The optimum sensor redeployment scheme using the most frangible clusters set

Chu-Fu Wang\textsuperscript{a,}\textsuperscript{*}, Jen-Wen Ding\textsuperscript{b}

\textsuperscript{a} Department of Computer Science, National Pingtung University of Education, No. 4-18 Ming-Shen Road, Pingtung 900, Taiwan
\textsuperscript{b} Department of Information Management, National Kaohsiung University of Applied Sciences, No. 415, Jiangong Road, Kaohsiung city 80778, Taiwan

\begin{abstract}
Sensors redeployment is a straightforward way to increase the lifetime of a Wireless Sensor Network (WSN) by deploying additional sensor devices in the sensing region. This paper considers the problem of finding positions to deploy sensors so that the network lifetime will be the most prolonged when a fixed total quantity of energy of sensor devices is given. We formulated the redeployment problem in WSNs as a network optimization problem called the OCSRP (Optimum Clusters Set Redeployment Problem). Two heuristic algorithms, the MFCSPA-static (Most Frangible Clusters Set Prediction Algorithm for static routing), and the MFCSPA-dynamic are proposed to solve the OCSRP in WSNs under static and dynamic routing schemes, respectively. By integrating the above sensor redeployment algorithms with monitoring facilities, a topology control scheme for maintaining the energy usage of a WSN is also proposed. The simulation results show that both proposed algorithms can find better solutions than other heuristic algorithms. Moreover, both proposed algorithms greatly outperform the other algorithms in the percentage of successfully predicting the earliest energy drain out node.
\end{abstract}

1. Introduction

Due to recent technological advances and the fact that many comprehensive wireless device usages are proposed in diverse scopes, there is an impetus to increase the speed of the development of wireless communication technology. One of the new growing fields of wireless computer networks in recent years is Wireless Sensor Networks (WSNs) [1,2]. A WSN consists of hundreds or thousands of sensor nodes that are usually scattered over a certain area to perform monitoring tasks. The sensor nodes are low-cost, are equipped with limited battery power, and are small sized devices. Besides, they can monitor specific measurement data (e.g. temperature, humidity, and pressure) to detect abnormal events (e.g. a forest fire). Initially the sensor nodes are deployed manually or randomly in the sensing field to form a WSN. The end user may connect to a data concentration center called the sink node (or base station) via Internet or Satellite to send a request for querying a specific event or phenomenon on this sensing field. Once a sensor node detects an unusual situation as designated by the sink node, it has to quickly propagate such an event occurred message to notify the end user. The WSN may also have fault-tolerance capability; that is, if the sensing data is lost, sensor nodes may recover the lost data from their inner caches. The end user will then eventually receive the reported information correctly from the sink node.

The WSNs not only apply in conventional work, but can also overcome difficult missions such as battlefield surveillance, weather monitoring, ecology tracking, and even universe discovery. Since WSNs are in widespread use, many researchers have paid much attention to this field [3–9]. For more information, we encourage the readers to refer to two survey papers [10,11]. Since the WSNs are usually deployed for use in harsh environments, the sensor nodes are not easily to be replaced or recharged when they run out of battery power. Thus, the most important research issue in the area of WSNs is how to conserve the limited battery energy to maximize the network lifetime. Several research topics have been broadly discussed to fulfill the above objective; for example, designing energy aware routings [12–16], designing scheduling power-saving modes of sensor nodes [17–19], and developing comprehensive power management and topology control methods [20–25], etc. After performing several rounds of monitoring and data relaying, the energy level of the sensor devices in a WSN will become lower and lower, and then the WSN will eventually cease functioning. Proper power management and topology control methods can monitor the energy level of the WSN. If the energy level drops below a given threshold, a warning message will be sent out to notify the supervisors. One of the straightforward methods for supervisors to deal with such a circumstance is the deployment of additional sensor nodes in the sensing field to prolong the residual network lifetime of the...
WSN. The sensor deployment issues can be classified into three categories: the pre-deployment, the movement-assisted deployment and the redeployment schemes. A brief introduction of each is given below.

Sensors pre-deployment scheme: The pre-deployment issues are considered in the construction phase of a WSN. Two important factors, network coverage and network connectivity, are usually considered as constraints [21–23]. In order to reduce the construction costs, the coverage issue aims at how to deploy as few sensor nodes as possible in a sensing field while ensuring that the constructed WSN has no coverage holes (or communication holes). Besides, since some of the sensor nodes may be drained of battery power after executing several rounds of tasks, the WSN will become disconnected. Accordingly, the network connectivity constraint will ensure that the constructed WSN has a certain level of connectivity. The higher the connectivity of a WSN, the more robust it will be, and then the network will possibly continue to be connected when some sensor nodes have exhausted their battery power.

Movement-assisted sensor deployment scheme: Assuming that some sorts of sensor nodes are capable of moving then we can accommodate the energy level of a WSN by moving sensor nodes into lower energy areas. Many lectures [24–27] have focused on this issue. For example, Wang et al. [24] modeled a WSN and the Voronoi diagram, which can detect the coverage hole of a network. They proposed three algorithms (VEC, VOR, and Minimax) to determine how to move the sensor nodes so that the problem of coverage holes will be relieved. Wu et al. [25] considered a balancing problem of a WSN with a Mesh structure. An algorithm called SMART was proposed to determine the moving strategy so that the number of sensor nodes in each square area of the WSN was equal, and the number of moves was minimized.

Sensor redeployment scheme: Deploying additional sensor nodes in a low energy level of a WSN is a straightforward method to increase its network lifetime [28]. The problem of where to deploy these additional nodes in the sensing fields is the key point of this issue. Intuitively, the sensor nodes that are close to the sink will consume much more battery energy than others. Therefore, deploying more sensor nodes into these areas will prolong the residual network lifetime of the WSN. However, the hop count to the sink node is just one of the key factors that affects the performance of the redeployment. The other key factors include the residual battery energy of sensor nodes, the occurrence probability of an event happening in each sensing area, and their routing scheme, etc. These factors will all influence the length of the residual network lifetime. To our best knowledge, little research has investigated the sensor redeployment problem for WSN.

In this paper, we consider the redeployment problem in WSNs, and formulate it as a network optimization problem called the OCSR (Optimum Clusters Set Redeployment Problem) to find the best locations for placing the additional sensor nodes. We propose two heuristic algorithms, the MFCSA-static [Most Frangible Clusters Set Prediction Algorithm for static routing] and the MFCSA-dynamic, to solve the OCSR on a static and a dynamic routing scheme, respectively. We also propose a sensor topology control scheme, which synthesizes the above methods to keep track of the WSN’s energy level. The remainder of this paper is organized as follows. Section 2 describes the network model and the formulation of the OCSR. Section 3 proposes the solution methods for solving the OCSR. In Section 4, we will give the simulation results for these algorithms. Finally, the concluding remarks are given in Section 5.

2. The network model and problem formulation

In this section, we firstly describe the network model of a two-tiered hierarchical WSN under consideration in this paper, and then formulate a sensor node redeployment problem based on this network architecture.

2.1. Network model

As a hierarchical structure, the cluster-based WSN is known to be an efficient architecture for organizing a large amount of sensor nodes. Therefore, several hierarchical routing protocols [39,30] have been proposed for this structure. These studies have revealed that the cluster-based WSN performs better, not only in terms of energy conservation for message routing, but also because it is easier for topology controlling than a flat WSN. A cluster-based WSN partitions the sensor nodes into a set of clusters. Each cluster contains sensor nodes which are close to each other, and one of the sensor nodes in each cluster is elected as a cluster head to be responsible for information relaying. Assuming an event randomly happens in a cluster’s sensing region, this event will be notified to the sink node by the following steps. First, the detected sensor node will transmit the event reporting message directly to its cluster head. Then the cluster head will relay the received message, hop by hop, toward the sink node along a predetermined routing path. The time interval from the start of the event to the message being correctly received by the sink node is called a successful event transmission round. We assume that a new event transmission round cannot be started before the event reporting data has been received by the sink node in the previous round.

A cluster-based WSN can be modeled as an undirected graph, called the cluster graph \( G = (V, E) \), where \( V \) denotes the collection of clusters, and \( E \) represents the possibility of directly communicating between clusters. That is, if both clusters \( u \) and \( v \) can communicate directly with each other, then \((u, v) \in E\). A residual energy function \( r: V \rightarrow R^+ \) maps each cluster node \( v \) to a positive real number \( r(v) \) to represent cluster \( v \)'s total residual battery power. Note that, since the sink node \((s) \in V\) is capable of using a charged power line, we thus assume that it has infinite battery power, i.e. \( r(s) = \infty \). Besides, in order to represent the possibility of an event happening in each cluster, an event occurrence probability mass function \( p: V \rightarrow [0, 1] \) associates each cluster node \( v \in V \) with a probability value \( p(v) \) to represent the possibility. Note that \( p(s) = 0 \) and \( \sum_{v \in V} p(v) = 1 \). If there is no hot-zone for an event occurring in the sensing region, then the function \( p \) will become a uniform distribution, which excludes the sink node \( s \); that is \( p(v) = \frac{1}{|V|}, \quad \forall v \in V - \{s\} \). In contrast, the function \( p \) is a non-uniform distribution. Given an event occurring sequence for cluster node \( v_k \) with \( k \in \{1, 2, ..., n\} \) with respect to a function \( p \), where \( v_k \in V - \{s\} \) indicates that an event will occur in cluster node \( v_k \) at the \( k \)'th event transmission round. Notice that the above event occurring sequence is designed to ease the discussion of the network model, and it will not be known in advance. The event transmission will be performed in rounds according to an event occurring sequence until a cluster node runs out of its battery power, and then the WSN will die. The number of event transmission rounds that the WSN can function well is defined as the residual network lifetime.

Fig. 1 shows an example of a two-tiered cluster-based WSN. The lower layer of this figure shows the physical topology of the WSN. The sensor nodes in each circle form a cluster, and each cluster corresponds to a node in the upper layer, which is an abstraction of the physical architecture. Assuming that a sensor node in cluster \( F \) detects an event occurring, then it will immediately emit a message to notify its cluster head. After the cluster head of \( F \) receives
the reported data, it will relay the data along a predetermined path in the cluster graph, say path $F \rightarrow E \rightarrow C \rightarrow S$. Now, considering the battery energy consumption in this transmission round, the cluster nodes $F$, $E$, and $C$ will spend their battery energy for data transmitting. Moreover, cluster nodes $E$, $C$, and $S$ will also consume their energy for receiving data from their predecessor cluster nodes in this path. In our work, the energy depletion of wireless communication follows the first order radio model [3]. This radio model assumes that driving the transmitter or receiver circuitry takes $E_{elec} = 50 \text{ nJ/bit}$ and the transmit amplifier costs $E_{amp} = 100 \text{ pJ/bit/m}^2$.

The energy consumption of transmitting a $k$-bits message for a distance $d$ is given by:

$$E_{tx}(k, d) = E_{tx-elec}(k) + E_{tx-amp}(k, d) = E_{elec} \cdot k + E_{amp} \cdot k \cdot d^2$$  \hspace{1cm} (1)

and receiving this $k$-bits message is given by:

$$E_{rx}(k) = E_{rx-elec}(k) = E_{elec} \cdot k$$  \hspace{1cm} (2)

For ease of discussion, we use $E_{consumption}(\cdot)$ to denote the total energy consumed for message relaying on cluster node $v$, in case the event occurred on cluster node $u$. As shown in Fig. 1, the routing path from $F$ to $S$ is $F \rightarrow E \rightarrow C \rightarrow S$. Then, $E_{consumption}(F) = E_{tx}(k, d(F, E))$, where $d(F, E)$ denotes the distance between cluster nodes $F$ and $E$. Since cluster nodes $E$ and $C$ have to carry out both the tasks of receiving a message from their predecessors and transmitting this message to their successors, $E_{consumption}(E) = E_{tx}(k) + E_{rx}(k, d(E, C))$ and $E_{consumption}(C) = E_{tx}(k) + E_{rx}(k, d(C, S))$. For the other cluster nodes not in the routing path, due to the fact that they need not spend any battery energy during this event transmission round, $E_{consumption}(A) = E_{consumption}(B) = E_{consumption}(D) = 0$.

The message transmission routing protocols for WSNs can be classified into a static routing and dynamic routing, depending on whether the route in each transmission round is fixed or not. Fig. 2 gives an example to illustrate the difference between these two routing schemes. In this figure, the routing path labeled $k$ represents the message transmission route for the $k$th event transmission round. Assuming the event occurring sequence is $(F, C, F, A, B, \ldots)$, notice that both of the event occurrence positions for the first and the third round are in the same cluster node $F$. For the static routing scheme (see Fig. 2(a)), path 1 and path 3 stay the same, since the routing is static. However, for the dynamic routing case (see Fig. 2(b)), the routing algorithm may determine their route according to some variable parameters (e.g., the current traffic volume or nodes’ residual energy), then the determined route may be altered during each event transmission round. Consequently the routes with respect to the same event occurrence cluster nodes in different rounds may not be the same.

The event occurrence cluster node sequence: $(F, C, F, A, B, \ldots)$

$$r_{[1]}(S) = r_{[1]}(V) + c \cdot E_{supply}, \text{ where } c \text{ denotes the multiplicity of } v \text{ in } U$$  \hspace{1cm} (3)

We use cluster graph $G$ of Fig. 1 as an example. Assuming the redeployment budget is 3 units of energy $E_{supply}$ (e.g., $E_{supply} = 50 \text{ mJ}$) and the energy redeployment set $U = \{B, D, D\}$, then the resulting residual energy function $r_v$ with respect to the energy redeployment set $U$ will become, $r_{[1]}(A) = 100$, $r_{[1]}(B) = 200$, $r_{[1]}(C) = 200$, $r_{[1]}(D) = 190$, $r_{[1]}(E) = 250$, and $r_{[1]}(F) = 300$.

Then, our considered probabilistic based network optimization problem, called the Optimum Clusters Set Redeployment Problem (OCSR), is to determine an optimum energy redeployment set $U^*$ for deploying these additional sensor nodes into the WSN, such that the residual network lifetime will be prolonged as much as possible; that is, $L(G, r_{[1]}, p) \geq L(G, r_{[2]}, p)$, for any feasible energy redeployment set $U$. Note that since the residual network lifetime of a WSN is determined by the event occurring sequence for cluster nodes, finding the exact optimum solution for OCSR will not be possible except when the event occurrence sequence is given in advance. We will propose heuristic algorithms for the OCSR and demonstrate their performance through simulations. In our considered problem, we only determine which clusters are for deploying the additional sensor devices. The exactly positions for placing

![Fig. 1. An example for cluster-based WSN.](image1.png)

![Fig. 2. The categories of routing protocols.](image2.png)
them in each determined cluster are not considered in this research.

3. The Sensor redeployment scheme

In general, when finding the positions for the redeployment of the additional sensor nodes, one may intuitively choose the lower energy value cluster nodes or the weak cluster nodes in the network topology with respect to certain evaluation metrics such as the network reliability or the fault tolerance, etc. Till now, many solutions have been proposed to fulfill these objectives. For example, finding the most vital node (edge) in a network [31, 32] is one of the attractive problems. The most vital node (edge) is defined as the node (edge) whose removal will minimize the network reliability when compared with other nodes (edges) in the network. However, using the solutions of these optimization problems as our redeployment positions might not be a good choice, because the traditional problems fail to consider the following two factors, which will certainly affect the objective value (the residual network lifetime) of the OCSR. The first one is the event occurrence probability mass function \( p \). Notice that in our consideration of the network model, the position of the occurrence of an detected event in each transmission round is not deterministic, but just happens by chance. The cluster nodes where detected events frequently occur will thus consume much more battery energy than other cluster nodes in the WSN. Besides, the routing scheme for message delivery is also another major factor that traditional problems fail to consider in the energy consumption of our network model. Both factors will be taken into consideration in our solution methods to gain better performance.

Since the residual network lifetime \( L(G, r, p) \) is stochastic, it is not easy to predict this value correctly. The job of predicting the best energy redeployment set will thus become harder. In order to cope with the variation in such a stochastic problem, we will use an expectation-based prediction approach to estimate the value \( L(G, r, p) \). Then, based on the estimation of the \( L(G, r, p) \), a cluster node set called the most frangible cluster set in cluster graph \( G = (V, E) \) is proposed as the redeployment positions under both the static and dynamic routing scheme, respectively. Finally a sensors topology control scheme is then developed.

3.1. The static routing case

Given a cluster graph \( G = (V, E) \), a residual energy function \( r \) and an event occurrence probability mass function \( p \), for each cluster node \( v \in V \), we let path \( P_{s,v} \) be the static route for message delivery from cluster node \( v \) to the sink node \( s \) in each transmission round. Note that one of the frequently used paths for static route is the shortest path. If we regard the Euclidean distances between the pair of cluster nodes in each edge as the link cost, then the static route with respect to each cluster node can be easily obtained by using the Dijkstra’s algorithm. Since the message delivery path with respect to each cluster node \( v \in V \) is determined and is static in each event transmission round, the energy consumption quantities \( E_{\text{consumption}}(v) \), \( \forall u, v \in V \) can be easily obtained using Eqs. (1) and (2) in Section 2. Due to the fact that the event occurrence probability of cluster node \( u \) is \( p(u) \), \( p(u) \cdot E_{\text{consumption}}(v) \) is equal to the energy consumption rate of cluster node \( v \) in an event transmission round when the event occurs at cluster node \( u \). Therefore, the Expectation of Energy Consumed of cluster node \( v \in V \), denoted by \( EEC(v) \) in one event transmission round can be defined as follows:

\[
EEC(v) = \sum_{u \in V} p(u) \cdot E_{\text{consumption}}(v)
\]

and the Expectation of Residual Lifetime of cluster node \( v \), denoted by \( ERL(v) \), can be obtained by:

\[
ERL(v) = \frac{r(v)}{EEC(v)}
\]

In the example of Fig. 1, since the routing paths that pass through node \( C \) are \( P_{s,C}, P_{S,C}, P_{c,C} \), and \( E_{C} \), the expectation of energy consumed with respect to node \( C \) (EEC(C)) in one transmission round according to Eq. (4) is shown as follows:

\[
EEC(C) = \frac{1}{6} \cdot E_{\text{consumption}}(C) + \frac{1}{6} \cdot E_{\text{consumption}}(C) + \frac{1}{6} \cdot E_{\text{consumption}}(C) + \frac{1}{6} \cdot E_{\text{consumption}}(C) + \frac{1}{6} \cdot E_{\text{consumption}}(C) + \frac{1}{6} \cdot E_{\text{consumption}}(C)
\]

Thus, using Eqs. (4) and (5), the expectation of energy consumed in one event transmission round, and the expectation of residual lifetime for each cluster node can then be obtained. Note that the network lifetime is defined as the time at the first cluster node dies; thus we can use the minimum value of the residual expectation lifetime among all cluster nodes as the residual network lifetime of \( G \), denoted by \( ERL(G) \). That is, \( ERL(G) = \min_{v \in V} ERL(v) \). We call the cluster node with the minimum value of expectation residual lifetime the most frangible cluster node.

Assuming the budget for sensor redeployment is \( k \) units of energy \( [E_{\text{supply}}] \), we now have to determine a best energy redeployment set \( U' \) of cardinality \( k \). Based on the above discussion, the solution method MFCSPA-static for the OCSR is stated as follows. Firstly, compute the EEC(v) and ERL(v) values for each cluster node \( v \in V \) with respect to the given cluster graph \( G \), the event occurrence probability mass function \( p \), and the residual energy function \( r \), using Eqs. (4) and (5). Let \( v' \) be the most frangible cluster node in \( G \); that is, \( ERL(v') = \min_{v \in V} ERL(v) \), then put \( v' \) into set \( U' \) \((U' = U' \cup \{v'\})\). Meanwhile, the new residual energy function \( r_{U'} \) with respect to \( U' \) has also to be updated according to Eq. (3). By repeating the above two steps \( k \) times, a complete energy redeployment set \( U' \), which consists of the most frangible cluster nodes can then be obtained. We call set \( U' \) the most frangible cluster set.

The complete Most Frangible Cluster Set Prediction Algorithm for static routing scheme (MFCSPA-static) is shown in Fig. 3.

3.2. The dynamic routing case

In general, using a dynamic routing scheme on WSNs for event reporting has the merit of balancing the energy consumption within networks so that their residual network lifetime will be prolonged, compared to the static routing scheme. Most of the proposed energy-aware routing protocols [12–15] for WSNs belong to the dynamic routing case. Among these studies, Huang and Jan [14] proposed an energy-aware, load balanced routing method, called the Maximum Capacity Path (MCP) routing, which can conserve battery energy usage better than many other proposed algorithms. In this paper, we will use the MCP routing as the underlying routing scheme to illustrate our proposed solution method in the dynamic routing case. Note that it is easy to modify our proposed redeployment method to integrate other existing dynamic routing schemes into the algorithm. In the following subsections, the MCP routing will be introduced first, followed by our proposed algorithm.

3.2.1. An example of a dynamic routing scheme – the Maximum Capacity Path (MCP)

Let \( G = (V, E) \) be a cluster graph, and \( r \) be the residual energy function that is associated with cluster node set \( V \). Let level \( L \) with respect to cluster node \( v \in V \) denote the minimum hop count from \( v \) to the sink node \( s \). The MCP firstly constructs \( G \) into a layered...
Algorithm MFCSPA-static\((G, r, p, k)\);

Input:
\(G\): the cluster graph;
\(r\): the residual energy function;
\(p\): the event occurrence probability mass function;
\(k\): the number of energy units to be allocated;

Output:
\(U^*\): an energy redeployment set

\{*/ compute the \(E_{\text{consumption}}(w)\) values */
for each \(v \in V\) do
for each \(w \in V\) do
let \(P_{vs}\) be the routing path from \(v\) to \(s\),
successor\((w)\) denotes the successor node of \(w\) in \(P_{vs}\),
and \(l\) denotes the message length;
\(E_{\text{consumption}}(w) = \begin{cases} E_T(l, d(w, \text{successor}(w))) & \text{if } v = w \\
E_{R}(l) + E_T(l, d(w, \text{successor}(w))) & \text{if } v \neq w \text{ and } w \in P_{vs} \\
0 & \text{otherwise} \end{cases}\)
\};
\(U^* = \emptyset\);
\(r_{v^*} = r\);
for \(s = 1\) to \(k\) do
\{*/ compute the EEC and ERL values for each cluster node */
for each \(v \in V\) do
\(\text{EEC}(v) = \sum_{u \in V} p(u) \cdot E_{\text{consumption}}(v)\);
\(\text{ERL}(v) = \frac{\text{EEC}(v)}{E_{\text{consumption}}(v)}\);
\};
let \(v^*\) be the cluster node such that \(\text{ERL}(v^*) = \min_{v \in V} \text{ERL}(v)\);
\(U^* = U^* \cup \{v^*\}\);
reset \(r_{v^*}\) according to \(U^*\);
\};
output(\(U^*\));
\}

Fig. 3. The complete algorithm for the MFCSPA-static.

network \(G\). The layered network \(G\) is a resulting directed subgraph of \(G\) after removing the edges \((u, v) \in E\), such that \(L_u = L_v\). Besides, a directed edge \((u, v) \in E(N)\), then \(L_u = L_v + 1\). As shown in Fig. 4(a), the value that is associated with each cluster node \(v\) is its residual battery energy \(r(v)\), and the level of each cluster node is shown at the top of this figure. For example, the level of cluster node \(d\) is 2, and the level of both cluster nodes \(f\) and \(g\) is 4. Moreover, since \(L_u = L_v = 1\) and \(L_f = L_g = 4\), edges \((a, b)\) and \((f, g)\) will be removed from \(G\), and then the resulting layered network \(N\) with respect to \(G\) is shown in Fig. 4(b).

Let \(P_{vs}\) denote a path from cluster node \(v\) to \(s\). The capacity with respect to path \(P_{vs}\) is defined as the minimum residual energy value in this path, and denoted by \(c(P_{vs})\). That is, assuming path \(P_{vs} = v, u_1, u_2, \ldots, u_k, s\), then \(c(P_{vs}) = \min(r(v), r(u_1), r(u_2), \ldots, r(u_k), r(s))\). Given two specific cluster nodes \(v\) and \(s\), the maximum capacity path \(P_{vs}\) is defined as the path which has the maximum capacity value among all \(v \rightarrow s\) paths. For example, in Fig. 4(b), there are three paths from \(e\) to \(s\) in \(N\); they are \(P_{es} = e, c, a, s\), \(P_{es} = e, c, b, s\), and \(P_{es} = e, d, b, s\). Since \(c(P_{es}) = 40\), \(c(P_{es}) = 5\), and \(c(P_{es}) = 5\), the maximum capacity path between cluster nodes \(e\) and \(s\) is \(P_{es}\). For detailed steps to select the minimum capacity paths from \(N\), one can refer to the paper of Huang and Jan [14]. Note that if we collect the maximum capacity paths that start from each cluster node \(v \in V\) to the sink node \(s\), a tree shaped topology will then be obtained (see Fig. 4(c)). Besides, in the case of the residual energy function \(r\) being altered, the MCP scheme will obtain a different set of maximum capacity paths. In a later subsection, this set of maximum capacity paths will be adopted for event reporting in each event transmission round.

3.2.2. The solution method

The residual energy function will alter after performing each event reporting task; consequently the set of maximum capacity paths is changed accordingly. Besides, the expectation of residual lifetime of a cluster node that is defined by Eq. (5), and the expectation of residual network lifetime for the static routing case, also need to be modified to fit the dynamic routing case. Given a cluster graph \(G = (V, E)\) and a residual energy function \(r\), we use MCP\((G, r)\) to denote the procedure call for determining the new set of the maximum capacity paths with respect to the given residual energy function \(r\). Recall that the expectation of energy consumption for a cluster node \(v \in V\) in each event transmission round is as follows:

\[
\text{EEC}(v) = \sum_{u \in V} p(u) \cdot E_{\text{consumption}}(v)
\]

It can still be used to estimate the expected energy consumption in following transmission rounds, provided that the maximum capacity paths are updated accordingly. The new estimation for value \(\text{ERL}(G)\) can be obtained by repeatedly performing the following two steps. In the first step, we can invoke the procedure call MCP\((G, r)\) to compute the set of current maximum capacity paths \(P_{vr}\) from each cluster node \(v \in V\) to the sink. Based on these paths, the expectation of energy consumption values \(\text{EEC}(v), \forall v \in V\) can then be obtained by Eqs. (1) and (2). Next, update each cluster.
node’s residual energy by subtracting the respective expectation of energy consumption value of this cluster node during this round, respectively. That is, \( r(v) = r(v) - EEC(v), \forall v \in V \). Repeat the above two steps until the first cluster node \( v^* \) runs out of energy, then the number of iteration that has been executed so far is defined as the expectation of the residual network lifetime \( ERL(G) \). Meanwhile, the cluster node \( v^* \) is defined as the most frangible cluster node, which is the first recommended position for redeployment in the dynamic routing case. We then add the cluster node \( v^* \) into the energy redeployment set \( U^* \). After finding the first most frangible cluster node \( v^* \), we restore the residual energy function to its initial function \( r \), and update the residual energy function \( r_U \) according to \( U^* \). Then repeat the above two procedures to find the next most frangible cluster node. Repeat the above finding process until the size (i.e., cardinality) of set \( U^* \) reaches value \( k \). We call the resulting set \( U^* \) the most frangible cluster set for the dynamic routing case. The complete redeployment algorithm for dynamic routing is shown in Fig. 5.

3.3. The sensor energy and topology control scheme

The sensor topology control is an energy usage monitoring scheme of a WSN, which tries to avoid the energy level of a WSN from dropping drastically and causing the network to cease functioning. The sink node has to be responsible for collecting the residual energy value and estimating the event occurrence frequency for each cluster node on a periodic basis. In order to collect this information, one of the methods is to demand that the sink node periodically query cluster nodes to obtain their residual energy values and the event occurrence frequency. However, by doing this the battery energy of the cluster nodes will be quickly drained off, and consequently the network lifetime of the WSN will drop sharply. Instead, the sink node can keep tracking these values through simulating each cluster node’s energy consumption behavior. Detailed operations are discussed as follows.

As a WSN is firstly deployed in a monitoring region, we assume that certain network related information, such as the network topology, the routing scheme, and the initial battery energy value with respect to each of the cluster nodes will be kept in the sink node. In the network operating phase, the cluster nodes will consume their battery energy when performing the following two tasks. The first task is environment sensing, in which the sensor devices in each cluster node will start their sensing facilities to monitor the environment and try to detect abnormal events. This task will spend a constant energy value for each cluster node during each period of time. The second task a sensor node will perform is abnormal event reporting. When an abnormal event occurs within a cluster node’s sensing range, then it will initiate an event occurring message back to the sink node. The cluster nodes that are located in the message reporting path from the detected node to the sink will also consume battery energy when executing the message relaying task. The amount of battery energy consumed with respect to each related cluster node can be easily determined by Eqs. (1) and (2). Thus, as the sink node receives a reporting message, the estimated residual energy values of each node involved in the message reporting path will be decreased by the above predetermined values, respectively. Using such an approach, the sink node can then obtain the approximate residual energy value of each cluster node in the WSN. Besides, the event occurrence probability mass function can also be updated accordingly. As soon as the residual energy function and the event occurrence probability mass function of a WSN are obtained, we can use the residual network lifetime values that were discussed in Subsections 3.1 and 3.2 for both static and dynamic routing cases, as the monitoring parameters. Whenever the residual network lifetime decreases below a certain threshold value, the sink node will inform the supervisors by sending out a warning message to notify them that the WSN is in a low energy level state. If additional sensor devices are available for deployment, the supervisors will transit the sink node from a monitoring state into a redeployment planning state and start the sensor redeployment work.

Note that the budget \( k \) for sensor redeployment may be a predetermined fixed value given by the supervisor. Otherwise, the supervisor can set the redeployment goal that the resulting residual network lifetime has to be greater than a certain value, say \( L \). Then, according to the estimation methods proposed in the previous subsections, it is easy to obtain the least redeployment budget \( k \), such that the resulting residual network lifetime will be greater than \( L \) after the sensor redeployment. Finally, the MFCS-SPA-static or the MFCS-SPA-dynamic algorithm will be invoked to determine the best locations for redeployment. Then the sensor devices can be placed in the regions suggested above. After completing the redeployment task, the sink node will then transit back into the monitoring state to keep tracking the energy level of the WSN. The complete algorithm for the sensor energy and topology control scheme is shown in Fig. 6.

Let us now turn our attention to the computation overhead for using the proposed topology control scheme. At first, during the monitoring state, the major computation overhead is to estimate the residual network lifetime \( ERL(G) \) of the WSN. Since the WSN can be incorporated with any routing scheme for message reporting, for ease of discussion, we now only consider the shortest path routing as our discussed routing scheme for complexity analysis. For the other routing schemes, the discussions are similar to this case so we omit them here. The Dijkstra’s algorithm can be used for obtaining the shortest reporting path \( P_m \), for each cluster node.

---

**Fig. 4.** Example of the MCP selection [14].

---

**Fig. 5.** Level 1-5,
Algorithm MFCSPA-dynamic\((G, r, p, k)\);
Input:
\(G\): the cluster graph;
\(r\): the residual energy function;
\(p\): the event occurrence probability mass function;
\(k\): the number of energy units to be allocated;
Output:
\(U^*\): an energy redeployment set
\[
U^* = \emptyset;
\]
for \(i = 1\) to \(k\) do {
still-alive=true;
reset \(r_{U^*} = r\);
update \(r_{U^*}\) according to \(U^*\);
while (still-alive==true) do {
MCP\((G, r_{U^*})\);
compute the \(E_{\text{consumption}}(w)\) values for every cluster node pair \((v, w)\), \(\forall v, w \in V\) according to the set of maximum capacity paths;
\(E_{\text{EC}}(v) = \sum_{u \in V} p(u) \cdot E_{\text{consumption}}(w), \forall v \in V\);
\(r_{U^*}(v) = r_{U^*}(v) - E_{\text{EC}}(v), \forall v \in V\);
if (there exist a node \(v^* \in V\), such that \(r_{U^*}(v^*) < 0\)) then {
\[U^* = U^* \cup \{v^*\}\]
still-alive=false;
};
};
output\((U^*)\);
\]

Fig. 5. The complete algorithm for the MFCSPA-dynamic.

Algorithm Sensors-energy-and-topology-control\((G, r, p, k, h)\);
Input:
\(G\): the cluster graph;
\(r\): the residual energy function;
\(p\): the event occurrence probability mass function;
\(k\): the number of energy units to be allocated;
\(h\): the threshold for determining whether the WSN is in a low energy state or not;
\{
while (true) do {
/* the monitoring state */
Loop
if (an detected event occurs) {
proceed the event reporting task;
update the residual energy function \(r\) and the event occurrence probability mass function \(p\);
}
re-estimate the residual network lifetime \(ERL(G)\);
until \(ERL(G) < h\);
/* the redeployment planning state */
let \(U^* = \text{MFCSPA-static}(G, r, p, k)\) (or \(U^* = \text{MFCSPA-dynamic}(G, r, p, k)\))
perform the redeployment task according to set \(U^*\);
update the residual energy function \(r\);
}
\}

Fig. 6. The sensors energy and topology control scheme.

\(v\) to the sink, which will cost \(O(n^2)\) computation time, where \(n\) is the number of cluster nodes in the WSN. The energy consumption quantities \(E_{\text{consumption}}(v), \forall u, v \in V\) can then be determined by \(O(n^2)\) computation time. Recall that, the Eqs. (4) and (5) defined the expectation of residual lifetime of a cluster node \(v \in V\) as \(ERL(v) = \frac{r(v)}{P(1/v)}, \) where \(E_{\text{EC}}(v) = \sum_{u \in V} p(u) \cdot E_{\text{consumption}}(v).\) Thus, taking \(O(n^2)\) computation time will obtain all cluster nodes' \(ERL(v)\) values in the WSN.
Since the residual network lifetime of a WSN (ERL(G)) is defined as the minimum value of the residual expectation lifetime among all cluster nodes (that is, \( \text{ERL}(G) = \min_{v \in V} \text{ERL}(v) \)), taking additional \( O(n) \) computation time will obtain this metric. Summing the above computation cost, we have that the computation for the residual network lifetime \( \text{ERL}(G) \) of a WSN in the monitoring state costs \( O(n^2) \). Besides, the computation for the residual energy value and the event occurrence frequency with respect to each cluster node takes only \( O(n) \) time. Therefore, the computation overhead in the monitoring state of the sink node takes only \( O(n^2) \) computation time. Similarly, we have that the computation overhead for the redeployment planning state of the sink node is \( O(k \cdot n^2) \), where \( k \) denotes the budget of the redeployment. On the other hand, due to all of the computations being performed on the sink node and the sink node being able to be charged by a power line, the energy consumption overhead for our proposed topology control scheme will have practically no effect on the network lifetime of the WSN.

4. Simulation results

4.1. Performance metrics and simulation setup

In this subsection, the performance metrics, environment setup, and the compared heuristic algorithms are described. The environment setup of our simulation follows the work of [14]. The network topologies used in the simulations were generated by randomly distributing \( n \) nodes over a \( 100 \times 100 \text{m}^2 \) square coordinate grid. In addition, the radio transmission range of each node is set to 25 meters, and the sink node is located at coordinate \((0, 0)\). For each of the following cases, \( n = 30, 40, 50, 60, 70, 80, 90, 100, 110 \) and 120, fifty cluster graphs were generated for simulation. Note that we only use the connected graphs for simulation. If a cluster graph is not connected, then it be discarded. The initial battery energy of each cluster node follows the uniform distribution in \([250, 550]\). In our simulation, two categories of probability mass function \( p \) for event occurrence were considered. They were uniform distribution and non-uniform distribution. For the uniform case, we set \( p(v) = \frac{1}{|V - \{s\}|} \) \( \forall v \in V - \{s\} \) and \( p(s) = 0 \). For the non-uniform case, we first randomly partitioned the \( n - 1 \) cluster nodes into 4 equal sized groups. And then, we let the ratio of the event occurrence probability for each cluster node between these four groups is 4:3:2:1. Therefore, for each cluster node \( v \in V - \{s\} \) in group \( i \in \{1, 2, 3, 4\} \), we set \( p(v) = \frac{4}{9} \) and \( p(s) = 0 \).

For each given connected cluster graph \( G = (V, E) \) with a residual energy function \( r \) and an event occurrence probability mass function \( p \), 10 event occurring sequences for cluster node were generated. Thus, there were totally 500 problem instances made for simulations with respect to each cluster node number \( n \). The event occurring sequences consist of cluster nodes, which indicate the position of an detected event occurring in each event transmission round. The cluster node in the event occurring sequence carries out the event reporting tasks sequentially, until the first cluster node dies. The budget for redeployment \( k \) that was considered in our simulations was set to \( k = 5 \). In our simulations, two performance metrics were considered. The first one is the residual network lifetime, which indicates how the evaluated algorithms perform on the lifetime extension after the redeployment. The other metric is the number of successful hits for predicting the first dying cluster node that belongs to the redeployment set \( U' \). Note that the higher the value of successful hits, the more robust the algorithm will be for predicting when the first cluster node will die.

In our simulation, our proposed algorithms were compared to two competitive greedy-based heuristic algorithms (the MENPA and the MDNPA). The descriptions of these two compared algorithms are as follows.

(1) The Minimum Energy Node Prediction Algorithm (MENPA)

Assuming the budget for redeployment is \( k \), the MENPA firstly selects the least residual energy cluster node \( v' \in V \); that is, \( r(v') = \min_{v \in V} r(v) \), and then put node \( v' \) into an energy redeployment set \( U' \) \((U' = U' \cup \{v'\})\). After that, update the residual energy function \( r_U \) with respect to set \( U' \). Repeat the above steps \( k \) times, then the complete energy redeployment set \( U' \) for the MENPA will be obtained.

(2) The Maximum Degree Node Prediction Algorithm (MDNPA)

Let \( v' \in V \) be the maximum degree node of cluster graph \( G = (V, E) \); that is, \( \text{deg}(v') = \max_{v \in V} \text{deg}(v) \). Then the MDNPA will deploy total \( k \times C_{\text{supply}} \) battery energy into the cluster node \( v' \).

In order to demonstrate the performance of our proposed algorithms, we not only provide the above two compared algorithms’ results and the optimum solution values for the static routing case also. The optimum values can be obtained by assuming the event occurring sequences are known in advanced. Since the computation complexity for determining the optimum values in the dynamic routing case are very high, we only have shown the optimum solution value for the static routing case.

4.2. Numerical results

For the static routing case, Figs. 7 and 8 show the performance results of the residual network lifetime metric for \( k = 5 \). The event occurrence probability mass functions are uniform and non-uniform, respectively. As shown in these results, our proposed method, MFCSPA-static, outperforms the other heuristic algorithms for both cases and the results are very close to the optimum values. Moreover, the resulting curves of the heuristic algorithms (MENPA and MDNPA) in these figures are very close to the curve of the original, which is the result that does not perform the redeployment. The performance results of the number of successful hits metric for \( k = 5 \), and the event occurrence probability mass functions for both uniform and non-uniform cases are shown in Figs. 9 and 10, respectively. These figures show that for over 70% of the simulation instances, our proposed method successfully predicts the cluster node that died first for both routing cases. As shown in these figures, our proposed algorithm significantly outperforms the other heuristic algorithms. Fig. 11 gives the results of comparisons between the evaluated algorithms, in terms of residual network life-

![Fig. 7. Residual network lifetime vs. number of cluster nodes using uniform distribution (with k = 5 and static routing).](image)
time versus redeployment budget (the percentage of the number of cluster nodes), for \( n = 70 \) and the event occurrence probability mass function is non-uniform distribution. The results show that the MFCSPA-static performs much better than others when the budget of redeployment increases. Fig. 12 gives the results of comparisons between the evaluated algorithms, in terms of number of successful hits versus redeployment budget, for \( n = 70 \) and the event occurrence probability mass function is non-uniform distribution.

For the dynamic routing case, Figs. 13 and 14 show the performance results of the residual network lifetime metric for \( k = 5 \), where the event occurrence probability mass functions are uniform and non-uniform cases, respectively. Figs. 15 and 16 show the performance results of the number of successful hits for \( k = 5 \), where the event occurrence probability mass functions are uniform and non-uniform cases, respectively. These results demonstrate that our proposed algorithms perform better than other heuristic algorithms. The results also show that the MFCSPA-dynamic is more suitable than the MFCSPA-static under the dynamic

Fig. 8. Residual network lifetime vs. number of cluster nodes using non-uniform distribution (with \( k = 5 \) and static routing).

Fig. 9. Number of successful hits vs. number of cluster nodes using uniform distribution (with \( k = 5 \) and static routing).

Fig. 10. Number of successful hits vs. number of cluster nodes using non-uniform distribution (with \( k = 5 \) and static routing).

Fig. 11. Residual network lifetime vs. redeployment budget using non-uniform distribution (with \( n = 70 \) and static routing).

Fig. 12. Number of successful hits vs. redeployment budget using non-uniform distribution (with \( n = 70 \) and static routing).
routing scheme. Finally, we also made comparisons between the evaluated algorithms, in terms of residual network lifetime versus redeployment budget and in terms of the number of successful hits versus redeployment budget, for both uniform and non-uniform event occurrence probability distributions. The performance results are shown in Figs. 17 and 18, respectively.

5. Concluding remarks

In this paper, we considered a redeployment problem in WSNs, which is formulated as a network optimization problem, called the OCSRIP. We proposed two heuristic algorithms (MFCSPA-static and MFCSPA-dynamic) to find the approximate solution for this problem under static and dynamic routing schemes, respectively. In the experiments, it was shown that our proposed algorithms significantly outperform other competitive heuristic algorithms and the performance results for the MFCSPA-static are very close to the optimum values. In addition, the MFCSPA-dynamic receives a higher number of successful hits, and prolongs the network lifetime much longer than the MFCSPA-static under the dynamic routing scheme.
Fig. 18. Number of successful hits vs. redeployment budget using non-uniform distribution (with $n = 70$ and dynamic routing).

routing scheme. In the future, we will consider the optimum pre-deployment problem under our consideration of the network environment and assumptions.

References