Optimized Sink Mobility for Energy and Delay Efficient Data Collection in FWSNs

Sharhabeel H. Alnabelsi, Hisham M. Almasaeid, and Ahmed E. Kamal
Dept. of Electrical and Computer Eng., Iowa State University, Ames, IA 50011, USA
E-mail:{alnabsh,hisham,kamal}@iastate.edu

***Abstract*—Network fragmentation is a potential problem in wireless sensor networks (WSNs) due to many reasons like, node failures or environmental conditions (obstacles) that prevent connected deployments. One approach to cope with this problem is to have a mobile sink node (MS) patrol the network field and collect the data from all the fragments across the network. In this paper, we use a dynamic programming (DP) approach to determine the mobility trajectory of the MS within each fragment such that the energy consumption at the sensor nodes within the fragment is minimized. Moreover, we study the problem of finding the shortest route (cycle) that the MS should take in its journey between fragments in order to reduce a fragment's inter-visit time. For this purpose, we propose an Integer Linear Programming (ILP) formulation to find the optimal route. As finding the optimal route is NP-hard, we also propose a heuristic approach to find a near optimal solution.**

I. Introduction

Mobility has been used in wireless sensor networks, by many researchers [1], for different objectives including prolonging the network lifetime [2], [3] and enhancing the connectivity of sparse networks [4], [5]. Although most of the current research attributes network disconnection to the sparse nature of the network, recent studies had turned the focus towards a new form of network disconnection, namely network fragmentation [6], [7]. In such form of disconnection, a WSN is partitioned into several sub-networks (usually referred to as fragments, where the fragment is a connected group of sensor nodes). Network fragmentation could be caused by node failures, due to physical damage or energy depletion, having a fragmented area of phenomenon, i.e., sensing coverage for the entire field is not needed, or by the existence of physical obstacles and restrictions that may forbid a connected deployment. This fragmentation makes the data dissemination process very challenging due to network disconnection. A viable solution in this case is to have a mobile sink (MS) patrol the network field and collect data from all fragments, and then relay it to a gateway server (or a central processing unit) through either long range communication or short-range communication (which requires the MS to get close enough to the central unit). The idea of sink mobility [8], [9] has been proposed in the literature as a data collection method to prolong the network lifetime by reducing the energy spent by static sensors to relay traffic.

The closest work in literature to this study is the work done by the authors in [6] and [7]. In these two papers, a number of mobile agents that could be data relays, data collectors, or data sinks, are used to recover the operation of a fragmented wireless sensor network (FWSN). A data relay would forward the data between adjacent fragments until an end-to-end path is established, over time, between any fragment and the fragment in which the sink node resides. A data collector, on the other hand, would physically carry the data from a fragment to the sink node. The third approach is to have the sink node itself be mobile and collect data from all fragments. Under any of the three approaches, the mobile agent stops at a predetermined point, called the service center, outside a fragment and collects the data through adjacent sensor nodes. As the main objective in those two papers was the evaluation of the end-to-end delay, through queueing network models, the energy consumption was not addressed. Having the mobile agent stop at a single point every time it collects the data from a fragments, means that the sensor nodes that are closer to that point will experience higher energy depletion rates, which will worsen fragmentation situation in the network. Therefore, in this paper, we focus on both energy consumption and delay. We propose that the mobile sink goes into the fragment and collects data from a number of cluster heads in the fragment to balance the energy consumption. To find the optimal movement route (in terms of total energy consumption) inside a fragment, we propose a dynamic programming solution. Furthermore, we formulate the problem of finding the shortest MS route between fragments, i.e., the minimum delay (MS inter-visit time) route, as an integer linear program. As finding the optimal route is NP-hard, we also propose a heuristic approach to find a near optimal solution.

The rest of this paper is organized as follows. The system model is explained in Section II. In Section III, we present the dynamic programming solution to find the optimal MS mobility within a fragment. Finding the shortest route for the MS to visit all fragments is studied in Section IV. In Section V, we evaluate the performance of the proposed solutions through simulation. We conclude the paper in Section VI.

II. System Model

Before proposing any solutions to resume the interrupted, due to fragmentation, data delivery process in a WSN, there is a need to first detect fragmentation and identify fragments. The problem of fragmentation detection and fragment identification is beyond the scope of this paper. We assume that fragments are identified, in terms of location and shape, using any of the existing approaches in the literature [10], [11], [12], [13], and provided as an input.

Throughout the paper, we assume that a single mobile sink (MS) node exists in the network, and is responsible for data collection from isolated fragments. In this paper, we address two mobility optimization problems: First, *optimizing the movement of the MS inside a fragment in terms of power consumption; we call this Intra-fragment mobility optimization*. Second, *optimizing the movement of the MS between fragments in terms of total patrol time; we call this Inter-fragment mobility optimization*.

a) Intra-Fragment Mobility Optimization: Finding the optimal movement pattern (route) that the MS should take inside a fragment to minimize the energy consumption depends on the coordinates (on the fragment's circumference) from which the MS enters and exits the fragment. Therefore, for optimal performance, the MS should have a precomputed optimal route for any combination of entrance and exit coordinates. Taking every point on the fragment's circumference into consideration makes the problem intractable. Therefore, to make the problem tractable, each fragment is discretized into square cells, such that the length of the cell side x is set as $x = \frac{R}{\sqrt{5}}$ [14], where R is the communication radius (assumed to be the same of all nodes including the MS). This value guarantees reachability between any pair of sensor nodes in horizontally and vertically adjacent cells. Figure 1 shows an example of this discretization process. The cell granularity depends on R, which also impacts the consumed routing power by sensor nodes and the MS's trip time within a fragment as we will see later.

b) Determining Entrance/Exit Points (EEPs): Based on the discretization process described above, we consider two cases to determine the entrance/exit points for the MS into/from each fragment. In the first case, the MS enters the fragment from any column in the uppermost (or lowest) row, and exits from any column in the lowest (or uppermost) row. The optimal columns are found using the dynamic programming approach we propose later. In the second case, the MS enters the fragment from any row in the leftmost (or rightmost) column, and exits from any row in the rightmost (or leftmost) column. The optimal rows are also found using the dynamic programming approach. The entrance/exit points of the best of the two cases (in terms of power consumption) are chosen as the permanent entrance/exit points (EEPs).

c) Inter-Fragment Mobility Optimization: As for inter-fragment mobility, we are looking for the shortest route (cycle) the MS should take to visit every fragment once during a cycle in which it visits all fragments. To visit a fragment means that the MS must enter the fragment through one of its permanent EEPs (determined as described earlier), follow the optimal intra-fragment route (obtained using the dynamic programming approach), and then leave through the other permanent EEP. The trip on this cycle will be repeated over and over by the MS.

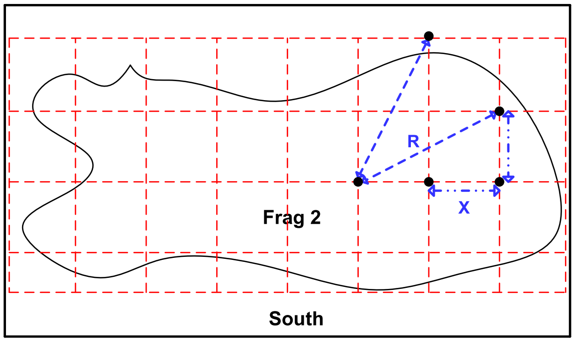
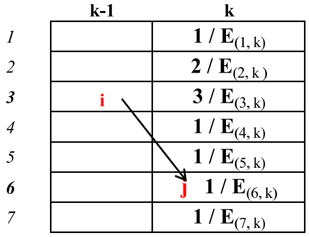


Fig. 1. One fragment discretization in a FWSN topology.



	k-1	k
1		**1 / E**$_{(1,k)}$
2		**2 / E**$_{(2,k)}$
3	i	**3 / E**$_{(3,k)}$
4		**1 / E**$_{(4,k)}$
5		**1 / E**$_{(5,k)}$
6		j **1 / E**$_{(6,k)}$
7		**1 / E**$_{(7,k)}$

Fig. 2. Depletion rates example for DP approach.

III. Intra-Fragment Mobility Optimization

We would like to start this section by presenting the assumptions made in this work. Then, we will discuss the details of the proposed dynamic programming approach for optimal intra-fragment mobility.

A. Assumptions

- The geographic location and shape of each fragment are known.
- Each fragment is discretized as described in Section II. Within each cell, a cluster-head (CH) is elected as the data aggregation node which will be responsible for relaying the cell's data to the mobile sink when it comes by. The cluster head is re-elected every predetermined amount of time to distribute the load (power consumption) within the cell.
- All sensor nodes are stationary and have the same transmission range R, which is also the transmission range of the mobile sink.
- We assume that the initial total energy of any cell at row n and column k denoted as $E_{(n,k)}$, is known, and is defined as the sum of the available energy levels at all sensor nodes which belong that cell. Therefore, different cells may have different initial energy levels. Also, an empty cell is assumed to have an infinite energy budget.
- P_k is the consumed transmission power by a CH at cell k due to routing the aggregated data of the cell to the MS or to adjacent cells to be relayed to MS.

B. Dynamic Programming (DP) Approach for Optimized Intra-Fragment Mobility

We propose a dynamic programming (DP) approach to find the most efficient traversal path within a given fragment.

As we have mentioned earlier, we consider two cases for the MS entrance/exit locations; (a) *the MS enters from any column in the uppermost (or lowermost) row, and exits through any column in the lowermost (uppermost) row*. (b) *the MS enters from any row in the leftmost (rightmost) column, and exits through any row in the rightmost (leftmost) column*. Throughout this section, we assume that the MS enters from the leftmost side and exits through the rightmost side. The proposed DP can be easily tweaked to work with the other three cases. In the proposed DP approach, we consider the general case of asymmetric clusters' energies and it minimizes the maximum energy consumption for each cluster, under the constraint of forward movement of the MS within the fragment.

***Definition** 3.1: Depletion rate (DR) of cell k*: is the ratio of the consumed power due to packet routing (from cell members to the mobile sink through the CH) to the initial energy level of the cell E_k.

To prolong the lifetime of each fragment in the network, we need to minimize the maximum depletion rate for fragment's cells. Thus, the potential trajectories the MS might follow through a fragment should be evaluated, and then the path with minimum-maximum depletion rate is selected as the optimal path. If sensing only changes sensors energy budgets marginally and by the same rate, then the evaluated MS movement trajectories will be valid for a long time.

In order to model the problem of finding the optimal movement trajectory for the MS within a fragment as a dynamic program, we discretize the fragment into cells (rows and columns[1]) as described earlier.

A formal definition of a dynamic program consists of *stages*, *states*, and *transition cost matrix*. In our formulation, stages correspond to columns of the discretized fragment, states correspond to all rows within a column, and the transition cost corresponds to the power depletion rate of making a transition from row i in stage $k-1$ to row j in stage k.

We assume the required transmission power for a packet is the same for all clusters which is one unit of power. Power depletion rate evaluation from stage $k-1$ to the adjacent stage k depends on the $cell(i)$ order in the previous stage $k-1$, the $cell(j)$ order in the next stage k, the fragment size represented by the number of rows and columns r and c respectively, and the clusters' energies in $k-1$ and k stages. We are given the following definitions:

- r and c: number of rows and columns in the discretized fragment, respectively.
- $DR_{n,m}^{(k-1,k,i,j)}$: $cell(n,m)$ power depletion rate cost of making a transition from row i in stage $k-1$ to row j in stage k.
- Each cell in $r \times c$ fragment's matrix has a transition depletion cost matrix DR of size $r \times r$, assuming there are no holes in the fragment, which is given by the following equations where n and m corresponds to cell's row and column respectively in the discretized fragment's matrix:

$$DR_{n,m}^{(k-1,k,i,j)} :$$

$$if(i<j) = \begin{cases} \frac{n}{E_{(n,m)}} & \text{if } 1 \leq n \leq i \\ \frac{1}{E_{(n,m)}} & \text{if } i < n \leq j \\ \frac{(r-n)+1}{E_{(n,m)}} & \text{if } j < n \leq r \end{cases}$$

$$elseif(i=j) = \begin{cases} \frac{n}{E_{(n,m)}} & \text{if } 1 \leq n < j \\ \frac{1}{E_{(n,m)}} & \text{if } n = j \\ \frac{(r-n)+1}{E_{(n,m)}} & \text{if } j < n \leq r \end{cases}$$

$$elseif(i>j) = \begin{cases} \frac{n}{E_{(n,m)}} & \text{if } 1 \leq n < j \\ \frac{1}{E_{(n,m)}} & \text{if } j \leq n < i \\ \frac{(r-n)+1}{E_{(n,m)}} & \text{if } i \leq n \leq r \end{cases}$$

- Figure 2 shows one instance for MSs transition, for the case when the MS enters the fragment from the left side and exits from the right side, from cell(3,k-1) to cell(6,k) where $i < j$ and DRs are shown in k stage cells. Then the maximum of $DR_{n,m}^{(k-1,k,3,6)}$ values is selected, where (1≤n≤7) and energy levels are asymmetric.
- $B_{i,j}^{k-1,k}$: total cost at row j in stage k due to making a transition from row i in stage k-1 to row j in stage k.
- $B_j^{k^*}$: minimum of maximum power depletion rate cost at row j in stage k.

Cost Evaluation in DP for Asymmetric Fragment's Cells Energy:

$$B_{i,j}^{k-1,k} = \max\left(\max_{i,1 \leq n \leq r} DR_{n,m}^{(k-1,k,i,j)}, B_i^{(k-1)*}\right) \tag{1}$$

$$B_j^{k^*} = \min_i B_{i,j}^{k-1,k} \tag{2}$$

The pseudo code for the DP is shown in algorithm 1. Clearly, the problem time complexity is polynomial and is $O(r^3 \times c)$.

Algorithm 1: Dynamic programming algorithm for intra-fragment mobility optimization.

```
input : Number of rows $r$.
        Number of columns $c$.
        The $r \times c$ fragment's cells' transition cost
  matrices $[DR_{n,m}^{k-1,k,i,j}]$, where each matrix is of size:
  $r \times r$.
output: The potential paths' coordinates in asymmetric
        fragment's cells' powers.
//for all columns;
for $k \leftarrow 1$ to $c$ do
    //for all rows in a column;
    for $j \leftarrow 1$ to $r$ do
        //for all rows in the previous column;
        for $i \leftarrow 1$ to $r$ do
            Compute $B_{i,j}^{k-1,k}$ using equation(1)
        Compute $B_j^{k^*}$ using equation(2)
```

[1] We use column and stage terms interchangeably.

IV. Inter-Fragment Mobility Optimization

The second phase of the problem is the inter-fragment mobility optimization. In this section, we propose an optimal formulation and a heuristic algorithm to find the optimal MS movement cycle between fragments such that MS trip length between fragments is minimized. This problem can be shown to be NP-hard by a reduction from the Traveling Salesman Problem to our problem (the proof is omitted due to lack of space).

A. Integer Linear Program Formulation

Let the number of EEPs be N, and let $\mathcal{F}(i)$ be the fragment of EEP i. Define a binary variable x_{ij} that is set to 1 if the directed edge (i, j) between the two EEPs i and j is selected to be part of the MS travel cycle, and set to 0 otherwise. Also, define f_{ij} as an integer variable that represents the flow on the directed edge (i, j). Finally, let the input parameter d_{ij} denote the Euclidean distance between EEP i and EEP j. Then, the problem of finding the optimal (shortest) MS travel cycle can be formulated as an Integer Linear Program (ILP) as shown below:

$$Minimize \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} d_{ij} \cdot x_{ij}$$

Subject to:

$$\sum_{\substack{j=1 \\ j \neq i}}^{N} x_{ij} = \sum_{\substack{j=1 \\ j \neq i}}^{N} x_{ji} = 1, \quad i = 1, \cdots, N \tag{3}$$

$$x_{ij} + x_{ji} = 1, \quad i, j \in \{1, \cdots, N\} \text{ and } \mathcal{F}(i) = \mathcal{F}(j) \tag{4}$$

$$\sum_{j=2}^{N} f_{1j} = N - 1 \tag{5}$$

$$\sum_{j=2}^{N} f_{j1} = 0 \tag{6}$$

$$f_{ij} \leq N \cdot x_{ij}, \quad i, j \in \{1, \cdots, N\}, \text{ and } i \neq j \tag{7}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{N} f_{ji} - \sum_{\substack{j=1 \\ j \neq i}}^{N} f_{ij} = 1, \quad i \in \{2, \cdots, N\} \tag{8}$$

Constraint (3) guarantees that exactly one incoming edge, and one outgoing edge are selected for each EEP. This will guarantee a cycle solution. However, multiple cycles might also be a feasible solution in this case. To avoid this, i.e., multiple cycles, we pick one of the EEPs as a flow-supplier (EEP 1 in the above formulation) with a supply of $N - 1$ units. As constraint (8) implies, each EEP (other than EEP 1) must consume one unit of flow and relay the remaining units through its outgoing edge. This will guarantee that if we start at EEP 1, we will be able to reach every other EEP. Given constraint (3), and the constraints (6), (7), and (8), this means that any feasible solution must be a Hamiltonian cycle. To make sure that the MS goes through the fragment when it reaches one of the fragments' EEPs, we added constraint (4). The constraint guarantees that exactly one of the two directed edges between the two EEPs of any fragment is selected as link in the MS travel cycle.

B. Heuristic Algorithm Formulation

In this subsection, we propose two algorithms to be jointly used to find a near optimal solution for the inter-fragment phase. The first is a Greedy-Solution Algorithm (GSA), algorithm 2, is used to find a near optimal solution for MS movement pattern cycle. Then, a Random Search Algorithm (RSA), algorithm 3, is used as an improvement heuristic for the solution resulting from the first algorithm. It is worth pointing out that the distance between the EEPs of a fragment depends on the MS trajectory within the fragment which is evaluated using Algorithm 1.

Define Ω as the resultant nodes' sequence for MS movement cycle pattern. GSA is a simple algorithm which starts by selecting one fragment's EEPs randomly as the first two nodes in Ω, lines (2-4). The distance between EEPs in the same fragment, is assumed to be zero. Then select the next node, j, in the sequence which is the closest node in EEPs to node i (head of the edge with the minimum distance from i), line 6. Therefore, select the next node, k, as the other node in j's fragment, line 7. Repeat these evaluation, line 5, $\forall$ EEPs.

RSA is proposed to improve the GSA results Ω. Before presenting the details of the RSA algorithm, we define the cost of a cycle (sequence) X, denoted $cost(X)$ as:

$$cost(X) = d_{X[|X|],X[1]} + \sum_{i=1}^{\frac{|X|}{2}-1} d_{X[2i],X[2i+1]} \tag{9}$$

where $d_{i,j}$ is the Euclidean distance between EEPs i and j.

RSA has two stages, in the First stage, RSA looks sequentially in the defined neighborhood of Ψ, line 1 (set $\Psi = \Omega$), the possible candidates, for a better MS movement pattern cycle. RSA starts switching the EEPs pairs for each two adjacent fragments (i) and (i+1) in Ψ sequentially. So the possible candidates number is $3 \times (M - 1)$ (M: number of fragments) where $s = 0$ line 1. For example, the first two fragments' EEPs in Ψ are (1,2) and (3,4), so the new possible candidate ζ_1, line 28, is {3,4,1,2} while other fragments' EEPs sequences remains the same. If ζ_1 cost is less than Ψ cost, then consider ζ_1 as the latest best solution $\Psi = \zeta_1$.

Then, ζ_1 is modified by just switching fragment(i) EEPs in the sequence, so call the new candidate sequence ζ_2, line 30. If ζ_2 is not a better solution, then back-track the last modification by setting $\zeta_2 = \zeta_1$. Now, modify ζ_2 by switching fragment (i+1) EEPs in the sequence, line 35. If ζ_2 is a better solution then $\Psi = \zeta_2$. In the next iteration, the next two fragments to be used as in the previous evaluations are fragments (i+1) and (i+2) with Ψ as the best sequence so far, line 9, and so on. These evaluations are repeated $(M - 1)$ times such that RSA looks one step ahead, $s = 0$.

Then RSA looks two steps ahead where $s = 2$, line 4, thus the possible candidates are $3 \times (M - 2)$, where fragments

Algorithm 2: Greedy search for MS movement cycle.

```
input : EEPs set S, EEPs' (x,y) coordinates, and $|S| = N$
output: MS movement pattern cycle $\Omega$ in FWSN.
Set $\Omega = \varnothing$, $c = N/2$;
Select a node, say u, randomly, out of S set;
Enforce the next node, i, to be the other node in u's fragment;
$\Omega = \Omega \cup \{u, i\}$, $c--$;
while $c \neq 0$ do
    $j = \text{argmin}_{\forall j \in S \setminus \Omega}\, d_{ij}$;
    Enforce the next node to be the other node in j's fragment, e.g.; k;
    $\Omega = \Omega \cup \{j, k\}$; $c--$; $i = k$;
```

(i) and (i+2) are used instead of fragments (i) and (i+1) in the previous evaluations explanation. In the second stage, first define new two version of Ψ which are Ψ_1 and Ψ_2. Ψ_1, line 10, is the reverse of nodes' sequence in Ψ. While Ψ_2, line 11, is defined by taking Ψ into two even halves and switch them. The goal of these two new version of Ψ is to search the neighborhood more globally rather than just locally.

As a next step in the second stage for each of Ψ, Ψ_1, and Ψ_2, respectively, lines (13-15), RSA does the following: RSA starts performing new sequence evaluations looking for a better sequence solution as in previous evaluation where RSA was looking one step a head. However after each fragments-pair evaluation, the next fragments pair or new solution candidate, is derived from latest derived candidate so far ζ_2, line 20. As mentioned earlier the reason of using ζ_2 to derive the next candidate, which may be not the best solution Ψ so far, is to search globally instead of locally. For each of three Ψs versions each has $3 \times (M-1)$ possible solution candidates. Pass parameters is used to tune the results accuracy by increasing its value. Thus the overall number of possible evaluated candidates in RSA is $pass \times (3 \times (M-1) + 3 \times (M-2) + 9 \times (M-1))$ that can be evaluated in a polynomial time $O(pass \times M)$.

V. Simulation Results and Discussion

To evaluate the intra-fragment mobility optimization solution, we compare our DP approach with the classical approach of having the MS take the shortest path between the two EEPs of the fragment that cuts the fragment into two halves.

Figure 3 shows a case-study FWSN topology which is used to evaluate the proposed DP approach and shortest path approach. Each fragment is discretized into clusters as illustrated in Figure 1 where the nodes and the MS communication radii are set to 80 units. Also 10 different energy levels are randomly distributed among clusters in the range [5-50] with a step of 5. Table I shows the maximum power DRs for each fragment where the possible EEPs for the MS are North(N), South(S), East(E), or West(W). Clearly, DP is better since its maximum power DR for any fragment is less than or equal to the classic shortest path algorithm, which causes the DP to reduce the maximum power DR in fragments.

Algorithm 3: Random search algorithm improvement heuristic for MS movement cycle Ω given by GSA.

```
input : EEPs' (x,y) coordinates, GSA solution, $\Omega$, for movement cycle pattern, and $|\Omega| = N$.
output: MS improved movement pattern cycle $\Psi$ in FWSN.
1  Set $\Psi = \Omega$; $\Gamma = \Omega$; $M = N/2$; $s = 0$;
2  for $r \leftarrow 1$ to 2 do
3      $P = M - r$;
4      If ($r = 2$) then $s = r$;
5      for $j \leftarrow 1$ to $pass$ do
6          for $k \leftarrow 1$ to $P$ do
7              $i = 2k - 1$;
8              Search(i,s);
9              $\Gamma = \Psi$;
10 Set $\Psi_1$ =Rev_$\Psi$;
11 Set $\Psi_2$ =SHlf_$\Psi$;
12 for $r \leftarrow 1$ to 3 do
13     If($r = 1$) then $\Gamma = \Psi$;
14     If($r = 2$) then $\Gamma = \Psi_1$;
15     If($r = 3$) then $\Gamma = \Psi_2$;
16     for $j \leftarrow 1$ to $pass$ do
17         for $k \leftarrow 1$ to $M - 1$ do
18             $i = 2k - 1$;
19             Search(i,0);
20             Set $\Gamma = \zeta_2$;
21
22 FUNCTION: cost(X,Y)
23 if $cost(X) < cost(Y)$ then
24     Y=X;
25
26 FUNCTION: Search(i,s)
27 $flg = 0$;
28 Set $\zeta_1 = \Gamma$, but modified by switching ($n_i$ with $n_{i+2+s}$) and ($n_{i+1}$ with $n_{i+3+s}$);
29 cost($\zeta_1$,$\Psi$);
30 Set $\zeta_2 = \zeta_1$, but modified by switching ($n_i$ with $n_{i+1}$)
31 if cost($\zeta_2$)<cost($\Psi$) then
32     $\Psi = \zeta_2$; flg=1;
33 if $flg = 0$ then
34     $\zeta_2 = \zeta_1$;
35 Now, modify $\zeta_2$ switching ($n_{i+2+s}$ with $n_{i+3+s}$);
36 cost($\zeta_2$,$\Psi$);
```

Now, to evaluate phase 2 (ILP, algorithm 2 and 3) that optimizes MS trip length between fragments (without considering the trip length inside fragments which is set to zero), we considered different fragment numbers in FWSNs. We generated 100 random topologies for each of the 13 FWSNs

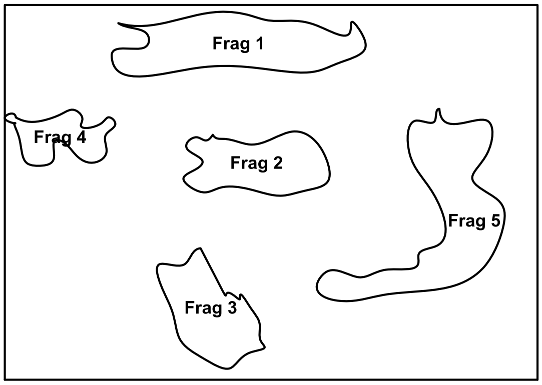


Fig. 3. FWSN topology for the case study.

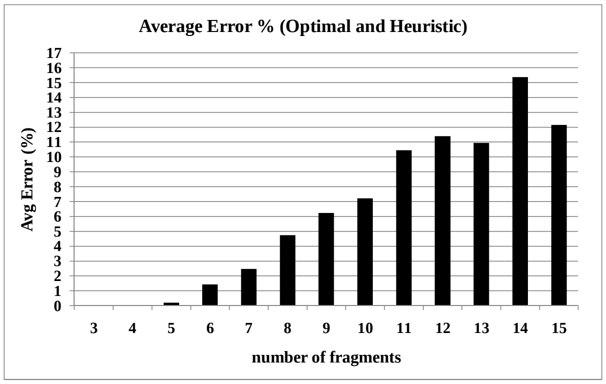


Fig. 4. Average error percentages for different FWSNs with fragments numbers (3-15) such that each has 100 random topologies.

cases. The considered FWSNs cases have (3-15) fragments.

Then using the optimal (ILP) and heuristic algorithms (greedy and RSA) the MS cycle pattern and the distances between fragments are evaluated for each generated topology. The average errors percentages, equation (10), for distances between fragments are evaluated as shown in figure 4, where the RSA pass parameter is set to 30. The results show that the average error is about 15% with respect to the optimal solution.

$$Error(\%) = [(OPT - Heuristic)/OPT] \times 100\% \quad (10)$$

VI. Conclusions

In this paper, we studied the problem of optimizing the mobility of a sink node that is used to repair connectivity in a FWSN. We studied two phases of mobility optimization namely: intra-fragment and inter-fragment. In intra-fragment optimization, we aim at minimizing power consumption within a fragment. In inter-fragment optimization, on the other hand, we aim at minimizing the total time needed for the MS to service all fragments in the network. A dynamic programming solution was proposed for the intra-fragment phase, and an optimal ILP formulation along with a near optimal heuristic were proposed for the inter-fragment phase. The results show that a significant reduction in the consumed power at each fragment is achieved by using the proposed intra-fragment optimization approach compared to that using the shortest-path within the fragment. On the other hand, the results also show the accuracy of the proposed heuristic solution for the inter-fragment phase, which has been about 15% error with respect to the optimal solution.

TABLE I

The maximum power depletion rates for DP and shortest path scenarios.

fragment	DP	$Shortestpath$	$EEPs$
1	0.20	0.40	W to E
2	0.20	0.40	W to E
3	0.20	0.40	N to S
4	0.20	0.20	W to E
5	0.20	0.80	N to S

Acknowledgments

This research was supported in part by the National Science Foundation under grants CNS-0626822, ECS-0601570, and ECCS-0926029.

References

[1] E. Ekici, Yaoyao Gu, and D. Bozdag. Mobility-based communication in wireless sensor networks. *Communications Magazine, IEEE*, 44(7):56–62, 2006.

[2] Wei Wang, Vikram Srinivasan, and Kee-Chaing Chua. Extending the lifetime of wireless sensor networks through mobile relays. *IEEE/ACM Transactions on Networking*, 16(5):1108–1120, 2008.

[3] Wei Wang, Vikram Srinivasan, and Kee-Chaing Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 270–283, 2005.

[4] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 187–198, 2004.

[5] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: modeling a three-tier architecture for sparse sensor networks. In *IEEE Sensor Network Protocols and Applications (SNPA) Workshop*, pages 30–41, 2003.

[6] H. Almasaeid and A. E. Kamal. Data delivery in fragmented wireless sensor networks using mobile agents. In *The tenth ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2007.

[7] H. Almasaeid and A. E. Kamal. Modeling mobility-assisted data collection in fragmented wireless sensor networks. In *IEEE GLOBAL Communications Conference*, 2008.

[8] J. Luo and J. Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. *INFOCOM*, 2005.

[9] Ioannis Chatzigiannakis, Athanasios Kinalis, and Sotiris Nikoletseas. Sink mobility protocols for data collection in wireless sensor networks. In *MobiWac '06: Proceedings of the 4th ACM international workshop on Mobility management and wireless access*, pages 52–59, 2006.

[10] S. Funke. Topological hole detection in wireless sensor networks. In *DIALMPOMC'05*, September 2005.

[11] Nisheeth Shrivastava, Subhash Suri, and Csaba D. Toth. Detecting cuts in sensor networks. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 28, 2005.

[12] Qing Ye and Liang Cheng. A study of reconnecting the partitioned wireless sensor networks. In *Networking - ICN 2005*, volume 3420/2005 of *Lecture Notes in Computer Science*, pages 561–568. Springer Berlin / Heidelberg, 2005.

[13] Gianluca Dini, Marco Pelagatti, and Ida Maria Savino. An algorithm for reconnecting wireless sensor network partitions. In *Wireless Sensor Networks*, volume 4913/2008 of *Lecture Notes in Computer Science*, pages 253–267. Springer Berlin / Heidelberg, 2008.

[14] J. Al-Karaki and A. E. Kamal. Efficient virtual-backbone routing in mobile ad hoc networks. *Elsevier Computer Networks*, 52, February 2008.