Connectivity-Aware Network Maintenance via Relays Deployment

Ahmed S. Ibrahim, Karim G. Seddik, and K. J. Ray Liu

Department of Electrical and Computer Engineering, and Institute for Systems Research
University of Maryland, College Park, MD 20742, USA.
{asalah, kseddik, kjrliu}@umd.edu

***Abstract*— In this paper, we address the network maintenance problem, in which we aim to maximize the lifetime of a sensor network by adding a set of relays to it. The network lifetime is defined as the time until the network becomes disconnected. The Fiedler value, which is the algebraic connectivity of a graph, is used as an indicator of the network health. The network maintenance problem is formulated as a standard semi-definite programming (SDP) optimization problem that can be solved efficiently in polynomial time. First, we present a network maintenance algorithm that obtains the near-optimum locations for a given set of relays. Second we propose a routing algorithm, namely, Weighted Minimum Power Routing (WMPR) algorithm, that significantly increases the network lifetime due to the efficient utilization of the deployed relays. Third, we propose an adaptive network maintenance algorithm that relocates the deployed relays based on the network health indicator. Finally, we consider the network repair problem, in which we find the minimum number of relays along with their near-optimum locations to reconnect a disconnected network. We propose an iterative network repair algorithm that utilizes the network maintenance algorithm.**

I. Introduction

Recently, there have been much interest in wireless sensor networks due to its various application areas such as battlefield surveillance systems and industry monitoring systems [1]. A sensor network consists of a large number of sensor nodes, which are deployed in a particular area to measure certain phenomenon such as temperature and pressure. These sensors send their measured data to a central processing unit (information sink), which collects the data and develops a decision accordingly. Often sensors have limited energy supply. Hence efficient utilization of the sensors' limited energy, and consequently extending the network lifetime, is one of the design challenges in wireless sensor networks.

The network lifetime is defined as the time until the network becomes disconnected [2]. The network is considered connected if there is a path, possibly a multi-hop one, from each sensor to the central processing unit. Deploying a set of relays in a wireless sensor network is one of the main approaches to extend the network lifetime. More precisely, relays can forward the sensors' data and hence they contribute to reducing the transmission power required by many sensors per transmission, which can extend the lifetime of these sensors. However, the problem of finding the optimum locations of these relays is shown to be NP-hard [3]. Therefore, there is a need to find a heuristic algorithm that can find near-optimum locations for the available set of relays in polynomial time. This problem is known in the literature as *network maintenance* problem. Recently, there have been numerous network maintenance algorithms [3]-[6]. Himsoon *et al.* proposed a relay deployment algorithm that maximizes the minimum sensor lifetime by exploiting the cooperative diversity in [5]. A mathematical approach to positioning and flying an unmanned air vehicle (UAV) over a wireless ad hoc network was proposed in [6].

In wireless sensor networks and after deploying the sensors for a while, some sensors may lose their available energy, which affects each sensor's ability to send its own data as well as forward the other sensors' data. This affects the network connectivity and may result in the network being disconnected. In this case, there is a need to determine the minimum number of relays along with their optimum locations that are needed to reconnect this network. Similar to the network maintenance problem, this problem is NP-complete [7] and there is a need for a heuristic algorithm to solve this problem in polynomial time. This problem is known as *network repair* problem. Several works have considered the network repair problem. For instance, the connectivity improvement using Delaunay Triangulation (CIDT) [7] constructs a Delaunay Triangulation in the disconnected network and deploy nodes in certain triangles according to several criteria.

In this paper, first we present an efficient network maintenance algorithm that finds the near-optimum locations for an available set of relays to maximize algebraic connectivity of a graph, which was proposed in [4]. It is based on the semi-definite programming (SDP) formulation of the problem, which can be solved in polynomial time. In this paper, we show that this algorithm can be utilized to maximize the network lifetime as well. Second, we propose a routing algorithm, namely, Weighted Minimum Power Routing (WMPR) algorithm, that can extend the network lifetime by assigning weights to the sensors that are different from that of the relays. Third, we propose an adaptive network maintenance algorithm that increases the network lifetime by relocating the relays depending on the network health indicator, which is the Fiedler value of the remaining network. Finally, we propose an iterative network repair algorithm, which finds the minimum number of relays along with their near-optimum locations needed to reconnect a disconnected network.

The rest of the paper is organized as follows. In the next section, we describe the network model. We formulate the network maintenance problem and describe the proposed solution in Section III. We build upon that algorithm and propose different lifetime-maximization strategies in Section IV. In Section V, we address the network repair problem and describe our proposed solution. In Section VI, we present some simulation results that show the significance of our proposed algorithms. Finally, Section VII concludes the paper.

1525-3511/08/$25.00

II. Network Model

In this section, we describe the wireless sensor network model. In addition, we review some concepts related to the algebraic connectivity of a graph. A wireless sensor network can be modeled as an undirected weighted graph $G(V, E)$, where $V = \{v_1, v_2, \cdots, v_n\}$ is the set of all nodes (sensors) and E is the set of all edges (links). Let n and m denote the number of nodes and edges in the graph, respectively, i.e., $|V| = n$ and $|E| = m$, where $|.|$ denotes the cardinality of the given set.

Let $d_{i,j}$ denote the distance between two nodes $\{v_i, v_j\} \in V$ and let α denote the path loss exponent. The channel between each two nodes $\{v_i, v_j\} \in V$, denoted by $h_{i,j}$, is modeled as a complex Gaussian random variable with zero-mean and variance equal to $d_{i,j}^{-\alpha}$. Thus, the channel gain $|h_{i,j}|$ follows a Rayleigh fading model [8]. Furthermore, the channel gain squared $|h_{i,j}|^2$ is an exponential random variable with mean $d_{i,j}^{-\alpha}$. The noise is modeled as a Gaussian random variable with zero-mean and variance N_0. We assume that binary phase shift keying (BPSK) modulation scheme is considered for the transmission between each two nodes. Thus, the probability of bit error, or bit error rate (BER), can be written as [8]

$$p = \frac{1}{2}\Big(1 - \sqrt{\frac{\gamma_{i,j}}{1+\gamma_{i,j}}}\Big) , \tag{1}$$

where $\gamma_{i,j} = \frac{P_{i,j}\, d_{i,j}^{-\alpha}}{N_0}$ denotes the signal-to-noise ratio (SNR) and $P_{i,j}$ is the transmission power of node v_i to transmit its data to node v_j. From (1) the transmission power of node v_i, required to achieve a desired average BER of p^o over link (v_i, v_j), is given by

$$P_{i,j}^o = d_{i,j}^{\alpha}\, N_0\, \frac{(1-2\,p^o)^2}{1-(1-2\,p^o)^2} . \tag{2}$$

We assume that each node $v_i \in V$ can transmit with power $0 \le P_{i,j} \le P_{max}$, where P_{max} denotes the maximum transmission power of each node. Also, we assume that the noise variance N_0 and the desired BER p^o are constant for all the transmissions in the network. Therefore, an undirected weighted edge (v_i, v_j) exists if $P_{i,j}^o \le P_{max}$, where $P_{i,j}^o$ is calculated as in (2). Furthermore, the weight of an edge l connecting v_i and v_j, denoted by $w_{i,j}$ or w_l, is a function of the transmitted power $P_{i,j}^o$ that depends on the considered routing scheme, as will be described in Section IV-A.

For an edge l, $1 \le l \le m$, connecting nodes $\{v_i, v_j\} \in V$, define the edge vector $\mathbf{a_l} \in \mathbf{R^n}$, where the i-th and j-th elements are given by $a_{l,i} = 1$ and $a_{l,j} = -1$, respectively, and the rest is zero. The *incidence* matrix $\mathbf{A} \in \mathbf{R^{n\times m}}$ of the graph G is the matrix with l-th column given by $\mathbf{a_l}$. The weight vector $\mathbf{w} \in \mathbf{R^m}$ is defined as $\mathbf{w} = [w_1, w_2, ..., w_m]^T$, where T denotes transpose.

The *Laplacian* matrix $\mathbf{L} \in \mathbf{R}^{n\times n}$ is defined as

$$\mathbf{L} = \mathbf{A}\,\mathrm{diag}(\mathbf{w})\,\mathbf{A}^T = \sum_{l=1}^{m} w_l\,\mathbf{a_l}\,\mathbf{a_l}^T , \tag{3}$$

where $\mathrm{diag}(\mathbf{w}) \in \mathbf{R}^{m\times m}$ is the diagonal matrix formed from $\mathbf{w}$. The diagonal entry $L_{i,i} = \sum_{j\in N(i)} w_{i,j}$, where $N(i)$ is the set of neighboring nodes of node v_i that have a direct edge with node v_i. $L_{i,j} = -w_{i,j}$ if $(v_i, v_j) \in E$, otherwise $L_{i,j} = 0$. Since all the weights are nonnegative, the Laplacian matrix is positive semi-definite, which is expressed as $\mathbf{L} \succeq 0$. In addition, the smallest eigenvalue is zero, i.e., $\lambda_1(\mathbf{L}) = 0$. The second smallest eigenvalue of $\mathbf{L}$, $\lambda_2(\mathbf{L})$, is the algebraic connectivity of the graph G [9], [10], [11], [12]. It is called *Fiedler value* and it measures how connected the graph is because of following main reasons. First, $\lambda_2(\mathbf{L}) > 0$ if and only if G is connected and the multiplicity of the zero-eigenvalue is equal to the number of the connected sub-graphs. Second, $\lambda_2(\mathbf{L})$ is monotone increasing in the edge set.

III. Network Maintenance

In this section, we briefly formulate the network maintenance problem. Network lifetime is defined as the time until the network becomes disconnected, which happens when there is no path from any existing sensor to the central unit. Consequently, the network dies (becomes disconnected) if there is no path between any two living sensors. Hence, there is a direct relation between keeping the network connected as long as possible and maximizing the network lifetime. As discussed in Section II, the Fiedler value defines the algebraic connectivity of the graph and it is a good measure of how connected the graph is. Based on that, we consider the Fiedler value as a measure of the network lifetime as well.

The network maintenance problem can be stated as follows. Given a base network deployed in a $g \times g$ square area and represented by the graph $G_b = (V_b, E_b)$, as well as a set of K relays, what are the optimum locations for these relays in order to maximize the Fiedler value of the resulting network? Intuitively, adding a relay to the base network may result in connecting two sensors or more, which were not connected together. Because this relay can be within the transmission range of these two sensors, hence it can forward data from one sensor to the other. Therefore, adding a relay may result in adding an edge or more to the original graph.

Let $E_c(K)$ denote the set of edges resulting from adding a candidate set of K relays. Thus, the network maintenance problem can be formulated as

$$\max_{E_c(K)} \;\; \lambda_2\Big(\mathbf{L}\big(E_b \cup E_c(K)\big)\Big) . \tag{4}$$

The main algorithm to solve the network maintenance problem (4) can be described as follows [4]. First, we divide the $g \times g$ network area into n_c equal square regions, each with width h. Thus, $n_c = (\frac{g}{h})^2$. We represent each region by a relay deployed in its center. The optimization problem (4) can be formulated as

$$\max \;\; \lambda_2\big(\mathbf{L}(\mathbf{x})\big) \;\; \text{s.t.} \;\; \mathbf{1}^T\,\mathbf{x} = K,\; \mathbf{x} \in \{0,1\}^{n_c} , \tag{5}$$

where

$$\mathbf{L}(\mathbf{x}) = \mathbf{L}_b + \sum_{l=1}^{n_c} x_l\,\mathbf{A}_l\,\mathrm{diag}(\mathbf{w}_l)\,\mathbf{A}_l^T , \tag{6}$$

Step 1 The first level: Divide the network area into n_c equal square regions. Each region is represented by a relay at its center.
Step 2 Solve the optimization problem in (7) and obtain the best $K < n_c$ relays among the n_c relays defined in *Step 1*.
Step 3 Start a new level: For each solution x_k, $k = 1, 2, \cdots, K$, divide the k-th region into n_c equal square regions and obtain the best area for this relay. This can be solved using (7) by setting $K = 1$.
Step 4 Repeat *Step 3* until there is no improvement in the resulting Fiedler value.

TABLE I

Proposed network maintenance algorithm.

and $\mathbf{1} \in \mathbf{R}^{n_c}$ is the all-ones vector. In (6), $\mathbf{A}_l$ and $\mathbf{w}_l$ are the incidence matrix and weight vector resulting from adding relay l to the original graph. We note that the optimization vector in (6) is the vector $\mathbf{x} \in \mathbf{R^{n_c}}$. Each element in $\mathbf{x}$ is either 1 or 0, which represents whether this relay should be chosen or not, respectively.

In [4] we have shown that by relaxing the Boolean constraint $\mathbf{x} \in \{0, 1\}^{n_c}$ to be a linear constraint $\mathbf{x} \in [0, 1]^{n_c}$, this problem is equivalent to the following SDP optimization problem [10], [12]

$$\begin{aligned} &\max \ s \\ &\text{s. t.} \ \ s(\mathbf{I} - \frac{1}{n}\mathbf{1}\,\mathbf{1}^T) \preceq \mathbf{L}(\mathbf{x}),\ \mathbf{1}^T\,\mathbf{x} = K,\ 0 \leq \mathbf{x} \leq 1\ , \end{aligned} \tag{7}$$

where $\mathbf{I} \in \mathbf{R}^{n \times n}$ is the identity matrix and $\mathbf{B} \preceq \mathbf{A}$ denotes that $\mathbf{A} - \mathbf{B}$ is a positive semi-definite matrix.

The optimization problem in (7) can be solved efficiently using any SDP standard solver such as the SDPA-M software package [13]. Then, we use a heuristic to obtain a Boolean vector from the SDP optimal solution, which is the solution for the original problem in (5). In this paper, we consider a simple heuristic, which is to set the largest K x_l to 1 and the rest to 0. We call this stage of the algorithm by *level*.

In order to improve the current solution, we repeat the same procedure by dividing each k-th region into n_c smaller areas and representing each one by a relay at its center. Then, we find the best location in these n_c regions to have the relay deployed there. This problem is the same as the one in (5) by setting $K = 1$. We do the same step for each region k, $1 \leq k \leq K$, obtained in the first step. This algorithm is repeated for a finite number of levels. In Table I, we summarize the implementation of our proposed network-maintenance algorithm.

IV. Lifetime-Maximization Strategies

In this section, we build upon the network maintenance algorithm described in Table I and propose two strategies that can maximize the network lifetime. First, we propose the WMPR algorithm, which efficiently utilizes the deployed relays in a wireless network. Second, we propose an adaptive network maintenance algorithm, which relocates the relays based on the network status.

A. Weighted Minimum Power Routing (WMPR) Algorithm

We begin by explaining the conventional Minimum Power Routing (MPR) algorithm. The MPR algorithm constructs the minimum-power route from each sensor to the central unit, by utilizing the conventional Dijkstra's shortest-path algorithm [14]. The cost (weight) of a link (v_i, v_j) is given by

$$w_{i,j}|_{MPR} = P_i^o + P_r\ , \tag{8}$$

where P_i^o is the transmission power given in (2) and P_r denotes the receiver processing power, which is assumed to be fixed for all the nodes.

In (8), it is obvious that the MPR algorithm does not differentiate between the original sensors and the deployed relays while constructing the minimum-power route. In most of the applications, it is very possible that the few deployed relays have higher initial energy than that of the many existing sensors. Intuitively to make the network live longer, the relays should be utilized more often than the sensors. Consequently, the loads of the sensors and relays will be proportional to their energies, which results in more balanced network. The WMPR algorithm achieves this balance by assigning weights to the sensors and the relays, and the cost of each link depends on these weights. Therefore, we propose to have the weight of the link (v_i, v_j) given by

$$w_{i,j}|_{WMPR} = e_i\,P_i^o + e_j\,P_r\ , \tag{9}$$

where e_i denotes the weight of node v_i. By assigning the relays smaller weight than that of the sensors, the network becomes more balanced and the network lifetime is increased. In conclusion, the WMPR utilizes the Dijkstra's shortest-path algorithm to compute the route from each sensor to the central unit using (9) as the link cost. More importantly, weights of the relays should be smaller than that of the sensors.

B. Adaptive Network Maintenance Algorithm

In the fixed network maintenance strategy, described in Table I, each relay will be deployed in a particular place and will be there until the network dies. Intuitively, the network lifetime can be increased by adaptively relocating the relays depending on the status of the network. Such a scheme can be implemented via low-altitude Unmanned Air Vehicles (UAVs) or movable robots depending on the network environment. For instance, we can utilize one UAV or more, which can fly along the obtained relays' locations to improve the connectivity of the ground network. In each location, UAV acts exactly as a fixed relay connecting a set of sensors through multi-hop relaying.

The proposed adaptive network-maintenance algorithm is implemented as follows. First, the initial locations of the deployed relays are determined using the network-maintenance algorithm described in Table I. Whenever a node dies, the Fiedler value of the remaining network is calculated. If it is greater than certain threshold, then the network is likely to be disconnected soon. Therefore, the deployment algorithm is calculated again and the new near-optimum relays' locations are obtained. Finally each relay is relocated to the new location, if it is different from its current one. The algorithm is repeated until the network is disconnected. The adaptive network maintenance algorithm is summarized in Table II.

Step 1 Compute the near-optimum locations for the available K relays using the network maintenance algorithm in Table I.
Step 2 If a node dies, compute the Fiedler value of the remaining graph. If the Fiedler value is lower than certain threshold, repeat *Step 1*. The relay is relocated if the new Fiedler value is higher than the current one.
Step 3 Repeat *Step 2*.

TABLE II

Proposed adaptive network maintenance algorithm.

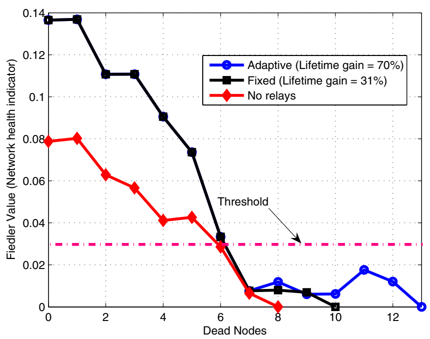

Fig. 1. Fiedler value (Network health indicator) versus the number of dead nodes, for $n = 20$ sensors deployed randomly in 6×6 square field, is plotted. Effects of adaptive and fixed network maintenance algorithms are illustrated.

In the sequel, we present an example to illustrate how effective the adaptive network maintenance algorithm can be. Consider a wireless network of $n = 20$ nodes deployed randomly in a 6×6 square area. We assume that only $K = 1$ relay is available. Data generated for each sensor follows a Poisson distribution with rate 10 packets per unit time. When a node sends a packet, the remaining energy is decreased by the amount of the transmission energy and it dies when it has no remaining energy. In addition, the Fiedler value threshold is chosen to be 0.03.

Fig. 1 depicts the Fiedler value of the network as a function of the number of dead nodes utilizing the MPR algorithm. The original network is disconnected after the death of 8 nodes. By adding a fixed relay, the network lifetime increases, resulting in a network lifetime gain of 31%. The network lifetime gain due to adding K relays is defined as $G(K) = \frac{T(K)-T(0)}{T(0)}$, where $T(K)$ is the network lifetime after deploying K relays. By considering $K = 1$ relay, the adaptive network-maintenance algorithm achieves lifetime gain of 70%. This example shows that the proposed adaptive network maintenance algorithm can significantly increase the network lifetime. We clarify that these lifetime gains are specific to that particular example and do not represent the average results. The average results of the various proposed network maintenance strategies are provided in Section VI.

It is worth to note that Fig. 1 shows that the Fiedler value of the living network can be thought of as a *health indicator* of the network. If the network health is below certain threshold, then the network is in danger of being disconnected. Thus, a network maintenance strategy, either fixed or adaptive, should be implemented. However, if the network becomes disconnected then intuitively we can consider reconnecting the network again via deploying the minimum number of relays.

Step 1 Initially, let $K = 1$ candidate relay.
Step 2 Implement the network maintenance algorithm in Table I utilizing K candidate relays.
Step 3 If the Fiedler value of the resulting graph is strictly greater than 0, stop. Otherwise, increment the number of relays by one and repeat *Step 2*.

TABLE III

Proposed network repair algorithm.

This is the network repair problem and it is discussed in the following section.

V. Network Repair

In this section, we consider the network repair problem. In particular, the network is initially disconnected and we need to find the minimum number of relays along with their optimum locations in order to reconnect the network. Let a disconnected base network deployed in a $g \times g$ square area be represented by the graph $G_b = (V_b, E_b)$. Hence, $\lambda_2\Big(\mathbf{L}\big(E_b\big)\Big) = 0$. The network repair problem can be formulated as

$$\min \quad K \quad \text{s.t.} \quad \lambda_2\Big(\mathbf{L}\big(E_b \cup E_c(K)\big)\Big) > 0 \,, \qquad (10)$$

where $E_c(K)$ denotes the set of edges resulting from adding a candidate set of K relays.

In [7], it was shown that the network repair problem is NP-complete and hence we propose a heuristic algorithm to solve it. We utilize our proposed solution for the network maintenance problem in solving the network repair problem. More precisely, we propose an iterative network repair algorithm, which is implemented as follows. First, we assume that $K = 1$ relay is enough to reconnect the network. Second, we solve the network maintenance problem in (5) to find the near-optimum location for that relay. If the Fiedler value of the resulting network is strictly greater than zero then the network is reconnected and the algorithm stops. Otherwise, the number of candidate relays is incremented by one and the algorithm is repeated. Table III summarizes the network repair algorithm.

VI. Simulation Results

In this section, we present some simulation results to show the performance of our proposed algorithms. In the simulations, we have used the SDPA-M software package [13] to solve the SDP problem in (7). We consider $n = 20$ nodes deployed randomly in 6×6 square area. Data generated at the sensors follow Poisson process with rate 10 arrival packets per unit time. The desired BER for the transmissions over any link is $p^o = 10^{-4}$, the noise variance $N_0 = -20$dBm, the maximum power $P_{max} = 0.15$ units, the receiver processing power is $P_r = 10^{-4}$ units, and the initial energy of every sensor is 0.1 unit. The shown results are averaged over 1000 different network realizations.

In [4], we have shown that 3 levels of the SDP-based network maintenance algorithm described in Table I gives accurate results. So, we use 3 levels in our simulations in this section. The number of candidate relay locations used in the network maintenance algorithm is $n_c = 25$ location. In Fig. 2, we show the effect of increasing the number of added relays

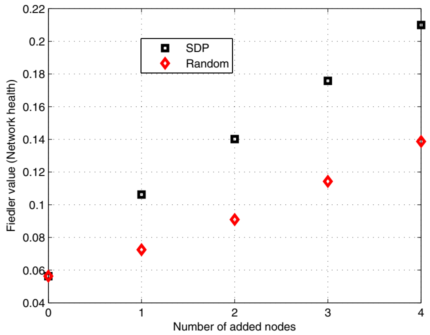

Fig. 2. The average Fiedler value versus the added number of relays is plotted.

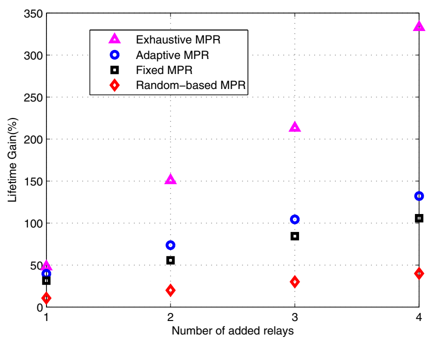

Fig. 3. The average network lifetime gain versus the added number of relays is plotted. Effects of deploying relays is illustrated.

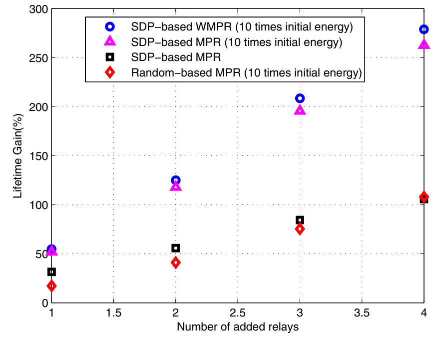

Fig. 4. The average network lifetime gain versus the added number of relays is plotted. Effect of increasing the relays' initial energy 10 times is illustrated.

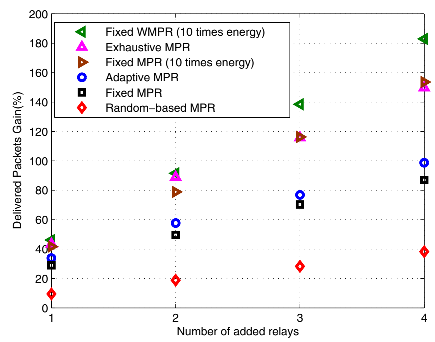

Fig. 5. The average delivered packets gain versus the added number of relays is plotted.

on the Fiedler value of the proposed network maintenance algorithm. Also, we notice that the random addition performs poorly compared to our proposed algorithm.

In Section III, we have chosen the Fiedler value as an intuitive and good measure of the network lifetime, which is our main objective. Fig. 3 depicts the network lifetime gain as a function of the added number of relays. The network lifetime gain due to adding K relays is defined as $G_T(K) = \frac{T(K)-T_{MPR}(0)}{T_{MPR}(0)}$, where $T(K)$ is the network lifetime after deploying K relays and $T_{MPR}(0)$ denotes the network lifetime of the original network utilizing the MPR algorithm. As shown, the proposed SDP-based network maintenance algorithm achieves significant network lifetime gain as the number of added relays increases, which is a direct consequence of increasing the Fiedler value as shown previously in Fig. 2. At $K = 4$ and by employing the MPR algorithm, the proposed network maintenance algorithm achieves lifetime gain of 105.8%, while the random deployment case achieves lifetime gain of 40.09%.

In Fig. 3, we also illustrate the impact of the adaptive network maintenance algorithm on the network lifetime gain. At $K = 4$ relays, the lifetime gain jumps to 132.1% for the MPR algorithm. We also compare the performance of our proposed algorithm with the exhaustive search scheme in Fig. 3. For practical implementation of the exhaustive search scheme, the optimum locations for a given set of relays are determined consecutively, i.e., one relay at a time. We have implemented the exhaustive search scheme by dividing the network area into many small regions and each region is represented by a relay at its center. The optimum location for the first relay is determined by calculating the lifetime of all the possible locations and choosing the one that results in maximum lifetime. Given the updated network including the first relay, we find the optimum location for the second relay via the same exhaustive search scheme. This algorithm is repeated until all the relays are deployed.

As indicated in Section IV-A, the proposed WMPR algorithm should intuitively outperform the MPR algorithm when relays have higher initial energy than that of the sensors. We set the weights of the deployed relays to be 0.1, while the weights of the original sensors to be 1. Therefore, sensors tend to send their data to the deployed relays rather than the neighboring sensors. In addition, the relays' energy are set to be 10 times that of the sensors. As a result, the WMPR algorithm achieves higher gain compared to that achieved by the MPR algorithm as shown in Fig. 4. At $K = 4$, the WMPR and MPR algorithms achieve network lifetime gains of 278.8% and 262.7%, respectively. In Fig. 4, we notice that the difference between the WMPR and the MPR performance curves increases as the number of relays increases. Intuitively, the WMPR algorithm utilizes the relays more frequently than the MPR algorithm. Hence it achieves higher lifetime gain by increasing the the relays' initial energy.

In addition to the network lifetime, the number of the delivered packets from all the sensors to the central unit before the network dies is an important measure of the network performance. Fig. 5 depicts the average delivered packets gain of the various network maintenance algorithms described before. The delivered packets gain due to adding K relays is defined as $G_D(K) = \frac{D(K)-D_{MPR}(0)}{D_{MPR}(0)}$, where $D(K)$ is the number of delivered packets after deploying K relays and $D_{MPR}(0)$ denotes the number of delivered packets for the original network utilizing the MPR algorithm. At $K = 4$ relays, it is shown that the delivered packets gains are 86.99% for the MPR algorithm. Moreover, considering the adaptive network maintenance algorithm results in delivered packets gain of 98.74% for the MPR algorithm. Furthermore, increasing the relays' initial energy 10 times increases the delivered packets gain to 153.6% and 183% for the MPR and WMPR algorithms, respectively.

Finally, we consider the network repair problem where the network is originally disconnected. In Fig. 6, we show the average number of added relays required to reconnect a disconnected network. n sensors are randomly distributed in 6×6 square area. The maximum transmission power of any node is $P_{max} = 0.07$. It is shown that for a disconnected network of $n = 25$ nodes deployed randomly in 6×6 area, the average number of added relays is 4. For $n < 15$, Fig. 6 depicts that the average number of added relays increases as n increases. This is because for small n, it is more likely that the added sensors will be deployed in new regions where there are very few or no sensors. Thus, more relays need to be deployed to connect these added sensors. On the other hand, as n increases beyond $n = 15$, the average number of added relays decreases. This is intuitive because as the the number of sensors increases to a moderate state, the network becomes more balanced, i.e., the sensors are uniformly deployed in the whole area. Beyond this moderate state, increasing the number of sensors keeps filling the gaps in the network. Consequently, the average number of needed relays decreases as n increases.

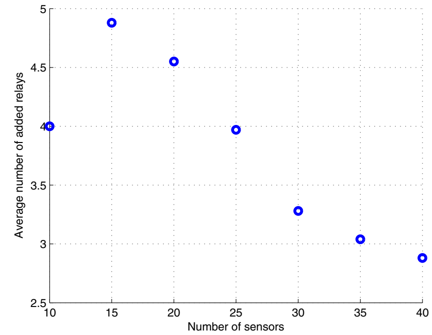

Fig. 6. The average minimum number of added relays required to reconnect a network versus the number of sensors in the network is plotted.

VII. Conclusion

In this paper, we have addressed the problems of network maintenance and network repair in wireless sensor networks. We have considered the Fiedler value, which is the algebraic connectivity of a graph, as a network health indicator. First, we have proposed a network maintenance algorithm, which finds the near-optimum locations for an available set of relays that results in the maximum possible Fiedler value. This algorithm finds the near-optimum location through a small number of levels. In each level, the network maintenance problem is formulated as a semi-definite programming (SDP) optimization problem, which can be solved efficiently in polynomial time using any SDP solver. In a sensor network of $n = 20$ sensors deployed in a 6×6 area, the network lifetime has increased by 105.8% due to the addition of 4 relays.

Second, we have proposed an adaptive network maintenance algorithm, where the relays' locations can be changed depending on the network health indicator. We have shown that a lifetime gain of 132.1% is achieved due to the proposed adaptive network maintenance algorithm. Third, we have proposed the Weighted Minimum Power Routing (WMPR) algorithm, which balances the load of the network among the sensors and the relays. By increasing the relays' initial energy 10 times, we have shown that the WMPR algorithm achieves network lifetime gain of 278.8% when 4 relays are deployed, while the MPR achieves 262.7%. Finally, we have proposed an iterative network repair algorithm, which finds the minimum number of relays needed to connect a disconnected network.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Communications Magazine, vol. 40, pp. 102-114, Aug 2002.
[2] C. Pandana and K.J.R. Liu, "Maximum connectivity and maximum lifetime energy-aware routing for wireless sensor networks," Proc. IEEE Global Telecommunications Conference (Globecom'05), vol. 2, pp. 1034-1038, Nov. 2005.
[3] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, "Prolonging sensor network lifetime with energy provisioning and relay node placement," Proc. IEEE Sensor and Ad Hoc Communications and Networks (SECON'05),pp. 295-304, Sep. 2005.
[4] A. S. Ibrahim, K. G. Seddik, and K.J.R. Liu, "Improving connectivity via relays deployment in wireless sensor networks," Proc. IEEE Global Telecommunications Conference (Globecom'07), pp. 1159-1163, Nov. 2007.
[5] T. Himsoon, W.P. Siriwongpairat, Z. Han, and K.J.R. Liu, "Lifetime maximization via cooperative nodes and relay deployment in wireless networks," IEEE Journal of Selected Areas in Communications, Special Issue on Cooperative Communications and Networking, vol 25, no 2, pp. 306-317, Feb. 2007.
[6] Z. Han, A. L. Swindlehurst, and K.J.R. Liu, "Smart deployment/movement of unmanned air vehicle to improve connectivity in MANET ," Proc. IEEE Wireless Communications and Networking Conference(WCNC'06), vol. 1, pp. 252-257, Apr. 2006.
[7] N. Li and J. C. Hou,"Improving connectivity of wireless ad hoc networks," Proc. The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'05), pp. 314-324, Jul. 2005.
[8] J. G. Proakis, *Digital communications,* 4th ed., McGraw-Hill, 2000.
[9] M. Fiedler,"Algebraic connectivity of graphs," Czechoslovak Lathematics Journal, pp. 298-305, 1973.
[10] A. Ghosh and S. Boyd, "Growing well-connected graphs," Proc. IEEE Conf. on Decision and Control (CDC'06), pp. 6605-6611, Dec. 2006.
[11] B. Mohar, "Some applications of Laplace eigenvalues of graphs," In G. Hahn and G. Sabidussi, editors, Graph Symmetry: Algebraic Methods and Applications, vol. 497 of NATO ASI Series C, pp. 227-275. Kluwer, Dordrecht, 1997.
[12] S. Boyd, "Convex Optimization of Graph Laplacian Eigenvalues," Proc. International Congress of Mathematicians, 3:1311-1319, 2006.
[13] SDPA-M package, Available online at: http://grid.r.dendai.ac.jp/sdpa/
[14] D. Bertsekas and R. Gallager, *Data networks, 2nd ed.*, Prentice Hall, 1991.