Scheduling Nodes in Wireless Sensor Networks: A Voronoi Approach

Marcos Augusto M. Vieira*, Luiz Filipe M. Vieira*, Linnyer B. Ruiz*†,
Antonio A.F. Loureiro*, Antonio O. Fernandes*, José Marcos S. Nogueira*

* Department of Computer Science
Federal University of Minas Gerais
Belo Horizonte, MG, Brazil

† Department of Informatics
Pontifical Catholic University of Paraná
Curitiba, PR, Brazil

{mmvieira, lfvieira, linnyer, loureiro, otavio, jmarcos}@dcc.ufmg.br

Abstract

A wireless sensor network is a special kind of ad-hoc network with distributed sensing and processing capability that can be used in a wide range of applications, such as environmental monitoring, industrial applications and precision agriculture. Despite their potential applications, such networks have particular features imposed by resource restrictions, such as low computational power, reduced bandwidth and specially limited power source. In case of a network with a high density of sensor nodes, some problems may arise such as the intersection of sensing area, redundant data, communication interference, and energy waste. A management application is necessary to make the most of network resources. On the other hand, a high-density network can introduce a fault-tolerant mechanism, increase precision, and provide multi-resolution data. The network density control depends on the application. In this paper, we propose a method to set up which nodes should be turned off or on. The management may take the sensor node out of service temporally. Our design uses a Voronoi Diagram, which decomposes the space into regions around each node. That schema could be used in a management architecture for a wireless sensor network.

1. Introduction

The maturing of integrated circuitry, micro electromechanical systems (MEMS), digital signal processing and low-range radio electronics on a single node has led to the design of wireless sensor network. This network may have hundreds or thousands of sensor nodes, each one with ability to sense its environment, perform simple computations, and communicate to its neighbors.

A large number of sensor nodes allows the sensing on a larger geographical region with a greater accuracy than previously possible. This type of network has the potential for innumerable applications cite1, including weather monitoring, security and tactical surveillance, and environment monitoring.

A wireless sensor network differs from other networks, having some unique characteristics. The most important feature is the need to be energy efficiency. A sensor node has a finite energy reserve supplied from a battery. It is often unfeasible to recharge the node's battery. Thus, the design of a wireless sensor network should be as energy efficient as possible.

In case of a network with a high density of sensor nodes, some problems may arise such as the intersection of sensing area, redundant data, communication interference, and energy waste. A management application is necessary to make the most of network resources. On the other hand, a high-density network can introduce a fault-tolerant mechanism, increase precision, and provide multi-resolution data. The network density control depends on the application.

In this paper, we propose a mechanism to control the network density based on a criterion to decide which nodes should be turned off or on. Then, we present a management function to solve this problem, which can take the sensor node out of service temporally. Our solution is based on the Voronoi Diagram, which decomposes the space into regions around each node, to determine which sensor node should be turned off or on.

To evaluate our design, we perform a simulation comparison. We evaluate the scheduling of nodes varying the network density. We show that our design can save energy without losing sensing area. This schema could be used in management architecture for Wireless Sensor Network [9].

The rest of this paper is organized as follows. Section 2 discusses some of the work related to this paper. Section 3 describes the system model and the scheduling schema. In Section 4, we present the experimental results for the scheduling schema. Finally, Section 5 presents our concluding remarks.

2. Related Work

Recently, there has been a lot of interest in wireless sensor networks. Most of it is related to energy-aware routing protocols. Chandrakasan et al. [5] proposed LEACH as an energy efficient communication protocol for wireless sensor networks. LEACH is a cluster-based routing algorithm in which a self-elected cluster head collects data from all the sensor nodes in its cluster, aggregates the collected data by executing data fusion algorithms, and transmits the data directly to the base station. A self-elected cluster head continues to be a cluster head for a period referred as a round. At the beginning of each round, every node determines if it can be a cluster head during the current round. If it is the case, the node announces its decision to its neighbors.

An interesting work to obtain aggregated information is Residual Energy Scan [12]. Zhao et al. propose an efficient monitoring infrastructure for wireless network sensor. Analog to a weather map or air-traffic radar images, the sensor network scans the geographical distribution of network resources, or activity of a sensor field. Instead of the detailed information of the residual energy at individual sensors, the scan provides an abstract view of the energy resource distribution. Sensor network scans could be used to help guide incremental deployment of sensors, but they do not have the precision to inform the best position to add a sensor node into the network and neither the amount of new nodes required to cover the desired monitoring area.

A Voronoi diagram has already been applied to solve other problems in a wireless sensor network. Meguerdichian et al. [6] proposed an algorithm for calculating the maximal breach and maximal support paths in a sensor network based on a Voronoi diagram.

A node-scheduling scheme was developed by Tian and Georganas [11]. In their approach, nodes take turns in saving the energy without affecting the service provided. The node-scheduling scheme turns some nodes on or off and certain redundancy is still guaranteed. A node decides to turn it off when it discovers that its neighbors can help it to monitor its whole area. The solution does not suppose a global knowledge of the network and is performed locally at each node. Thus, it does not guarantee the optimal solution. The proposed scheme increases communication cost, requires synchronization, and involves the calculation of a geometric representation's intersection. The cost of calculating the regions defined on plane by n circles is exponential (with n circles, there can be $2n$ such regions).

The MANNA [9] management architecture defines a management service of node density maintenance that could use our design.

3. Scheduling Nodes and Voronoi Diagram

In this section, we describe our system model, define a schema for node scheduling, briefly explain the concept of a Voronoi Diagram, and present our design. Some examples will illustrate the idea of our schema and why scheduling nodes are important for a wireless sensor network.

3.1. System Model

Figure 1 shows the system model we use in this work. We have a desired area (A) that we wish to monitor, and a set of N wireless sensors that together define a cover monitoring area. This area gives the fraction of the desired area that is actually being monitored. We will define it as a quality of service (QoS) metric of the network. The data from the network is sent to a base station.

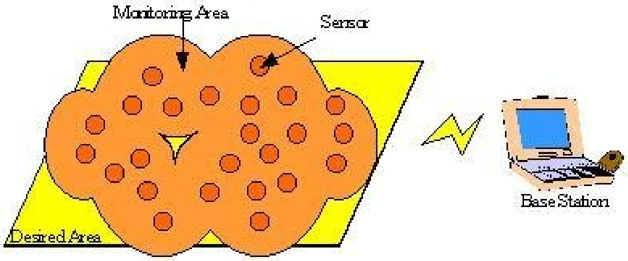


Figure 1. System Model.

The density of sensor nodes in the network (ρ) is a function of the number of sensors (N) in a desired area (A) as shown in Equation 1.

$$\rho = N/A \tag{1}$$

A wireless sensor network can have hundreds to thousands of sensor nodes. However, this information is incomplete if it is not clear the density of sensor nodes in the network, i.e., the distribution of nodes in the monitoring environment.

3.2. Assumptions

We work with a flat and homogenous wireless sensor network distributed on a 2-D plane field, but we could easily extend to n dimensions. Each node is immobile, although the network topology can be dynamic, since nodes can become unavailable permanently or temporarily.

Each node knows its location on the plane. The position does not need to be global, and can be relative to the base station or to a known point. As point out in [12], obtaining reliable node location has been studied in different contexts. Using Global Positioning System (GPS), we can determine a geographical location with a good accuracy [7]. Other solution is to have common nodes calculate their distances to beacons and estimate their locations [3, 4]. Beacons are special nodes that know their coordinates in advance and can transmit periodically a signal to be processed by common nodes.

Our design uses both a sensor range and a radio range. There are three possibilities, as shown in Figure 2, when considering sensor range and radio range: sensor range greater than, less than, or equal to radio range.

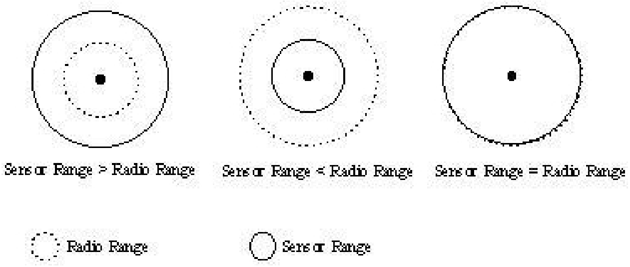


Figure 2. Sensor and Radio range possibilities.

The solution proposed in this work can use either a sensor range or a radio range as the criteria to calculate the area of the Voronoi diagram.

3.3. Scheduling Nodes

As mentioned before, some problems may arise if the network has a high density of sensor nodes. In the following, we present a management schema to deal with this problem. The management may take a sensor node out of service temporally, scheduling the nodes that will be turned on and off.

Suppose we have a network with a topology as depicted in Figure 3 and all nodes transmit at the same frequency. Node 1 wants to transmit an information to node 5 and/or node 2 wants to transmit to node 3. But, if node 4 transmits something to another node, it will cause interference at both transmissions, as illustrated in Figure 4. It could be argued that the MAC layer with carrier sense would solve it. The MAC layer will still be necessary, but if the density of the network is too high at that region, the number of collisions will increase and energy will be wasted unnecessarily. A management solution could temporally turn node 4 off in case the monitoring application does not need another node in that sensing area that is already covered by other nodes.

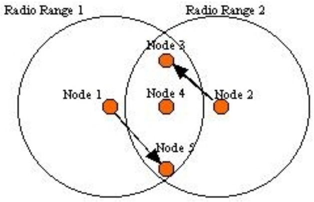


Figure 3. Example of a network topology.

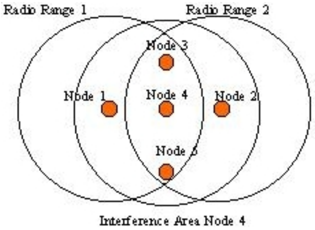


Figure 4. Interference of node 4.

Figure 5 illustrates another example of a network topology. A flat wireless sensor network uses a multi-hop connection to save energy and increase the communication range of of a node. Suppose node 1 wants to communicate to node 3. If the information has to pass through node 2, it will waste energy and bandwidth and increase latency. Node 1 could transmit directly to node 3. A management solution could turn node 2 temporally off. It will still be necessary to use a multi-hop communication, in case node 1 wants to communicate to node 4. The important point is that increasing the number of hops does not necessary saves energy. An interesting work that discusses this and other issues is [2]. Given the network topology and the distance D from node 1 to node 4, the optimal number of hops to save energy is $D/dchar$, where the distance $dchar$ is called the characteristic distance and is independent of D and depends on the transceiver hardware. If a different value is used it leads to energy inefficiencies.

The same examples that showed radio interference can be used to illustrate sensing interference. A large number of sensor nodes in a sensor field makes it infeasible to collect redundant detailed state information from each individual sensor node, given energy and communication constraints. Pruning redundant sensing information is an important task in a wireless sensor network that saves energy.

The scheduling node schema discussed in this work may be used for different purposes. Here we present two ideas. When an active node leaves the network, due to energy

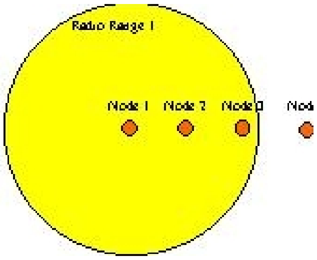


Figure 5. Another example of a network topology.

problems for instance, the management may activate some nodes that are off. Energy is saved and coverage of the monitoring area is not affected. The second purpose is for security reason. The nodes that are not sensing could stay in a mode listening to the network, verifying if the nodes are transmitting what they received, increasing the security of the network. When necessary, the node that is off would become an active node.

To determine if a node is going to be turned off, we will use a Voronoi diagram, which is explained in the following.

3.4. Voronoi Diagram

Let $S = \{p_1, p_2, \ldots, p_i, \ldots, p_n\}$ be a set of points in a two-dimensional Euclidean plane. These points are called sites. A Voronoi diagram decomposes the space into regions around each site, such that all points in the region around p_i are closer to p_i than any other point in S.

Using the definition in [8], the Voronoi region $V(p_i)$ for each p_i is expressed as:

$$V(p_i) = \{x : |p_i - x| \leq |p_j - x|, \forall j \neq i\}$$

$V(p_i)$ consists of all points that are closer to p_i than any other site. The set of all sites form the Voronoi Diagram $V(S)$.

The follow example, extracted from [8], illustrates a simple Voronoi diagram. Consider two points p_1 and p_2. Let $B(p_1, p_2) = B_{12}$ be the perpendicular bisector of the segment $\overline{p_1 p_2}$. Then every point x on B_{12} is equidistant from p_1 and p_2. This can be seen by drawing the triangle (p_1, p_2, x) as depicted in Figure 6. By Euclids side-angle-side theorem, $|p_1 x| = |p_2 x|$.

To sum up, given input points presented in Figure 7a, the corresponding Voronoi diagram is depicted in Figure 7b.

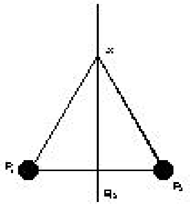
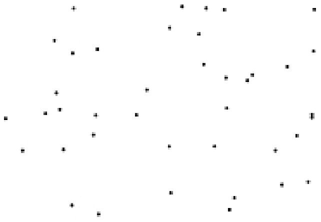
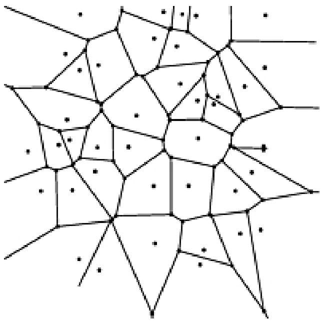


Figure 6. Two points $|p_1 x| = |p_2 x|$.



(a) Input points



(b) Voronoi diagram

Figure 7. A set of points and its Voronoi Diagram [10].

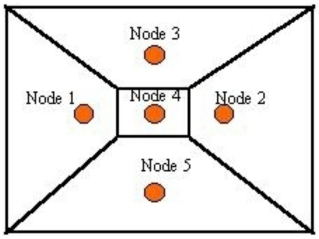
3.5. The Algorithm

In this section, we discuss the algorithm used to calculate which nodes are turned on or off. Given the location of the nodes and the area to be monitored, each node represents a point, and the desired area to monitor is the polygon that is defined by the Voronoi diagram. The objective is to determine the area each node is responsible for. Then, we pick up the node with the smallest area and if it is too small, the node should be turned off. The neighbors of that node become responsible for that area, updating the Voronoi diagram. This process continues until there is a node responsible for an area smaller than a given threshold. Figure 8 shows the pseudo-code of the algorithm.

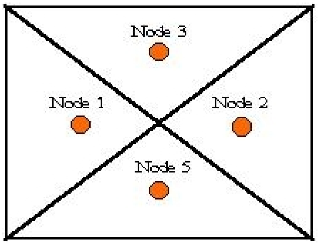
The algorithm is illustrated taking the network topology of Figure 3. Figure 9a shows the Voronoi diagram. Because node 4 is responsible for a small area (smaller than a threshold), the algorithm decides that node 4 should be turned off. Figure 9b illustrates the new topology. At this step, there is no node responsible for a smaller area than the threshold, and it ends.

```
Input: set of points.
Output: nodes that should be turned off.
Calculate Voronoi();
do begin
    for every node begin
        calculate Vonoroi_Area();
    end
    get SmallestArea();
    if (smallest_area $<$ threshold) then
        begin
            node_responsible $\leftarrow$ turn off;
            update_Voronoi();
            keep_searching $\leftarrow$ true;
        end
    else keep_searching $\leftarrow$ false;
end
while (keep_searching)
```

Figure 8. Algorithm.



(a) Initial Voronoi diagram



(b) New Voronoi diagram

Figure 9. Example of the algorithm that uses Voronoi diagram to decide if a node should be turned off or on.

The worst-case complexity for calculating the Voronoi diagram is $\Theta(n \log n)$. Thus, the algorithm seems feasible to be executed in a base station. A simple naive approach, to update the Voronoi diagram, is to rebuild it, with a worst-case complexity of $\Theta(n^2 \log n)$. However, an incremental approach to update the Voronoi diagram could be used.

3.6. Discussion

A local algorithm for deciding the scheduling of nodes could use the idea presented in [5]. A self-elected cluster head collects data position from all sensor nodes in its cluster, calculates the Voronoi diagram, and transmits its decision back to the nodes in a distributed fashion. However, there are some disadvantages: it spends energy to choose a cluster head and to transmit the information to each node; the decision if a node should be turned on or off is very important and should be done in a management layer, since it could affect the entire network; and it does not solve the problem in the neighborhood of each cluster head.

If the network is hierarchical, we can devise three options. If the cluster head does not sense, it can be left out of the Voronoi diagram algorithm. The second option is to treat the cluster head as a common node using the previous design. The third option is to assign a weight to each node. Some choices are Multiplicatively Weighted Voronoi diagram and Additively Weighted Voronoi diagram. Let *dis* represent the Euclidean distance and w_i be the weight of each point p_i. A Multiplicatively Weighted Voronoi diagram is generated by using a distance function in Equation 2. An edge is generally a circular arc. An Additively Weighted Voronoi diagram is generated by using a distance function in Equation 3. An edge is generally a hyperbolic arc.

$$d(p, p_i) = dis(p, p_i)/w_i \tag{2}$$

$$d(p, p_i) = dis(p, p_i) - w_i \tag{3}$$

In case a sensor network has mobile nodes, a Voronoi diagram of moving points can be applied [1].

4. Experimental Results

In order to evaluate our design, we created a Java application to simulate our experiments for large-scale sensor networks. In this section, we present our results and discuss their implications.

4.1. Metrics

The key performance criterion in most wireless sensor networks is energy. We consider the energy saving as the number of off-nodes that our design outputs. Since there are many types of nodes with different types of energy level, our energy saving metric is the number of off-nodes, which is independent of sensor node types.

Another metric used is the percentage of the desired sensing area that is not being covered by the network. The idea is to verify if sensing area is lost when a node is turned off.

A node that is turned off can also be described as a backup node. It is turned on when an operating node is not working properly, such as when it does not have enough energy or does not respond anymore.

4.2. Settings

Our experiments were conducted on a square sensing area. For all experiments, the network size is 100 nodes. To vary the density, we change the area. The position of each node is generated at random. We set up the threshold area as being a percentage of the sensing area of each sensor:

THRESHOLD AREA = PERCENTAGE COEFFICIENT ×
SENSING AREA

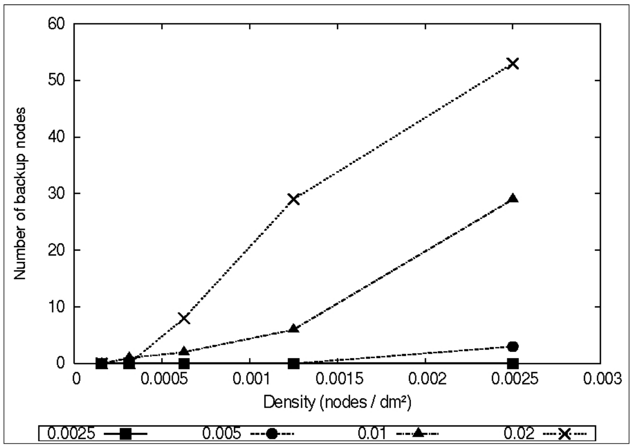
4.3. Results

Figure 10 shows the number of backup nodes as the density changes, for different threshold areas. Figures 10 10a, b and c differ the sensor range, which is respectively 89, 178 and 356 dm. As expect, when the density grows, the number of backup nodes also increases. This also happens when we compare the threshold of the Voronoi area and the number of backup nodes. Figure 10c illustrates a degenerate case, when one node is sufficient to cover the desired area, turning all the others 99 nodes into backup nodes. This is the best case, since our design saves 99 percent of the network energy. Figure 10a illustrates when the network density is low, and there is no backup node.

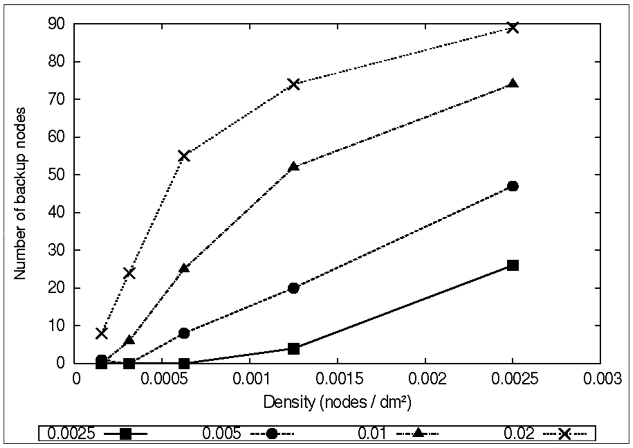
Table 1 shows the area not covered by increasing the number of backup nodes. Because the threshold was low, no sensing area was lost in the experiments. Thus the savings depend on network density.

5. Concluding Remarks

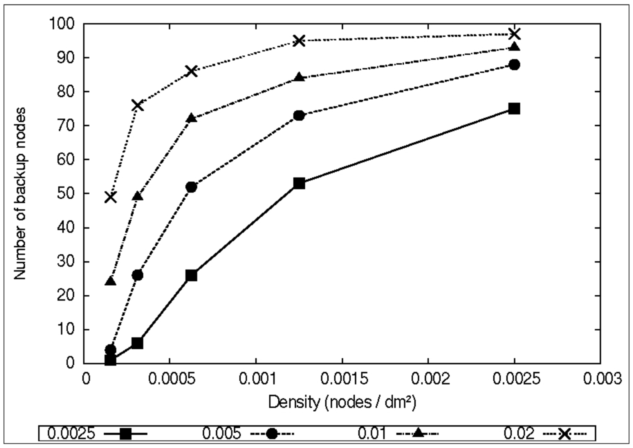
Verifying if sensor nodes are actually monitoring a desired area is an important metric for wireless sensor networks. We define the coverage of a monitoring area as a quality of service metric of the network. It gives the percentage of the desired area that is actually being monitored. Energy savings in a wireless sensor network is critical, thus a management application is necessary to make the most of the network resource. Our design defines a schema for saving network resources. We presented the idea of backup node, and defined and evaluated a criterion for determining which sensor nodes should be turned off. Simulation results show that our approach is scalable and presents energy-efficiency characteristics. The amount of backup nodes depends on the network density. It can save energy without loosing sensing area.



(a) Number of backup nodes × density, varying the threshold area, sensor range is 89 dm.



(b) Number of backup nodes × density, varying the threshold area, sensor range (178 dm) is the double of (a).



(c) Number of backup nodes × density, varying the threshold area, sensor range (356 dm) is the double of (b).

Figure 10. Results of experiments.

% Coefficient = 0.0025		**% Coefficient = 0.05**		**% Coefficient = 0.01**		**% Coefficient = 0.02**	
% BackUp nodes	**% Area not covered**	**% BackUp nodes**	**% Area not covered**	**% BackUp nodes**	**% Area not covered**	**% BackUp nodes**	**% Area not covered**
0	0	1	0	0	0	8	0
0	0	0	0	6	0	24	0
0	0	8	0	25	0	55	0
4	0	20	0	52	0	74	0
26	0	47	0	74	0	98	0

Table 1. Area not covered by the sensors × the percentage of backup nodes at the network, for different values of the threshold area.

As a future work, we plan to study and evaluate some of the issues stated in Section 3.6.

Acknowledgment

This work has been partially supported by CNPq, Brazil, under process 55.2111/2002-3.

References

[1] G. Albers et al. Voronoi diagrams of moving points. *International Journal of Computational Geometry and Applications*, 8(3):365–380, 1998.

[2] M. Bhardwaj, T. Garnett, and A. Chandrakasan. Upper bounds on the lifetime of sensor networks. In *IEEE International Conference on Communications*, pages 785–790, 2001.

[3] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, 2000.

[4] L. Girod. Development and characterization of an acoustic rangefinder. Technical Report TR–00–728, University of Southern California, 2000.

[5] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocols for wireless microsensor networks. In *33rd International Conference on System Sciences*, pages 3005–3014, 2000.

[6] S. Meguerdichian et al. Coverage problems in wireless ad-hoc sensor networks. In *IEEE INFOCOM*, pages 1380–1387, 2001.

[7] B. Nath et al. Special issue on “smart spaces and environments”. *IEEE Personal Commununications*, 7(5), October 2000.

[8] J. O’Rourke. *Computational Geometry in C*. Cambridge University Press, second edition, 1998.

[9] L. Ruiz, J. Nogueira, and A. Loureiro. Manna: A management architecture for wireless sensor networks. *IEEE Communications Magazine*, 41(2):116–125, February 2003.

[10] S. Skiena and S. Skiena. *The Algorithm Design Manual*. Telos Pr, 1997. Voronoi diagrams: `http://www.cs.sunysb.edu/˜algorith/files/voronoi-diagrams.shtml`.

[11] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *First ACM International Workshop on Wireless Sensor Networks and Applications*, pages 32–41, 2002.

[12] Y. Zhao, R. Govindan, and D. Estrin. Residual energy scans for monitoring wireless sensor networks. In *IEEE Wireless Communications and Networking Conference*, pages 356–362, 2002.