared with Greedy-MSC and HA-MDS, the proposed algorithm has the number of set covers increased by 13 % and 21 %, respectively.

Keywords: *energy conservation, distributed algorithm, maximum set covers, wireless sensor network*

Shema očuvanja energije u bežičnim mrežama osjetila temeljena na maksimalnom broju skupova prikupljenih podataka

Izvorni znanstveni članak

U ovom se radu predlaže učinkovita shema za očuvanje energije u bežičnim mrežama osjetila (senzora) rješavanjem problema maksimalnog broja skupova podataka. Najprije se uvodi distributivni mehanizam za prikupljališta u svrhu pronalaženja najviše K putanja do svakog osjetila. Tada se uvodi algoritam nazvan MDP-MSC za rješavanje problema maksimalnog broja skupova prikupljenih podataka. Koristeći prikupljene podatke o putanji u prvom koraku, predloženi algoritam dijeli sve čvorove u maksimalno mogući broj razdvojenih skupova prikupljenih podataka, a čvorovi u svakom skupu pokrivaju sve ciljeve, osiguravajući povezanost mreže. Kod konstruiranja skupa podataka, ključna ideja predloženog algoritma je izbor čvora pridruženog skupu ako ima minimalnu udaljenost do čvorova koji su već u skupu. Simulacija je obavljena i u usporedbi s Greedy-MSC i HA-MDS predloženim algoritmom broj skupova prikupljenih podataka je porastao za 13 % odnosno 21 %.

Ključne riječi: *bežična mreža osjetila, distribuirani algoritam, maksimalni broj skupova prikupljenih podataka, očuvanje energije*

1 Introduction

A Wirless Sensor Network (WSN) consists of a large number of sensor nodes which have the capabilities of sensing, computing and wireless communicating. Generally, the sensor nodes are randomly deployed in a specific area to sense the environment, gather information about the monitoring targets and transfer the information to the sink nodes hop-by-hop through the wireless communication network constructed by the sensors themselves. Due to the low power, low cost, intelligent and self-organization features, wireless sensor networks have been widely applied in the military, environmental, medical, commercial, disaster prediction and rescue fields.

In a WSN, there are two types of wireless devices i.e. sensor and sink, and the most part of wireless nodes are sensors. Sensors are energy-constrained, used to monitor the targets, and sometimes only for communication purposes. Sink nodes have more energy and the better processing ability than the sensors, and generally act as the gateway or clusternodes to collect the data from the sensor nodes. Under normal circumstances, sinks can communicate directly with each other, but the number of sinks is limited in a WSN.

To achieve the effective collection of the data, WSNs must meet two requirements. One is the full coverage of the targets. The other is the complete connectivity of the network. The full coverage refers to the sensors can monitor all targets, and the complete connectivity means that the data originated from the monitoring targets can reach one of the sinks through the multi-hop wireless transmission. The two requirements are the prerequisites for nodes deployment and organization in WSNs.

For efficiently running of WSN, full coverage and complete connectivity should last as long as possible. However, due to limited energy, sensor nodes would fail to work after a certain runtime, at that time, full coverage and complete connectivity would be broken and network failure would happen. Consequently, much research effort has been done to sensors deployment and organization for the extension of the network lifetime. In this paper, we define the network lifetime as the period of time from the network beginning to network failure.

In a WSN, a large number of sensors are randomly deployed in a specific area, and the sensing area of sensors may overlap in some regions. Hence, there are many redundant sensors in a WSN, a part of the sensors would be enough to ensure network connectivity and full coverage, and other sensors can be switched to the sleep mode for energy saving. When the running sensors suffer the sensing or communication failures for energy consumption, it can arouse the sleeping sensors to take the place of the failure nodes. Consequently, through effective schedule of sensors, the network lifetime can be prolonged.

This paper focuses on energy efficient scheme for lifetime prolonging in WSNs. For the existence of the redundant sensors, activating only the necessary number of sensors at any particular moment can save energy, meanwhile satisfying the full coverage and connectivity. Therefore, one way of energy conservation and lifetime extension is to divide the sensors into disjoint sets, and have the sub-networks constructed by each set meet the two requirements. With the nodes in each set activated successively, the network lifetime will be prolonged to some extent related to the number of the sets. In order to extend the lifetime as long as possible, in this paper, an efficient scheme is presented. First, a distributed mechanism is used to find some paths between a sink and a sensor. Then, based on the selected paths, an algorithm is proposed to find the maximum number of sensor sets. Simulations are done to compare the proposed solution with other alternatives.

The rest of this paper is structured as follows. In Section 2, we describe and analyse the related work. In Section 3, the network model and the research problem are presented. In Section 4, an efficient scheme for energy conservation in WSNs is proposed. Simulation results are presented in Section 5. The last section concludes this paper.

2 Related work

Energy conservation is one of the main goals in the research of WSNs, and has been extensively studied. In the previous work, the node organization and schedule for energy conservation and lifetime extension was deeply analyzed from multiple aspects, such as optimizing node placement to reduce the communication overhead, constructing the virtual backbone based on minimum dominating set to balance the energy consumption on the sensors, scheduling the node status to extend the network lifetime, and power control to avoid the energy holes.

In the research of node placement, the existing work tried to place minimum number of sensors to achieve the full coverage and complete connectivity, while balancing the load on the sensors, thus prolonging the network lifetime. Wang [1] considered the sensing field as an arbitrary-shaped region possibly with obstacles in the sensor placement, proposed the placement algorithm without care of the relationship between the communication range and sensing range, the key idea being to partition the sensing field into smaller sub-regions and to deploy sensors in these sub-regions. LU [2] and other scholars were concerned with uneven power consumption of the sensors nearby the sink, and deduced a relay node density function according to which relay nodes are placed in the sensing field. Experiment showed that the approach based on the density function had higher energy utilization than other alternatives. Misra [3] and other scholars studied constrained versions of the relay node placement problem, where relay nodes can only be placed at a subset of candidate locations. They paid attention to the solution with lower computational complexity. Younis and Akkaya [4] made a survey on the node placement in WSNs, analysed the previous work, and highlighted open problems and research directions.

In WSNs, dominating set based virtual backbone has been proposed as the routing infrastructure to alleviate communication overhead. In the past several years, many algorithms have been proposed to construct the connected dominating set, resulting in the extension of the network lifetime. RAEI [5] studied the minimum connected dominating set problem, proposed a distributed and UDG based algorithm which had outstanding time complexity. ZENG [6] proposed a distributed framework for connectivity and coverage maintenance in WSNs, and the key idea lay in connected dominating set constructing and self-scheduling and controlling of the RF and sensing status. YU [7] introduced the correlation degree between sensor nodes by evaluating the entropy of Gaussian random variables, and proposed a distributed algorithm to construct a connected correlation dominating set by removing redundant sensor nodes.

In addition to node deployment and the virtual backbone, effectively scheduling and organization of the sensors can decrease the total energy consumption. Since redundant nodes exist in the network, it is not necessary for all nodes to be running at the same time. Therefore, all the nodes are divided into several groups, ensuring the full coverage and connectivity for the nodes in each group. Obviously, scheduling the groups successively would prolong the network lifetime, while satisfying the coverage and connectivity requirements. In the research context, many scholars proposed the methods or theory to partition the sensors into maximum number of disjoint groups or sets. Slijepcevic [8] addressed the maximum set covers problem, and designed a heuristic algorithm to partition all sensors into as many as possible disjoint set covers. Cardei [9] proved the maximum set covers problem was NP-hard, proposed two algorithms based on linear programming and greedy searching respectively to solve the problem. LI [10] analysed the problem with graph theory, and proposed a greedy solution. In [11], Zorbas and the co-authors proposed a greedy heuristic algorithm to find the cover sets, and a node selection strategy was presented based on critical control factor, which took much consideration on critical targets and their remaining battery life. Since centralized algorithms are not suitable for large self-organized sensor networks, Pervin [12] and the co-authors proposed a localized algorithm, which used only local information at individual nodes to find a solution. Different from others in the literature, the work from Mohamadi [13] introduced the concept of critical target and sensor, and the proposed algorithm employed learning automata to determine the sensors that should be activated at each stage for monitoring all the targets. Experimental result showed that the existing solutions to maximum set covers problem were most central and based on local information, and the result can be further optimized from the points of the computational complexity and performance.

In this paper, we focus on the maximum set covers problem based on the work in [9, 10], emphasize the design of distributed approach, and take some similar global information into account while partitioning the nodes into the disjoint sets.

3 System description

In this section, we introduce the network model and some definitions, followed by problem description.

3.1 Network model

In WSNs, there are two types of nodes i.e. sensors and sinks. The sensors are energy constrained, acting as the routers for traffic forwarding, and also used to monitor the targets with their sensing unit. As the data aggregation points, the sinks have larger battery power than the sensors. In this paper, it is assumed that the sinks have no limitation of energy, and can communicate directly with each other.

Supposing the set of sinks $W=\{w_1, w_2, \ldots, w_d\}$ and the set of sensors $S=\{s_1, s_2, \ldots, s_n\}$, a WSN can be modeled as an undirected graph $G(V, E)$, where $V = S \cup W$ is the node set composed of sinks and sensors, and E is the set of links. For the simplification of model, but no side effect to the result, it is supposed the edge between one sink and another sink does not exist in E. Each sensor has a transmission range R, and a sensing range r. It is also supposed each sink has the transmission range of R. Given two nodes v and u (one of them should be a sensor), they are connected by an edge $(u, v)\in E$, if and only if they are within transmission range of each other.

In a WSN, the targets should be monitored by the sensors. Supposed $T=\{t_1, t_2, \ldots, t_m\}$ is the set of targets, to a sensor $s_i\in S$, if the target t_j lies within the sensing range of s_i, we denote $t_j\rightarrow s_i$. Then, we define the target subset $T_i=\{t_j \mid t_j\rightarrow s_i, 1 \le j \le m\}$ as the target set of the sensor s_i.

3.2 Definitions

Definition 1: Full coverage

As a subset of S, if S' satisfies $\bigcup_{s_i\in S'} T_i = T$, we call S' a full coverage subset of S. The graph $G'(V', E')$ is the full coverage subgraph of G, where $V' = W \cup S'$ composed of all sinks and the sensors in S'. E' is the subset of E, composed of the links between one sensor and another sensor, and the links between any sink and any sensor.

Definition 2: Sink connected

To each node $v\in S'$ and $T_V \neq \emptyset$, if there exists at least one path between v and one of sinks in subgraph G', we call S' is a sink connected subset of S, and G' is a sink connected subgraph of G.

Definition 3: Set cover and complete partition

It is supposed that $S_1, S_2, \ldots\ldots, S_k$ are k subsets of S, their corresponding subgraphs are $G_1(V_1, E_1)$, $G_2(V_2, E_2)$, …, $G_k(V_k, E_k)$, and satisfying the conditions as follows:

(1) $S_i \cap S_j = \emptyset\ (1 \le i, j \le k, i \neq j)$;

(2) $V_i = S_i \cup W\ (1 \le i \le k)$;

(3) $E_1 \cup E_2 \cup \ldots \cup E_k \subseteq E$;

(4) $V_i \cap V_j = W, E_i \cap E_j = \emptyset\ (1 \le i, j \le k, i \neq j)$;

(5) $G_i\ (1 \le i \le k)$ is the full coverage and sink connected subgraph of G.

Then, we call $S_i (1 \le i \le k)$ is a set cover for all targets in T. If there does not exist a new set cover S_{k+1} satisfying $S_i \cap S_{k+1} = \emptyset\ (1 \le i \le k)$, we call $\{S_1, S_2, \ldots\ldots, S_k\}$ is a complete partition of S.

3.3 Maximum set covers problem

To a complete partition $\{S_1, S_2, \ldots\ldots, S_k\}$, scheduling all nodes in any set cover $S_i (1 \le i \le k)$ to be running status, others to sleeping status, would construct a WSN $G_i(V_i, E_i)$ $(V_i = S_i \cup W)$, which are sink connected and covering all targets. Thus, the WSN G_i can meet the two requirements of data collection. For the battery power of sinks is much larger than sensors, we pay no attention to the energy consumption of sinks in this paper. Supposing that each WSN constructed by $S_i\ (1 \le i \le k)$ has the lifetime of one time unit, to a complete partition with k set covers, the WSN G would have the lifetime of k time units. Consequently, the larger k will lead to the longer of the network lifetime, and reasonably we should maximize k to prolong the lifetime.

In this paper, our purpose is to prolong the lifetime as possible as it can be. From the above analysis, the problem is to find the complete partition of sensors with maximum set covers, called maximum set covers (MSC) problem. The problem can be described formally as follows.

The MSC problem is NP-hard [9]. The existing work has proposed some central and greedy algorithms to solve the problem. Aiming to the decrease of computational complexity and improvement of the result, we propose a distributed mechanism for network information collection and a minimum distance path based algorithm to maximize the number of set covers.

4 An efficient scheme for energy conservation

For energy conservation and lifetime extension, an efficient scheme is proposed. First, a distributed feedback mechanism is designed to find the paths between a sink and a sensor. Through the mechanism, sinks can collect up to K paths to each sensor. Then, based on the collected paths, a greedy algorithm MDP-MSC is presented to solve the MSC problem, aiming to find the maximum set covers, and the key idea is to select the sensors with minimum distance to the constructing set cover to join into the current set cover in each step.

4.1 Distributed mechanism for K paths collection

In a large-scale WSN, there exists a large number of paths between a sink and a sensor, and it is energy-wasted way to find all paths information. According to the Pareto principle i.e. 80 ÷ 20 rule, a small part of paths may be enough to find the optimized solution. For example, there exist many paths between source node and terminal node in Fig. 1, but the K shortest paths are mostly used in data communication, and the parameter K depends on the specific application.

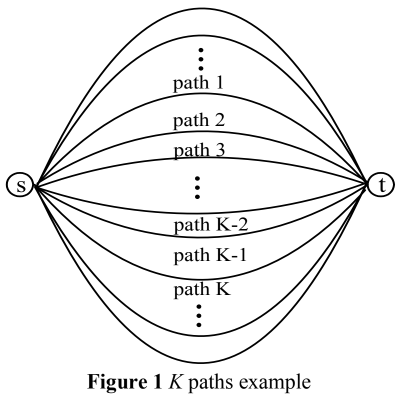

Figure 1 K paths example

In the mechanism, K is given as the upper bound number of the more useful paths between each sink and each sensor, and our goal is to find most K paths for each sensor to a sink. Consequently, K is a parameter to control the communication overhead. The larger K leads to the more network information, but the higher communication overhead. Hence, K should be determined by the reality requirements. The distributed mechanism is described as follows.

max $k = K$

s.t.

$\{S_1, S_2, ..., S_k\}$ is a complete partition of S.

(1) To any sensor $s_i \in S$, if its target set T_i is not empty, it will send message to its neighbors, the message structure is shown in Fig. 2.

sender	hops	Node list			
s_i	n	v_1	v_2	……	v_n

Figure 2 Message structure

(2) To any sensor $s_j \in S$, if it received a message, and the forwarding conditions are satisfied, the message will be modified to add s_j's information and forwarded. The forwarding conditions are as follows. (a) The information of the sensor s_j is not included in the message; (b) The number of the forwarded messages from s_i is not larger than *DMAX*; (c) The hops the message has travelled must be less than *HMAX*. The condition (a) prevents the generation of loop paths. On the one hand, condition (b) will effectively limit the number of messages in the network, thus controlling the communication overhead. On the other hand, condition (b) increases the diversity of the paths. As an addition to (b), condition (c) will decrease the communication overhead too.

(3) To a sink node received a message originated from s_i, it will get a path to s_i from the message, and each sink can save up to K paths for a sender. When it receives a new message different from the previous saved K messages, if the number of hops in the new message is larger than the existing K paths, it will discard the new one. Otherwise, the new one will substitute a path to get more diversity in the paths. To the path P_i, the diversity to the other $K-1$ paths is defined as the Eq. (1), where $| P_i \cap P_j |$ is the number of the sharing sensors in the paths P_i and P_j.

$$Diversity(P_i) = \frac{\sum_{j=1,j\neq i}^{K} \left|P_i \cap P_j\right|}{K-1}. \quad (1)$$

Through the above mechanism, the sinks can obtain up to K paths to each sensor whose target set is not empty. In the mechanism, to each node, *DMAX* is given as the maximum number of forwarded messages from the same sender. Therefore, each sensor should forward n**DMAX* messages at most, and n is the number of sensors in the WSN. Obviously, *DMAX* will be useful for the control of communication overhead. Moreover, a small value of *DMAX* will be helpful to ensure the necessary diversity in the K paths. If *DMAX* is set to 1, the K paths to each sensor will be disjointed with each other.

```
MDP-MSC(G, S, W, T, PATH[])
{
    k=1; Sk=W; //Initialization
    //constructing set covers
    //until it doesn't exist
    while(BuildSetCover (S k, G, S, W, T,
PATH[])){
        k = k + 1; Sk=W;
    }
    //the last set is not a full coverage
    k = k - 1;
    //S1 , S2 , ……, Sk are the set covers
    return { S1 , S2 , ……, Sk }
}
```

```
bool BuildSetCover (Sk, G, S, W, T, PATH[])
{
    //Initialization, CTk is target set of Sk
    Sk = W; CTk={};
    //select sensors and join into Sk
    while(SelectSensor(Sk, CTk, G, S, T,
PATH[]){

    }
    //If Sk is full coverage, return success
    if(CTk==T)
    return true;
    else
    return false;
```

```
bool SelectSensor(Sk, CTk, G, S, T, PATH[])
{
    //find min distancepath/subpath to Sk
    //SUBPATH is the node set of the
path/subpath
    if(MinDistPah(SUBPATH, Sk, S, PATH[])){
        //update the node set and target set
        Sk = Sk + SUBPATH;
        S = S - SUBPATH;
        CTk= CTk + Tsubpath;
        // Tsubpath is target set of subpath
        return true;
    }
    else
        return false;
}
```

```
bool MinDistPath(SUBPATH, Sk, S, PATH[])
{
    //compute the distance between each
    related //sensor andSk, and find the
    minimum one
    minDist = ∞;
    for each valid path in PATH[]
        for the nearest sensor v in the path
to Sk
    If (dist = Dist(v, Sk)< minDist)
              minDist = dist;
    if (minDist <∞){
        SUBPATH ← the selected path/subpath
        return true;
    }
    else
```

Figure 3 MDP-MSC pseudo code

4.2 Minimum distance path based algorithm for MSC problem

Based on the collected paths, a minimum distance path based algorithm MDP-MSC is presented to find the maximum set covers. When constructing a set cover, in each step, extracting information from all collected paths, MDP-MSC selects a path or subpath with minimum distance to the nodes in the current set cover, and the nodes in the path or subpath will all join into the set cover. From the point of sensors, the subpath of the collected path is also a path to the destination. Thus, in the following description, a path may be a subpath of the collected paths. Given a WSN G, the set of sensors S, the set of sinks W, the set of targets T, and the set of paths *PATH[]*, MDP-MSC can be described in pseudo code as figure 3.

$$Dist(v_1, S_i) = \frac{x-1}{\sum_{j=1}^{x-1} w_j}. \quad (2)$$

In each step, MDP-MSC calls the function *BuildSetCover* to find a set cover, and returns the maximum number of set covers. When a set cover is being constructed, *BuildSetCover* selects some sensors to join into the constructing set cover through the function of *SelectSensor*. In sensor selecting, the function *MinDistPah* tries to find one of the sensors which are monitoring one uncovered target at lease. If a sensor has the minimum distance path to the constructing set cover, the nodes in the path will join into the set cover. To a sensor v_1, if there exists a x-hop path $v_1 v_2 \dots v_x$ to the node set S_i, and in these sensors, only v_x is in the set of S_i, the distance from v_1 to S_i is denoted as Eq. (2). In the equation, w_j is the number of targets monitored by v_j, and $x-1$ means the hop count between v_1 and v_x.

For example, it is supposed in Fig. 4 that v_1, v_2, and v_3 are the sensors which are not in the node set S_i, and the sensor v_4 is in S_i, v_1 is monitoring a target uncovered by S_i, and v_3 is monitoring two other targets uncovered by S_i. Then, the distance $Dist(v_1, S_i)$=3/3=1, and $Dist(v_3, S_i)$=1/2.

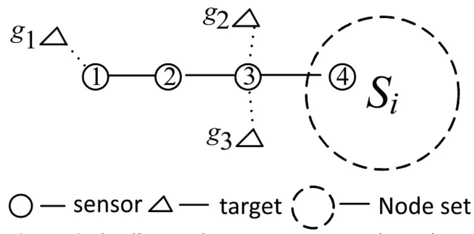

Figure 4 The distance between a sensor and a node set

Furthermore, if more than one sensor has the equal minimum distance path to the constructing set cover, *MDP-MSC* will select one of them considering the degree of sensors and the number of the affected downstream sensors.

4.3 Theoretical analysis

Given a WSN with n sensors and m targets, for some sensors are the relay nodes, we can assume at most cm sensors monitoring the targets, and c is a constant. Through the distributed mechanism, the sinks can collect at most Kcm paths from the sinks to the sensors monitoring a target at least. In the function *MinDistPath*, the maximum size of *PATH[]* is Kcm, the length of each path is at most n, thus the *MinDistPath* has the time complexity of $O(Kcmn)$. The function *SelectSensor* calls the *MinDistPath*, and has some sensor set operations, its time complexity is determined mainly by *MinDistPath*. Therefore, *SelectSensor* has the time complexity of $O(Kcmn)$. When set cover constructing, in each round, *SelectSensor* would let at least one target join the temporary target set CT_k. Therefore, if a set cover exists, with at most m rounds of *SelectSensor*, a set cover can be built up. Thus, *BuildSetCover* has the complexity of $O(Kcm^2n)$. Supposed at most k set covers in a WSN, we can draw the conclusion that the *MDP-MSC* algorithm has the complexity of $O(Kkcm^2n)$, where K and c are the constants, and k is much less than n. Consequently, the time complexity of *MDP-MSC* is $O(m^2n)$.

Compared with LP-MSC, Greedy-MSC, and HA-MDS with the time complexity $O(p^3 n^3)$, $O(dm^2n)$, $O(n^3 \lg n)$ respectively, MDP-MSC has the lowest complexity. In the next section, the simulations have been done to evaluate the performance of MDP-MSC.

5 Simulation

We have performed a simulation-based analysis of the proposed algorithm, and validated its correction and effectiveness. In the simulation, we have realized the related algorithms in Microsoft Visual C++ 6.0 running on the PCs with CPU Pentium4—3,06 GHz, RAM 2 GB, and Windows XP installed.

In the simulation, the given number of nodes is randomly placed in the 500×500 m square, and some sinks and targets are randomly generated. The communication distance for each sensor is set at 100 m, and the sensing range varies from 20 m to 100 m. Then, 100 random WSN topologies are generated. To each topology, MDP-MSC, Greedy-MSC [9] and HA-MDS [10] are used to find the maximum set covers respectively, and the results are shown in the following figures and tables.

5.1 Performance comparison

To the various number of sensors, the maximum number of sensors monitoring the same target is fixed, and the number is the upper bound of set covers. In the simulation, the sensing range is set to 50 m, and MDP-MSC has the parameters as K=20, $DMAX$=5, $HMAX$=15. The result is shown in Fig. 5, compared with Greedy-MSC and HA-MDS, MDP-MSC has the number of set covers increased by 13 % and 21 % respectively.

In the simulation, each algorithm runs on the 100 random topologies for the maximum set covers. To the same node number and condition, it is supposed that L_a denotes the number of topologies on which MDP-MSC has better result than Greedy-MSC, L_e denotes the number of topologies on which MDP-MSC and Greedy-MSC have the same result, and L_s denotes the number of topologies on which Greedy-MSC outperforms MDP-MSC. For the various numbers of nodes, the values of L_a, L_e, L_s are listed in Tab. 1. On average, MDP-MSC outperforms Greedy-MSC with the ratio 76,8 %.

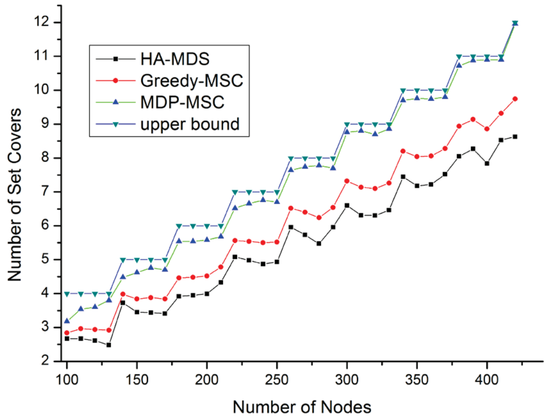

Figure 5 Performance vs. node number

In the simulation, each algorithm runs on the 100 random topologies for the maximum set covers. To the same node number and condition, it is supposed that L_a denotes the number of topologies on which MDP-MSC has better result than Greedy-MSC, L_e denotes the number of topologies on which MDP-MSC and Greedy-MSC have the same result, and L_s denotes the number of topologies on which Greedy-MSC outperforms MDP-MSC. For the various numbers of nodes, the values of L_a, L_e, L_s are listed in Tab. 1. On average, MDP-MSC outperforms Greedy-MSC with the ratio 76,8 %.

In the above simulation, the CPU execution average time of each algorithm is shown in Tab. 2. Consistent with the theoretical analysis, MDP-MSC has the lower time complexity than the other two algorithms.

Table 1 Performance comparison

Node numbers	100	140	180	220	300	340	380	420
L_a	22	64	68	84	88	90	98	100
L_e	64	32	30	14	11	10	2	0
L_s	14	4	2	2	1	0	0	0

Table 2 Executive CPU Time (millisecond)

Node numbers	100	140	180	220	300	340	380	420
HA-MDS	8	15	7	68	157	249	318	407
Greedy-MSC	6	13	5	54	15	73	253	342
MDP-MSC	6	11	9	41	83	144	195	261

Given the number of sensors is 300, and the sensing range varies from 20 m to 100 m, the result is shown in Fig. 6. With the sensing range increasing, the algorithms can find more set covers, and MDP-MSC outperforms the other two algorithms.

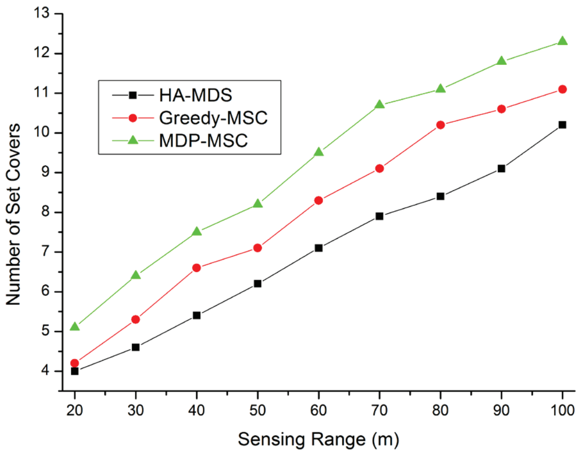

Figure 6 Performance vs. sensing range

5.2 The impact of the parameters on MDP-MSC

In MDP-MSC, the selection of sensors joining into the set cover is based on the collected paths, and the number of paths is determined by the parameter *K*. Therefore, it is obvious that *K* has the impact on the performance of MDP-MSC. At the scene of 300 and 400 sensors, given that the maximum number of set covers is 8, the impact of various *K* on MDP-MSC is shown in Fig. 7. In the figure, when *K* is less than 15, the increase of *K* has more impact on the result. When *K* is larger than 15, the increase of *K* does little to the result. Hence, if the sensors are densely placed, setting *K* as 15 is reasonable, ensuring better performance and effective control of communication overhead.

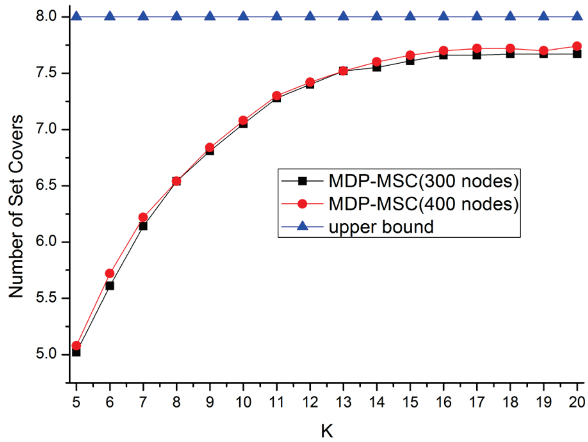

Figure 7 Impact of *K* on MDP-MSC

Given *K*=15 and the sensor number is 300, Fig. 8 shows the performance of MDP-MSC with various *HMAX* and *DMAX*. As is shown in the figure, the small value of *HMAX* and *DMAX* can meet the performance requirement. From the viewpoint of energy consumption, the smaller *HMAX* and *DMAX* would lead to the lower communication overhead.

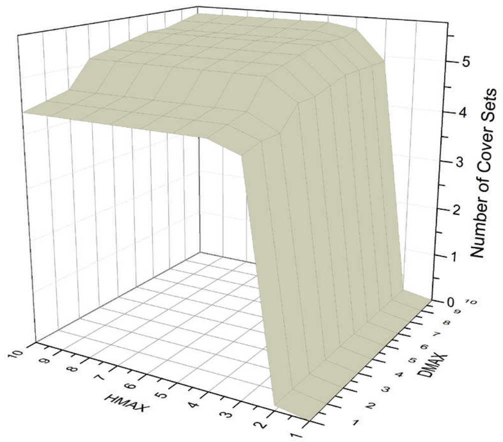

Figure 8 Impact of *HMAX* and *DMAX* on MDP-MSC

6 Conclusion

Lifetime extension and energy conservation in WSNs have been studied extensively. This paper studies the schedule and organization of sensors, and proposes an efficient scheme for lifetime extension. Simulation results show that the proposed algorithm MDP-MSC outperforms the other two algorithms, and the algorithm parameters have the impact on the result. Based on experimental analysis, the suggestion for parameters setting is given. This work would be valuable to the application in the WSNs. In the next research, we would pay more attention to the balance of energy consumption in the sensor nodes.

Acknowledgements

This research was financially supported by the National Natural Science Foundation of China under Grant No. 61103202 and the Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20110162120046.

7 References

[1] You-Chiun, Wang; Chun-Chi, Hu; Yu-Chee, Tseng. Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks. // First International Conference on Wireless Internet, 2005, pp. 114-121.
[2] Lu, K.; Liu, G.; Mao, R. et al. Relay node placement based on balancing power consumption in wireless sensor networks. // IET Wireless Sensor Systems. 1, 1(2011), pp. 1-6.
[3] Misra, S.; Hong Seung, Don; Xue, Guoliang et al. Constrained Relay Node Placement in Wireless Sensor Networks: Formulation and Approximations. // IEEE/ACM Transactions on Networking. 18, 2(2010), pp. 434-447.
[4] Younis, M.; Akkaya, K. Strategies and techniques for node placement in wireless sensor networks: a survey. // Ad Hoc Networks. 6, 4(2008), pp. 621-655.
[5] Raei, H.; Sarram, M.; Adibniya, F. et al. Optimal distributed algorithm for minimum connected dominating sets in wireless sensor networks. // 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 29 Sept-2 Oct 2008, Piscataway, NJ, USA: IEEE, 2008, pp. 695-700.
[6] Yuanyuan, Zeng; Cormac J. Sreenan; Naixue, Xiong; et al. Connectivity and coverage maintenance in wireless sensor networks. // Journal of Supercomputing. 52, 1(2010), pp. 23-46.
[7] Rui-Yun, Yu; Xing-Wei, Wang; Yong-He, Liu. Correlation dominating set construction based upon entropy evaluation in wireless sensor networks. // Jisuanji Xuebao/Chinese Journal of Computers. 34, 1(2011), pp. 87-95.
[8] Slijepcevic, S.; Potkonjak, M. Power efficient organization of wireless sensor networks. // Proceedings of International Conference on Communications, 11-14 June 2001, Piscataway, NJ, USA: IEEE, 2001, pp. 472-476.
[9] Cardei, M.; Thai, M. T.; Yingshu, Li et al. Energy-efficient target coverage in wireless sensor networks. // Proceedings IEEE Infocom 2005, 13-17 March 2005, Piscataway, NJ, USA: IEEE, 2005, pp. 1976-1984.
[10] Li, Liu; Bin, Hu; Huifang, Miao; et al. Achieving Energy Conservation, Coverage and Connectivity Requirements in Wireless Sensor Networks. // 29th IEEE International Conference on Distributed Computing Systems Workshops, 2009, pp. 227-232.
[11] Zorbas, D.; Glynos, D.; Kotzanikolaou, P. et al. Solving coverage problems in wireless sensor networks using cover sets. // Ad Hoc Networks. 8, 4(2010), pp. 400-415.
[12] Nargis, Pervin; Dipankar, Layek; Nabanita, Das. Localized Algorithm for Connected Set Cover Partitioning in Wireless Sensor Networks. // International Conference on Parallel, Distributed and Grid Computing, 2010, pp. 229-234.
[13] Hosein, Mohamadi; Abdul Samad, Ismail; Shaharuddin, Salleh et al. Learning automata-based algorithms for finding cover sets in wireless sensor networks. // The Journal of Supercomputing, April 2013, DOI: 10.1007/s11227-013-0941-8.

Authors' addresses

Lan Yao, assoc. prof., dr.
College of Mathematics and Econometric,
Hunan University
410082 Changsha, Hunan, China
E-mail: yao@hnu.edu.cn

***Feng Zeng, assoc. prof., dr.* (*corresponding author*)**
School of Software, Central South University
410083 Changsha, Hunan, China
E-mail: fengzeng@csu.edu.cn