On the Minimum k-Connectivity Repair in Wireless Sensor Networks

Hisham M. Almasaeid and Ahmed E. Kamal
Dept. of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011
Email:{hisham,kamal}@iastate.edu

***Abstract*—Repairing connectivity and achieving a certain level of fault tolerance are two important research challenges in wireless sensor networks that have, in many papers in the literature, been jointly studied. Most of the proposals that aim at restoring network connectivity deal with the network as a general graph of N nodes with the edge cost being the number of nodes needed to establish connectivity between the two ends of the edge. This assumption ignores the topological properties of the network, especially the overlap between sensors' communication ranges, and the node-failure pattern that caused the disconnection. In this paper, we try to exploit these properties to minimize the number of additional nodes needed to repair the connectivity.**

I. Introduction

The randomness of the deployment process and the vulnerability of sensor nodes may cause network disconnection. Therefore, researchers have tackled the network disconnection problem at both the deployment phase, i.e., *deployment with guaranteed connectivity*, and during network operation, i.e., *connectivity repair*. Although 1-connectivity, which means that one path exists between every pair of node, is the basic requirement for all networks, some networks are required to be k-connected, that is, to have k node disjoint paths between every pair of nodes in the network. k-connectivity may be used for different purposes including fault-tolerance, i.e., being able to maintain operation in a timely manner with the occurrence of failures, and high throughput through multi-path routing. The problem of establishing k-connectivity in a network with the minimum number of additional nodes, referred to in this paper as relay nodes, is called the *minimum k-connectivity repair* problem, and it is known to be NP-complete.

k-connectivity can be either *full* or *partial* [1]. *Full k-connectivity* implies that k node-disjoint paths exist between every pair of nodes including the added relay nodes. However, the *partial k-connectivity* implies k-connectivity between original sensor nodes only. Among the first studies of k-connectivity in wireless sensor networks is the work in [2]. The goal was to provide fault-tolerance and capacity guarantees by establishing k-connectivity in the network. Approximation and greedy algorithms based on the minimum k-CONNECTED-SUBGRAPH were proposed to achieve *full* k-connectivity.

In [3], S. Misra and et. al. have studied a constrained version in which relay nodes can only be placed at a set of candidate locations in a heterogeneous and hybrid network that consists of three types of nodes: *sensors*, *relays*, and *base stations*. The objective was to guarantee k-connectivity to a base station with the least number of relays. Partial and full k-connectivity in heterogeneous wireless sensor networks, where sensor nodes possess different transmission radii, were studied in [4]. For a more detailed and comprehensive report on the current state of research on the minimum relay node placement problem we refer the reader to the recent survey in [5].

Assuming normal failure conditions, sensor nodes fail individually when they run out of power or get damaged. This kind of failure inherently creates intersection regions of the communication ranges of a number of nodes that is equal to the degree of the failed node. In other words, if a sensor node has a degree of n, then it lies in the communication ranges of all the n nodes. Thus, if it fails leaving its neighbors disconnected from each other, it will leave a vacant spot such that if we place a new node at, it with the same communication radius as the original one, the n nodes will recover connectivity with each other. Therefore, if a network is originally connected and failures occur randomly at the individual level, then we expect a significant amount of intersections between communication ranges of sensor nodes to exist in a disconnected network.

This observation is the basic motivation behind this paper. What this observation basically means is that the total number of relay nodes necessary to connect a disconnected network can be reduced by considering the intersection between the communication ranges of sensor nodes. To illustrate the idea, consider the example shown in Figure 1. The traditional way of using the minimum spanning tree approach to connect the network will result in a cost of 4 nodes (the left-hand side). However, if the intersection between communication ranges is considered, we can find two spots by placing a node at each of which, three sensor nodes are connected together which reduces the total cost to 2 nodes only (the right-hand side). Based on this observation, we aim at proposing a centralized solution that utilizes the amount of intersection between communication ranges of existing sensor nodes to reduce the total number of additional nodes needed to repair k-connectivity in a k-disconnected network. The objective of this paper is not merely to provide more efficient algorithms, but rather to show that the intersection between communication ranges can improve the performance of existing algorithms.

The rest of this paper is organized as follows. In Section II we give some preliminary definitions and assumptions. We address the problem of 1-connectivity in Section III. Then, we

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

978-1-4244-3435-0/09/$25.00 ©2009 IEEE

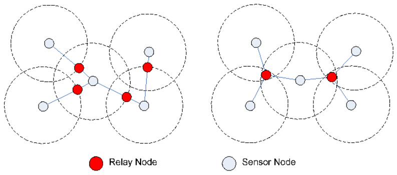
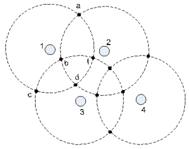


Fig. 1. Graph based (left) versus topology aware (right) relay node placement.



IR_i	$Contour$	IS_i
IR_1	(a, b, d, f)	$\{1, 2\}$
IR_2	(b, c, d, f)	$\{1, 3\}$
IR_4	(b, d, f)	$\{1, 2, 3\}$

Fig. 2. A component of four nodes (left) and some of its intersection regions (IRs) and their corresponding intersection sets (ISs) (right).

extend the discussion to the case of k-connectivity repair in Section IV. Experimental results are provided in Section V. Then, we conclude in Section VI.

II. Preliminaries

Let the network be given as graph $G(V, E)$. V ($|V|$=N) is the set of vertices (i.e., sensor nodes), N is the cardinality of this set, and E is the set of edges (communication links).

A. Definitions

Definition 2.1: The *communication range* of a node is defined as the circle centered at the location of that node with a radius equal to the maximum transmission range of the node.

Definition 2.2: A set of sensor nodes $C \subseteq V$ is said to be a *component* in a wireless sensor network *iff* for any two arbitrary subsets $C_1, C_2 \subset C$ and $C_1 \cup C_2 = C$, there exists at least one pair of nodes (u, v) where $u \in C_1$ and $v \in C_2$ such that their communication ranges intersect.

Depending on the topology, number of failed nodes, and the failure pattern, a disconnected sensor network may contain several components, Figure 1 two components.

Definition 2.3: A region that results from the intersection of the communication ranges of an arbitrary set of nodes (we call this set of nodes the *intersection set (IS)*) is called an *intersection region (IR)*. The intersection points that bound this IR are called the *contour points* of the IR.

For example, in Figure 2, the intersection of the three communication ranges of nodes $\{1, 2, 3\}$ creates four IRs as shown in the figure that are associated with four intersection sets $IS_1 = \{1, 2\}, IS_2 = \{1, 3\}, IS_3 = \{2, 3\}$, and $IS_4 = \{1, 2, 3\}$. As IRs may overlap, for example $IR_1 \cap IR_2 = IR_4$ in Figure 2, we define the set of *independent intersection regions (IIR)*.

Definition 2.4: A set of *intersection regions* $\mathcal{R}$ is said to be an *Independent Intersection Regions* set, IIR for short, *iff* for any two regions $IR_i, IR_j \in \mathcal{R}$, $IR_i \cap IR_j = \emptyset$.

It can be easily shown that establishing connectivity within a *single component* with the minimum number of additional nodes can be achieved by deploying those nodes only inside IR's. In other words, any solution that includes the deployment of an additional node(s) outside the IIR's of a component can be replaced with another solution in which all additional nodes are deployed within the IIR's with less or equal cost. Therefore, we propose two solutions:

- *At contour points of an IR*: for example, if we want to place a relay node in IR_4 shown in Figure 2, we have three candidate locations; b, d, or f.
- *At the center of an IR*: having relay nodes placed at the intersection points will cause the nodes that belong to the intended *IS* to use unequal transmission power levels to maintain the same level of channel reliability. For example, the placement of a relay node at point b to connect the *IS*= $\{1, 2, 3\}$ will make the channel between nodes 1 and the relay node at b much more reliable than those between nodes $2, 3$ and the relay forcing nodes 2 and 3 to use higher transmission power than node 1. Therefore, we propose another strategy that is *to place relay nodes at the center of every IR in the selected* IIR. This will make the connections between a relay node and the intended *IS* equally reliable and balance the transmission power among all members of the IS.

Contour points can be beneficial because they lie at the boundary of IR's. The cost of connecting two components together might be reduced by considering a pair of boundary relay nodes, each belongs to one of the components, that are close enough to each other (if such pair exists). Therefore, we use the *Contour points* approach for all the algorithms proposed in this paper. Then, nodes can be repositioned to the centers of the regions to which they belong.

B. Assumptions

Before we start analyzing the problem and proposing solutions, we list the assumptions made in this work.

- All nodes, original sensor nodes and the added relay nodes, have the same communication radius R.
- The locations of sensor nodes are known to a central station where computations take place.
- Full connectivity is desired, i.e., if the network is required to be k-connected, this must hold for original sensor nodes as well as added relay nodes.

III. 1-Connectivity repair

We divide the *k-connectivity repair* problem into two phases: *intra-component k-connectivity repair*, in which k-connectivity is restored within each component, and *inter-component k-connectivity repair*, in which k-connectivity is restored between components. In this section, we study the two phases for 1-connectivity repair and defer the k-connectivity repair case to Section IV. The network is said to be 1-connected if there is a path between any pair of nodes in the network, and we use the words 1-*connected* and *connected* equivalently throughout the paper. Let,

- $\mathcal{X}$ be the set of original sensor nodes.
- $\mathcal{Y}$ be a set of nodes that represents relay nodes placed at the candidate locations determined according to any of the two methods described in Section II.
- $d(u, v)$ be the Euclidean distance between u and v.

Algorithm 1: G-MCC: A greedy-algorithm for MCC.

```
input  : An auxiliary graph $G_a = (\hat{\mathcal{X}} \cup \mathcal{Y}, E)$
output: The subset of selected relays $\hat{\mathcal{Y}} \subset \mathcal{Y}$.
$\hat{\mathcal{Y}} \longleftarrow \emptyset$;
while $|\hat{\mathcal{X}}| > 1$ do
    Pick a vertex from $\mathcal{Y}$, say $y$, that connects the maximum number of vertices from $\hat{\mathcal{X}}$ and let this set of vertices be $\mathcal{S}$;
    $\hat{\mathcal{Y}} = \hat{\mathcal{Y}} \cup y$;
    Add a new vertex, say $x$, to $\hat{\mathcal{X}}$ and extend an edge between $x$ and all the neighbors of the set $\{y \cup \mathcal{S}\}$;
    $\mathcal{Y} = \mathcal{Y} \backslash \{y\}$;
    $\hat{\mathcal{X}} = \hat{\mathcal{X}} \backslash \mathcal{S}$;
return $\hat{\mathcal{Y}}$;
```

- E be a set of edges that includes:
  - an edge between $x_i, x_j \in \mathcal{X}$ denoted as (x_i, x_j) if $d(x_i, x_j) \leq R$; denote this set of edges as $E_{s\leftrightarrow s}$,
  - an edge between $y_i, y_j \in \mathcal{Y}$ denoted as (y_i, y_j) if $d(y_i, y_j) \leq R$; denote this set of edges as $E_{r\leftrightarrow r}$,
  - an edge between $x_i \in \mathcal{X}$ and $y_j \in \mathcal{Y}$ denoted as (x_i, y_j) if $d(x_i, y_j) \leq R$; denote this set of edges as $E_{s\leftrightarrow r}$.

Definition 3.1: Minimum Component Connectivity (MCC) problem: given the graph $G(\mathcal{X} \cup \mathcal{Y}, E)$, where $E = E_{s\leftrightarrow s} \cup E_{s\leftrightarrow r} \cup E_{r\leftrightarrow r}$, find a connected subgraph $G'(V, E')$ such that $\mathcal{X} \subset V$ $(V \subset \mathcal{X} \cup \mathcal{Y})$ and $E' \subset E$ with the minimum number of vertices.

The MCC problem can easily be shown to be NP-Complete by a reduction from the set cover problem [6]. Therefore, we propose a greedy algorithm to solve it. But before introducing the details of the algorithm, we describe the construction of an auxiliary graph that would be an input to the algorithm.

A. Auxiliary graph construction

1) For the graph $G(V = \mathcal{X}, E = E_{s\leftrightarrow s})$, define an auxiliary graph $G_a(V_a = \mathcal{Y}, E_a = E_{r\leftrightarrow r})$.
2) For every vertex $x \in \mathcal{X}$, such that there does not exist an edge $(u, v) \in E_{s\leftrightarrow s}$ for which x is one of the endpoints,
   - Add x to V_a.
   - Add all the edges $(x, z) \in E_{s\leftrightarrow r}$ where $z \in \mathcal{Y}$.
3) For every connected subgraph $\mathcal{G}$ of size ≥ 2 in G,
   - Add a new vertex, say v, to V_a.
   - If there exists an edge $(x, y) \in E_{s\leftrightarrow r}$ such that $x \in \mathcal{G}$ and $y \in \mathcal{Y}$, then add the edge (v, y) to E_a.
4) Remove every vertex in $\mathcal{Y}$ that is connected to one vertex only in $\hat{\mathcal{X}}$. Such vertices represent intersection points of communication ranges of two sensor nodes that belong the same connected group.

B. Greedy algorithm for the MCC problem ($G{-}MCC$)

Algorithm 1 outlines a greedy approach to achieve *intra-component* 1*-connectivity* using the fewest possible number of relay nodes. The basic idea of the algorithm is to pick at each iteration the relay node that connects the maximum number of connected groups of sensor nodes until the whole component is connected. Therefore, at each iteration the algorithm picks the vertex from $\mathcal{Y}$ that is connected to the maximum number of vertices in $\hat{\mathcal{X}}$. Note that each vertex in $\hat{\mathcal{X}}$ represents a connected group of sensor nodes or just one sensor node. As the algorithm progresses, a vertex in $\hat{\mathcal{X}}$ will represent sensor nodes and relay nodes selected at previous iterations. Whenever a vertex from $\mathcal{Y}$ is selected, this vertex and all the vertices connected to it from $\hat{\mathcal{X}}$ are replaced by one new node. Then, every vertex that was connected to any of the replaced vertices will have an edge to the new vertex. The algorithm progresses until there is exactly one node in the set $\hat{\mathcal{X}}$, which means that the *component* is now connected.

Algorithm 2: $G{-}MST(1)$: 1-Connectivity repair.

```
input  : A 1-disconnected sensor network given as a set of sensor nodes $\mathcal{X}$ and their locations.
         The communication radius $R$.
output: A minimum spanning tree $T_{min}$ that connects all original nodes and the new added relay nodes.
Define $\mathcal{V} = \mathcal{X}$;
Detect all components in the network and let the set of these components be $\mathcal{S}_c$;
foreach component $C(X, Y) \in \mathcal{S}_c$ do
    Construct an auxiliary graph $G_a(X, Y)$;
    Invoke Algorithm 1 with $G_a(X, Y)$ being the input to make $C(X, Y)$ 1-connected;
    Add the output of the algorithm, i.e., the set $\hat{\mathcal{Y}}$, to $\mathcal{V}$ $(\mathcal{V} = \mathcal{V} \cup \hat{\mathcal{Y}})$;
Construct a complete graph $G(\mathcal{V}, E = \mathcal{V} \times \mathcal{V})$ and assign each edge $(u, v)$ a cost of: $\lceil \frac{d(u,v)}{R} \rceil - 1$;
Invoke any Minimum Spanning Tree algorithm over $G$ and let the resulting tree be $T_{min}$;
At every edge $(u, v) \in T_{min}$ of weight $\omega > 0$, place $\omega$ relay nodes along the line between $u$ and $v$ evenly spaced by a distance of $\frac{d(u,v)}{\omega+1}$ and add them to $T_{min}$;
return $T_{min}$
```

C. Inter-Component k-connectivity repair

As the network may consist of several components, there is a need for an *inter-component k-connectivity repair* phase, after which the whole network becomes connected. For this phase, we simply apply any minimum spanning tree algorithm with edges assigned weights equal to the number of additional nodes needed to extend the edge. Algorithm 2 outlines our proposed approach to achieve 1-connectivity in a disconnected network. We first use Algorithm 1 to establish intra-component connectivity for all the detected components in the network. Then, we apply a minimum spanning tree algorithm over a complete graph of the original sensor nodes and the relay nodes selected in the *intra-component repair* phase.

IV. K-CONNECTIVITY

In some applications, the network is required to have some level of fault tolerance. In such cases, redundant nodes are added to the network in a way that makes it *k-connected*, i.e.,

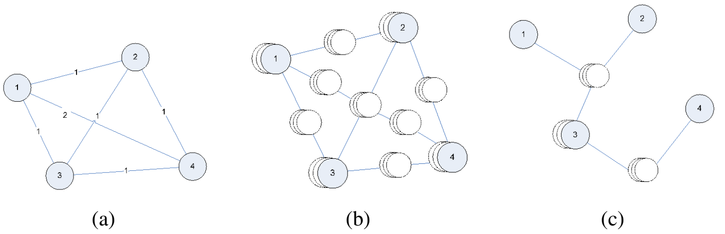


Fig. 3. (a) A complete graph with edge weights (b) A 3-connectivity repaired network using the *minimum k-connected subgraph* approach. (c) A 3-connectivity repaired network using the intersection points. Dashed circles represent additional relay nodes.

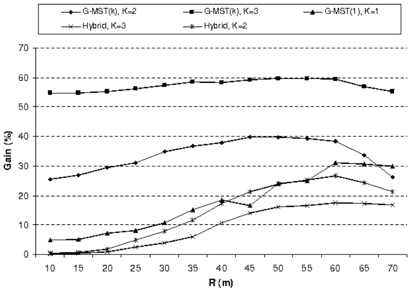


Fig. 4. The gain achieved by considering the intersection regions.

k node disjoint paths exist between every pair of nodes. Our claim is that if the intersection between communication ranges of different sensor nodes is considered, a considerable gain, defined as the reduction in the total number of added nodes, will be achieved. In [2], the authors proposed an algorithm to establish k-connectivity based on finding the *minimum k-connected subgraph*. A complete graph with edges assigned weights that are equal to the number of additional nodes needed to establish connectivity along the edge is fed to a *minimum k-connected subgraph* algorithm. Then, for every edge (u, v) in the resulting subgraph that has a weight $\omega > 0$, the following *k-augmentation process* is performed:

- $\omega \cdot k$ relay nodes are added along the edge in groups of size k each (to guarantee *full*-connectivity) evenly spaced with distance $\frac{d_{u,v}}{\omega+1}$ and,
- $k-1$ relays are added at the location of each of u and v.

As we will be using this algorithm as a baseline approach for comparison, we further illustrate its operation with an example. Consider a network that consists of the nodes $\{1, 2, 3, 4\}$ in Figure 2. Figure 3(a) shows the complete graph for the aforementioned set of nodes with an edge assigned a weight that is equal to the number of additional nodes needed to establish connectivity between its end-nodes. Figure 3(b) is the result of the minimum K-CONNECTED-SUBGRAPH algorithm used in [2]. The total number of nodes required using this algorithm is 29. However, we could achieve the same level of connectivity by adding only 6 nodes, i.e., a gain of $\approx 79\%$, if the intersection between communication ranges is considered as Figure 3(c) shows.

Therefore, we propose another approach for k-connectivity repair that is based on Algorithm 2 used for establishing *1-connectivity*. The new approach is outlined in Algorithm 3. We first use Algorithm 2 to make the network 1-connected. Then, we add additional $k-1$ nodes at the location of every non-leaf node in the minimum spanning tree that results from Algorithm 2. Figure 3(c) shows the output of our algorithm to repair the network in Figure 2 to be 3-connected.

For the sake of comparison, we propose another algorithm to show how the traditional *minimum k-connected subgraph* can be enhanced by considering the intersection of communication ranges. The approach is outlined in Algorithm 4. We first find the minimum k-connected subgraph to make the original network graph $G_0(V_0, E_0)$ k-connected (the first phase of Algorithm 1 in [2] without k-augmentation, and we used Algorithm 2 in [2] to find the minimum k-connected subgraph), we call the new graph $G_k(V_k, E_k)$. At the same time, we apply Algorithm 1 on the graph $G_0(V_0, E_0)$ to select relay nodes at intersection points to establish 1-connectivity within each component, let the resulting graph be $G_1(V_1 \cup \mathcal{Y}, E_1)$ where $\mathcal{Y}$ is the set of added relay nodes. Then, we remove every edge in the k-connected subgraph $G_k(V_k, E_k)$ that has a weight $\omega > 0$ and its two endpoints are connected to a relay node $y \in \mathcal{Y}$ in the graph $G_1(V_1, E_1)$. As a replacement, we add the two undirected edges (u, y) and (v, y) to G_k and place $k-1$ relay nodes at the position of y (a total of k including y). Finally, we apply the *k-augmentation process* described earlier.

Algorithm 3: $G{-}MST(k)$: K-connectivity repair.

```
input : A disconnected sensor network given as a set of
        sensor nodes $\mathcal{X}$ and their locations.
        An integer $k$.
        The communication radius $R$.
output: A k-connected graph.
1 Invoke Algorithm 2 with $\mathcal{X}$ and their locations as input
  to produce a tree $T_{min}$;
2 foreach non-leaf node $y$ in $T_{min}$ do
3     Add $k-1$ nodes to $T_{min}$ at the location of $y$;
4 return $T_{min}$;
```

V. Results

Intuitively, the performance of the proposed algorithms in this paper depends on the amount of intersection between communication ranges. Therefore, no gain is expected if the network contains no components. We use a network of 50 nodes randomly distributed in an area of $720m \times 416m$ with a variable communication radius R in the range $[10m, 70m]$ with an increment of $5m$. We experimented with 20 topologies that are randomly generated using these settings and results are taken as the averages over all the topologies. Figure 4, shows the gain of using $G{-}MST(1)$, $G{-}MST(k)$, and the *Hybrid k-connectivity repair* algorithms over the minimum *k-CONNECTED-SUBGRAPH* based algorithm proposed in [2] (Algorithm 1). Note that for k=1, the minimum k-connected subgraph is the minimum spanning tree. As for 1-connectivity, the $G{-}MST(1)$ algorithm attains a maximum gain of about 31% as Figure 4 shows. On the other hand, the gain of using

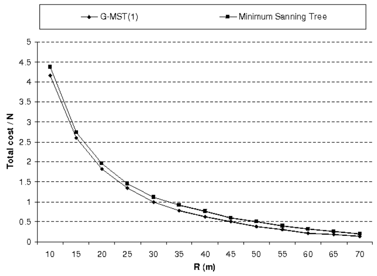


Fig. 5. The average cost per node for 1-connectivity repair.

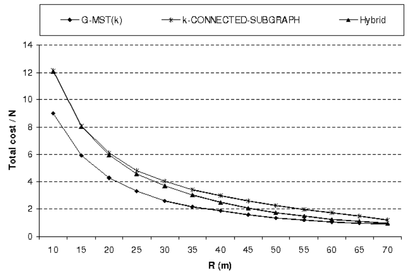


Fig. 6. The average cost per node for 2-connectivity repair.

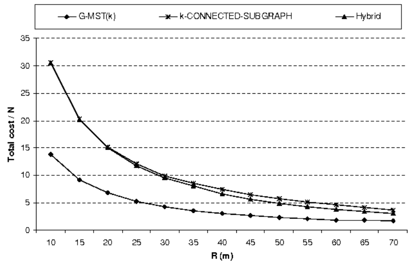


Fig. 7. The average cost per node for 3-connectivity repair.

Algorithm 4: Hybrid k-connectivity.

```
input : A disconnected sensor network given as a graph
        $G_0(V_0, \emptyset)$. An integer $k$.
output: A k-connected graph $G_k(V_k, E_k)$.
Use Algorithm 1 to establish 1-connectivity in all
components. Let the result be $G_1(V_1{=}V_0 \cup \mathcal{Y}, E_1)$;
Find the minimum k-connected subgraph of
$G_0(V_0, V_0 \times V_0)$, with the weight of $(u, v)$ be
$\left(\frac{d_{u,v}}{R} - 1\right)$. Let the result be $G_k(V_k{=}V_0, E_k)$, where
$E_k \subseteq V_0{\times}V_0$;
$E'_k \leftarrow E_k$;
foreach edge $(u, v) \in E_k$ with a weight
$\omega = \frac{d_{u,v}}{R} - 1 > 0$ do
    if $\exists\ (u, y), (v, y) \in E_1$ then
        $V_k = V_k \cup y$;
        $E'_k = E'_k \backslash (u, v) \cup \{(u, y), (v, y)\}$;
        Place $k{-}1$ relays at $y$ and add them to $V_k$;
$G_k \leftarrow (V_k, E'_k)$;
Invoke the k-augmentation process over all edges of
non-zero weight in the graph $G_k$;
return $G_k$;
```

$G{-}MST(k)$ is significant reaching a maximum of around 40% for k=2 and 61% for k=3 both at R=50. The gain starts small and keeps increasing with R until it reaches the point of best intersection settings and then starts to decease as the number of connected sensor nodes increases. The cost per node defined as the number of added relay nodes divided by the number of original sensor nodes N is shown in Figure 5 for 1-connectivity repair. The two curves for $G{-}MST(1)$ and *Minimum spanning tree* are very close because the number of added relay nodes for 1-connectivity is relatively small compared to N for large values of R, and the gain of our approach is too low for small values of R (using few intersections). It is clear in the figure how the gain increases with R as more intersection regions are created. If no intersections exist, $G{-}MST(1)$ performs exactly the same as the minimum spanning tree. Figure 6 shows the per node cost of 2-connectivity repair. Note how the difference between the two curves of $G{-}MST(k)$ and *k-CONNECTED-SUBGRAPH* has increased compared to that for k=1, and how it has increased further for k=3 (to a maximum of around) as Figure 7 shows. This leads to the conclusion that the gain increases with k. There are two sources of gain for $G{-}MST(k)$. The first is attained by considering intersections between communication ranges. And the second is due to the fact that the result of $G{-}MST(k)$ is a tree with k duplicates at non-leaf nodes in the tree. As the basic tree, without the duplicate nodes, is the same for the same network regardless of k, the number of leaf nodes in this tree is another source of gain as no duplicate nodes are added there. Figure 3(c) shows these two sources. To prove that considering intersection regions can improve existing algorithms, we proposed Algorithm 4. As Figure 4 shows, the *hybrid* algorithm attains a maximum gain of $\approx 27\%$ for k=2 and $\approx 18\%$ for k=3 both at R=60. Here the case is reversed, i.e., the gain is higher for k=2 than for k=3. This is due to the fact that the set of relay nodes selected by invoking Algorithm 1 is the same for both k=2 and k=3.

VI. Conclusions

The problem of exploiting the intersection between communication ranges to reduce the number of additional nodes needed to repair k-connectivity in a wireless sensor network is studied in this paper. We have proposed k-connectivity repair algorithms that utilize the intersection for both cases of k=1 and $k > 1$. We have also proposed a hybrid algorithm that takes the output of one of the proposed algorithms in literature and enhances it by considering intersection regions. This shows that existing algorithms can be improved by taking into account the intersection between communication ranges. Our algorithms attain considerable gain over traditional minimum k-CONNECTED-SUBGRAPH based algorithms.

References

[1] J. Pu, Z. Xiong, and X. Lu, "Fault-tolerant deployment with k-connectivity and partial k-connectivity in sensor networks," *Wireless Communications and Mobile Computing*, 20 May 2008.

[2] J. Bredin, E. D. Demaine *et al.*, "Deploying sensor networks with guaranteed capacity and fault tolerance," *MobiHoc*, May 2005.

[3] S. Misra, S. D. Hong *et al.*, "Constrained relay node placement in wireless sensor networks to meet connectivity and survivability requirements," *INFOCOM 2008*, April 2008.

[4] X. Han, X. Cao *et al.*, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," *INFOCOM 2007*, May 2007.

[5] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.