Estimating the Free Region of a Sensor Node

Laxmi Gewali, Navin Rongratana,
Jan B. Pedersen
School of Computer Science,
University of Nevada
4505 Maryland Parkway
Las Vegas, NV, 89154, USA

Abstract

We consider the problem of relocating a sensor node in its neighborhood so that the connectivity of the network is not altered. In this context we introduce the notion of *in-free* and *out-free regions* to capture the set of points where the node can be relocated by conserving connectivity. We present a characterization of maximal free-regions that can be used for identifying the position where the node can be moved to increase the reliability of the network connectivity. In addition, we prove that the free-region computation problem has a lower bound $\Omega(n \log n)$ in the comparison tree model of computation, and also present two approximation algorithms for computing the free region of a sensor node in time $O(k)$ and $O(k \log k)$.

1 Introduction

Problems dealing with the deployment and relocation of sensor nodes have attracted the interest of many investigators [4, 3, 2, 5, 8]. In the deployment problem, we are given a fixed region R such as a terrain surface and we need to deploy nodes on it, such that the region can be covered by the sensing group of nodes. At the same time, the network must be connected. In the relocation problem, we are given a pre-deployment of nodes in a fixed region R and we need to relocate some nodes by small displacement so that the region R can be covered.

In this paper, we introduce the notion of a *free-region* for nodes in a sensor network that can be used relocation. Intuitively, the free-region $FR(p)$ of a node p is the maximal connected region contained in the broadcast

JCMCC **74** (2010), pp. 53-64

disc of p so that the local connectivity properties of p are preserved even if p is moved to any point inside R. In Section 2, we characterize the free-region problem of a sensor node. We show that the problem of computing the free-region of a sensor node has a lower bound $\Omega(n \log n)$. In Section 3, we propose two approximation algorithms for estimating a sub-set of the free-region. The first approximation algorithm we propose is called the *Empty Circle Approximation* and runs in $O(k)$ time. The second approximation algorithm, called *Convex Approximation* computes the free-region in $O(k \log k)$ time, where k is the number of in-bound and out-bound nodes of the candidate node. Both algorithms are simple and efficient, and can be used for practical implementation in environmental monitoring and surveillance applications. Finally, we discuss possible extensions of the proposed algorithms.

2 Preliminaries

Consider n sensor nodes $v_1, v_2, ..., v_n$ deployed on a terrain surface, which is taken as a two dimensional plane. The location of node v_i is represented by point q_i with coordinates x_i and y_i. The transmission range r of all sensor nodes is assumed to be identical and the implied transmission region is taken as the *transmission disk* $TD(i)$ of radius r centered at q_i. The circle of the transmission is denoted as $TC(i)$. We can imagine a network obtained by connecting all pair of nodes within each others transmission range. Such a network is often called *Unit Disk Graph UDG* [1] and we denote it by *G(V,E)*, where V and E are the set of nodes and the set of edges, respectively. Fig. 1 shows an example of the unit disc graph

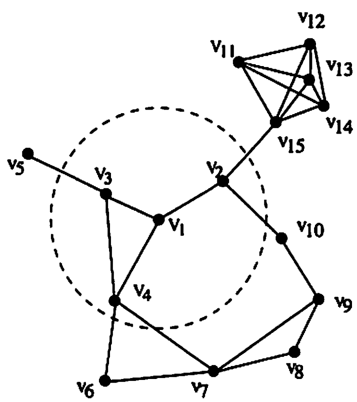


Figure 1: Illustrating a Unit Disk Graph (UDG).

induced by 15 nodes, where the disk with dashed boundary indicates the transmission region corresponding to node v_1.

We first start with a few more definitions. A pair of nodes v_i and v_j are called *neighbors* or *adjacent* if they are within each others' transmission range. Similarly, a pair of non-adjacent nodes v_i and v_j are called *adjoining* if their transmission disks $TD(i)$ and $TD(j)$ intersect.

Now consider what happens to the connectivity of the network when a node, say v_1, is moved very slightly. It is very likely that the connectivity will remain the same. If we continue to slowly move the node in some direction, two kinds of events can occur. A node that was within the transmission region of v_1 at the beginning may fall outside the range. For example, if node v_1 is moved along the y-direction, node v_4 will fall outside the transmission region of v_1. We call such event as an *excluding event.* If the node continues to move further along the y-direction, node v_5, which was outside the transmission range of v_1 at the start, will eventually appear within the range. We call this type of event an *including event.* This observation leads us to model free-region for a sensor node as follows.

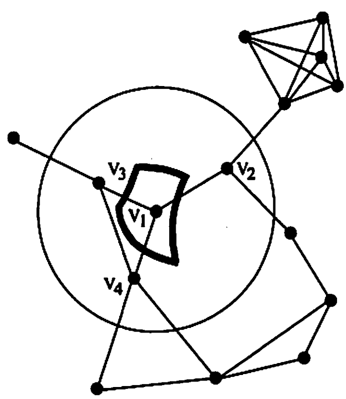


Figure 2: Illustrating a Free-Region of a node.

Definition 1 *The* **free-region** *of a node v_i, denoted by $FR(i)$ is the connected set of points in its neighborhood that preserves the connectivity of the network. This means if we move node v_i to any point in the free-region, the network connectivity does not change.*

A free-region $FR(i)$ of a node v_i is called *maximal* if it is not a proper subset of any other free-region of v_i. Fig. 2 illustrates a free-region for node v_1. It can be verified that that this free-region is also maximal.

Consider the *outer circle* $OC(i)$ of radius $2r$ centered at node v_i. The outer circle together with the transmission circle form the *annulus* $ANL(i)$ induced by node v_i. Sensor nodes lying within the transmission disk $TD(i)$ are referred to as the *in-bound nodes* of v_i. Similarly, nodes lying between the transmission circle and the outer circle are referred to as *out-bound nodes* of v_i. These definitions are illustrated in Fig. 3. The notion of free-

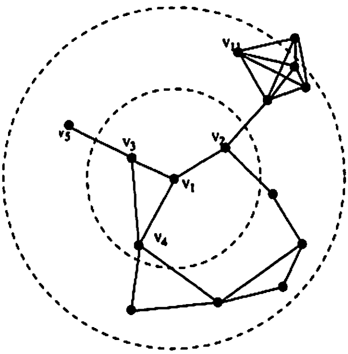


Figure 3: Illustrating Annulus, In-Bound Nodes, and Out-Bound Nodes.

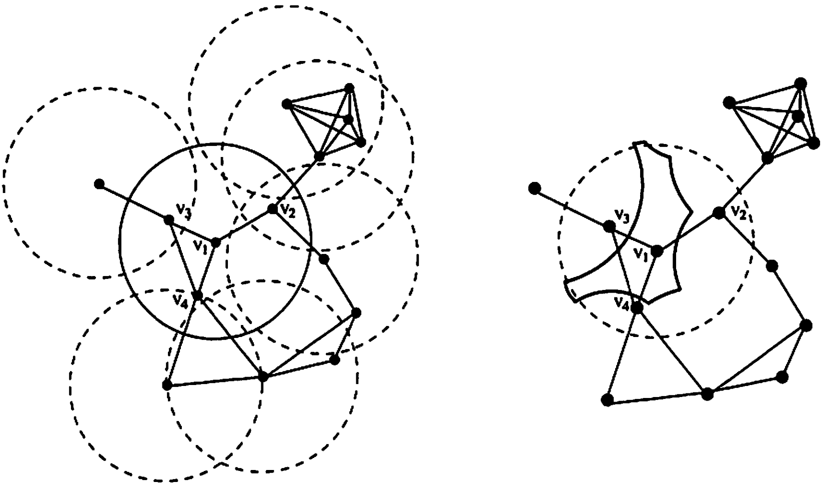
region can be captured in term of (i) the transmission disks of node v_i, (ii) its in-bound nodes, and (iii) its out-bound nodes. The region of intersection of transmission disks of in-bound nodes gives the region in which node v_i can be relocated without disconnecting with its adjacent nodes, even though some new nodes may become adjacent. This region which we call *in-free-region* $IFR(i)$ can be expressed in term of transmission disks as:

$$IFR(i) = \bigcap_j TD(j), \text{ where } j = i \text{ or } V_j \text{ is a neighbor of } v_i. \quad (1)$$

The portion of the transmission disk $TD(i)$ that overlaps with the transmission disks of its out-bound nodes is referred to as *fringe region*. The region obtained by removing fringe region from $TD(i)$ is called *out-free-region* (see Fig. 4). The out-free-region can be formally expressed as:

$$OFR(i) = TD(i) - \bigcup_j TD(j), \text{ for all outbound nodes } v_j \text{ of node } v_i. \quad (2)$$

It is noted that as long as a node stays within its out-free-region, the set of nodes that were outside its transmission range at the initial position will



b: Formation of out-free region

Figure 4: Illustrating an Out-Free-Region of a Node.

continue to remain outside. It can be observed that the free-region $FR(i)$ of node v_i is given by the intersection of its in-free-region and out-free-region. In fact, the maximal free-region shown Fig. 2 is the intersection of free-regions shown in Fig. 5 and Fig. 4. (Note, in Fig. 4, the transmission circles for the nodes labelled v_8, v_{13}, and v_{14} in Fig. 1 have been left out in order to keep the picture simple; they do not impact the final out-free region anyway.)

$$FR(i) \quad = \quad \bigcap(OFR(i), IFR(i)) \tag{3}$$

Remark 1: Both $OFR(i)$ and $IFR(i)$ are bounded regions whose boundary consists of arc-chains. Such regions are essentially special polygons whose edges are circular arcs and we refer to them as *arc-gons.*

It is interesting to look into the structural properties of free-regions. Even if a node v_i has a single neighbor, its in-free-region $IFR(i)$ is not empty. The following properties of free-regions can be verified easily.

Property 1: The in-free-region $IFR(i)$ of any node v_i that has at least one neighbor is non-empty.

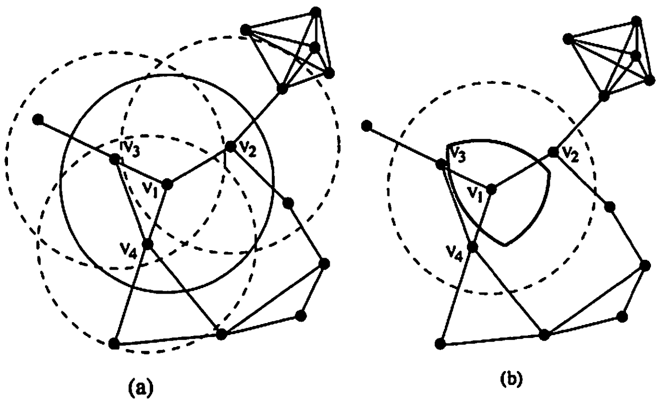


Figure 5: Transmission Circles and In-Free-Region.

Property 2: The in-free-region $IFR(i)$ of any node v_i is a convex arc-gon.

For most node distributions the arc-gon representing $IFR(i)$ will have only a few number of arcs. For some special node distributions the arc-gon could have $O(n)$ arcs. Such examples can be constructed easily and are omitted in this paper.

Property 3: In-free-region $IFR(i)$ can have $O(n)$ arcs in the worst case.

We now turn to the first of the main results of this paper; namely the low bound results of computing free-regions. It is interesting to note that the problem of computing the free-region of a sensor node is at least as difficult as the sorting problem. This can be established by examining the lower bound for computing the out-free region.

Theorem 1 *The sorting problem is transformable to the out-free-region problem (OFR) in linear time. Hence OFR has a lower bound of $\Omega(k \log k)$ in the comparison tree model of computation, where k is the number of in-bound and out-bound vertices of v_i.*

Proof: *Given a set of numbers $a_1, a_2, a_3, ..., a_k$ to sort, we map them to points in two dimensions as follows. Let l and u be the minimum and maximum values of the input numbers. We transform each a_i to Θ_i by using the formula $\Theta_i = 360 * (a_i - l)/(u - l)$. By this transformation, each Θ_i falls in the range 0-360 and the initial order of the input number is preserved.*

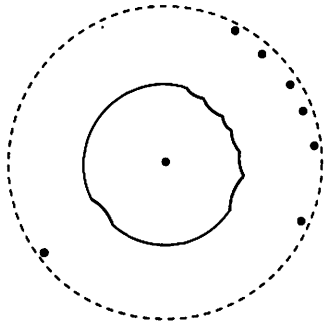


Figure 6: Reducing Sorting Problem to Out-Free-Region Computation Problem.

These angles are used to locate points. Let (x_i, y_i) *be coordinates of node* v_i. *Corresponding to number* a_j, *we locate a node at* $(x_i + 2r' \cos\theta_j, 2r' \sin\theta_j)$, *where* r' *is little less than* r. *This results in out-bound nodes of* v_i *at distance* $2r'$ *along the outer circle as shown in Fig. 6. The out-free-region* $OFR(i)$ *for this distribution of nodes consists of* k *very small arcs near the transmission circle of* v_i. *From the* $OFR(i)$ *we can read off the original input numbers in sorted order. The time required to make the transformation is linear. Since* $\Omega(k \log k)$ *is the lower bound for the sorting problem we conclude that* $\Omega(k \log k)$ *is also lower bound for the problem of computing* $OFR(i)$. □

An exact algorithm for computing $OFR(i)$ can be developed by using an incremental approach in which out-bound nodes are processed one at a time. Such an approach takes $O(k^2)$ time and the detail are available in [6, 7].

3 Approximation Algorithms

The shape of a free-region can be non-convex and complicated and exact algorithms for determining such shapes tend to become non-simple. For practical application, it would be desirable to have easily computable convex shapes. In this section we describe approaches for developing simple approximation algorithms for obtaining such solutions. The notions of *image points* and *antipodal points* are needed to develop the intended algorithms. Image points are defined for out-bound nodes, antipodal points are defined for in-bound nodes as follows.

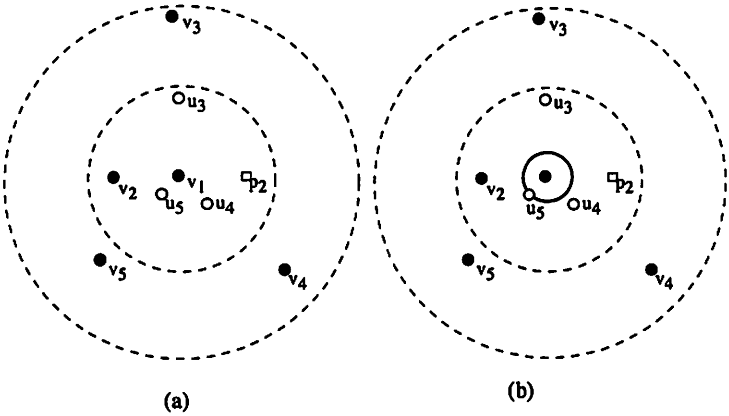


Figure 7: Illustrating (a) Image Points, Antipodal Points, and (b) Empty Circle Approximation.

Definition 2 *The* **image point** u_j *of an out-bound node* v_j *is defined as the point located at distance* $(2r - d(i,j))$ *from* v_i*, where* $d(i,j)$ *is the distance of* v_j *from* v_i *and* r *is the transmission radius. In Fig. 7 (a), the image points of three out-bound nodes are shown as unfilled white dots.*

Definition 3 *The* **antipodal point** *of an in-bound node* v_j *is defined by considering the circle* $C(i,j)$ *with center at* v_i *and passing through* v_j*. The point* p_j *on the circle* $C(i,j)$*, diametrically opposite to* v_j*, is its antipodal point.*

Definition 4 *The collection of inbound nodes, antipodal points, and image points are together referred to as pseudo nodes.*

These definitions are illustrated in Fig. 7 (a), where image points and antipodal points are drawn as small unfilled dots and unfilled squares, respectively.

3.1 Empty Circle Approximation

Consider a circle enclosing node v_i that does not enclose non of the pseudo nodes. Such circles as *empty circle.* One of the easiest way to obtain an empty circle is find the nearest pseudo node x from v_i and construct a circle centered at v_i and passing through x as shown in Fig. 7 (b).

Empty Circle Approximation Algorithm	
Input:	(i) Sensor node v_i, and (ii) Its in-bound and out-bound nodes.
Output:	A circle approximating the free-region of v_i.
Step 1:	/* Determine pseudo nodes */ (i) Find image points of out-bound nodes of v_i. (ii) Find antipodal points of in-bound nods of v_i.
Step 2:	Determine the distance r' to nearest pseudo node by examining the coordinates of image points, antipodal points, and in-bound nodes.
Step 3:	Output the disk centered at v_i and radius r' as the free-region.

It can be easily verified that the empty circle approximation algorithm can be executed in $O(k)$ time, where k is the number of in-bound and out-bound nodes of node v_i.

3.2 Convex Region Approximation

In this approach we seek to construct a convex polygon containing the candidate node v_i that does not encloses any pseudo nodes. Corresponding to each out-bound node v_j we construct a directed line L_j called the *separating line*, which passes through its image point u_i and is perpendicular to the line through v_j and v_i. The direction of the separating line is such that the candidate node v_i lies to the left of the line. The transmission circle of each in-bound node is approximated by a regular polygon of size c, where c is some constant integer. A typical value of c is 8. The approximating regular polygon corresponding to in-bound nodes is required to satisfy the "antipodal property" which is stated as follows.

Definition 5 *An inscribed regular polygon approximating the transmission circle of an in-bound node v_j is said to have* **antipodal property** *if one of its vertices passes through the antipodal point of v_j.*

The bounding edges of the approximating polygons are assigned direction implied by the counterclockwise traversal of its boundary. Consider the left half-plane corresponding to each directed line. The intersection of the right half planes of directed lines (all separating lines and lines corresponding to all regular c-gons) give a convex polygon which can be taken as an approximation for free-region $FR(i)$ of node v_i. (Fig. 8 shows partial construction.) This approximation scheme is listed as Convex Polygon Approximation Algorithm.

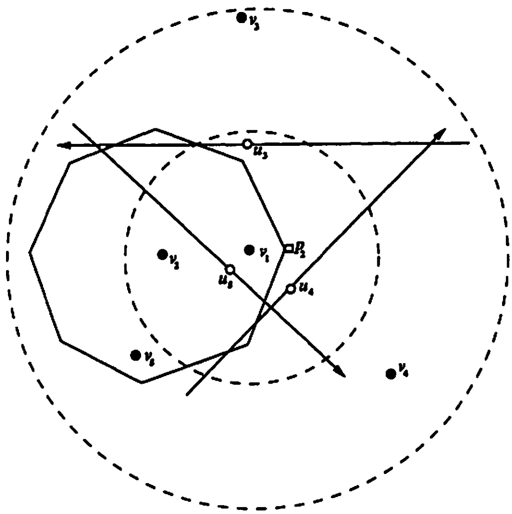


Figure 8: Illustrating Convex Approximation.

Convex Approximation Algorithm

Input:	(i)	Node v_i and its in-bound and out-bound nodes,
	(ii)	A constant c.
Output:	A convex polygon approximating the free-region of v_i.	
Step 1:	For each in-bound node v_j of v_i do	
	(i)	Find antipodal point p_j of v_j.
	(ii)	Construct the regular c-gon inscribed in the transmission circle $TC(j)$ and satisfying the antipodal property.
	(iii)	Assign direction to the bounding edges of the c-gon implied by its counterclockwise traversal.
Step 2:	(i)	Construct a regular c-gon inscribed in $TC(i)$.
	(ii)	Assign direction to the bounding edges of the c-gon implied by its counterclockwise traversal.
Step 3:	For each out-bound node v_j of v_i do	
	(i)	Find the image point u_j of node v_j.
	(ii)	Construct the corresponding directed separating line L_j.
Step 4:	Construct the intersection region R of the left half planes implied by all directed lines and output it as the free-region.	

Lemma 2 *The output region R generated by Convex Approximation Algo-*

rithm is (i) a sub-set of maximal free-region $FR(i)$ and (ii) contains vertex v_i.

Proof: *Since the transmission circles are approximated by inscribing regular c-gons, their intersection region R' is a subset of the in-free region $IFR(i)$. Any point in the right half-plane of the directed separating line is at a distance greater than the transmission range. This implies that any point in the intersection region Q' of the half planes of separating lines is at a distance at least r from any out-bound nodes. Since R is the intersection of R' and Q' it is a proper sub-set of $FR(i)$* □

Theorem 3 *Convex Approximation Algorithm can be executed in $O(k \log k)$ time, where k is the total number of in-bound and out-bound nodes of v_i.*

Proof: *For a constant c, a regular c-gon having the antipodal property can be constructed in time $O(1)$. Since there are $O(k)$ in-bound nodes, all c-gons corresponding to in-bound nodes can be constructed in $O(k)$ time. Hence, step 1 takes time $O(k)$. Step 2 takes $O(1)$ time. Separating he lines corresponding to out-bound nodes can be constructed in time $O(1)$. Since there are $O(k)$ out-bound nodes, all separating lines, and hence step 3, takes time $O(k)$. . The problem of computing the intersection of k half-planes can be reduced in linear time to the problem of computing the convex hull of k points in two dimensions by using duality methods of computational geometry [3]. Since the convex hull of k points can be computed in $O(k \log k)$ time [3], it implies that Step 4 can be done in $O(k \log k)$ time. Hence the total execution time for all steps adds to $O(k \log k)$.* □

4 Discussion

We introduced the notion of free-regions for sensor nodes. We investigated several interesting properties of free-regions and established a tight lower bound of $\Omega(k \log k)$ for the free-region computation problem. We also presented efficient approximation algorithms for constructing the free-region of sensor nodes.

Several extensions of the proposed problems and algorithms are planned as future work. It would be very interesting to establish a tight bound on the quality of the solution obtained by the proposed approximation algorithms. So far we have only determined the free region of sensor nodes; if we move a node in the free-region its local connectivity is not changed but it can change the free-regions of other nodes in its proximity. One interesting related problem would be to develop an algorithm to identify those nodes that will have very small change in their neighbor's free-region when they are relocated.

References

[1] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. *Wireless Networks*, 7(6):609–616, 2001.

[2] M. Burkhart, P. Rickenbach, R. Wattenhofer, and A. Zollinger. Does Topology Control Reduce Interference? In *International Symposium on Mobile Ad Hoc Networking and Computing, Proceedings of the 5th ACM International Symposium on Mobile ad hoc Networking and Computing, Geometry and Positioning*, pages 0–10, 2004.

[3] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho. Grid Coverage for Surveillance and Targe Location in Distributed Sensor Networks. *IEEE Transactions on Computers*, 51:1448–1453, 2002.

[4] V. Coskum. Relocating Sensor Nodes to Maximize Cumulative Connected Coverage. *Wireless Sensor Networks*, 8:2792–2817, 2008.

[5] K. Kar and S. Banerjee. Node Placement for Connected Coverage in Sensor Network. In *Proceedings of WiOpt*, 2003.

[6] N. Rongratana. Relocation and Deployment of Sensor Nodes. Master's thesis, University of Nevada, Las Vegas, 2008.

[7] N. Rongratana, L. Gewali, H. Selvaraj, and J.B. Pedersen. Free-Regions of Sensor Nodes. *Accepted for Publication in the Journal of Systems Sciences*, 2009.

[8] H. Zhang and J. Hou. Maintaining Sensing Coverage and Connectivity in Large Sensor Network. Technical Report UIUCDCS-R-2003-2351, University of Illinois, Urbana-Champain, 2003.