A Geometric Tiling Algorithm for Wireless Sensor Networks

Adam P. MartinezÀ, Timothy NorfolkÀ, and Kathy J. Liszka ÿ

†The University of Akron
Department of Mathematics
norfolk@uakron.edu

ÿ The University of Akron
Department of Computer Science
liszka@uakron.edu

Abstract

The cover generation problem is relevant to the problem of creating large-scale wireless sensor networks with short-ranged sensor nodes that may not be capable of transmitting to the base station. Quickly and efficiently placing relay nodes allows the sensors to save on battery power and transmit information back to the base station via the relay nodes. Placing a minimal cover of relays is at least an NP-hard problem. We present a geometric tiling algorithm to construct an approximation to a minimal covering set in O(n) time.

Keywords: geometric tiling, minimal covering sets, wireless sensor networks

1. Introduction

There are a wide variety of polynomial time approximation schemes (PTAS) that can approximate solutions to the minimum geometric disk cover (MGDC) problem, but none in current literature can do so in O(n) runtime. We present an algorithm that computes an approximation to the MGDC problem with reasonable disk-to-point efficiency for many instances of the problem in linear runtime. The motivation for this research is to computer optimal designs for building wireless sensor networks (WSNs). Some WSN structure problems can be cast as MGDC problems. While for many applications long runtimes of current algorithms are not an issue, for time-sensitive problems or very large regions with large sets of sensor nodes (SNs), computing a covering set of relay nodes (RNs) can take unreasonably long.

Some applications can tolerate tens of hours computing an optimal networking solution that requires as few relays as possible. However, not all networking environments have the luxury of unlimited design and setup time. For time-sensitive applications, computing a fast and reasonably accurate solution to a covering set of a network can achieve a ñgood enoughò solution that will save lives. The network will be more costly, but it can start being built immediately. This kind of algorithm could be useful for providing real-time logistical and tactical information to moving front-line military units and ensuring that search-and-rescue teams have real-time information. Because these kinds of environments do not tolerate time delays, a less efficient network now is far more valuable than a more efficient network later.

This paper presents a geometric tiling algorithm for approximating a minimal covering set in the context of a two-tiered, single-hop WSN. This can alternately be described as an approximation scheme for the MGDC problem. The next section gives background on the problem along with related research. A formal description of the geometric tiling algorithm and an analysis of its performance is given in section 3. Analysis and experimental results are given in section 4. Conclusions and future work are given in the final section.

2. Background and Related Research

WSNs consist of a set of sensor nodes that collect information and wirelessly communicate with one another. There is either a Base Station (BS) that aggregates the information in the network or an outside access point to the network. There may or may not be RNs that act as network gateways for SNs within a small region around them. The presence of RNs determines if a WSN is *one-tiered* or *two-tiered* [1]. One-tiered WSNs have only sensor components. SNs are either deliberately placed or randomly dropped into a region, and an appropriate way to route data through the network must be found.

There has been significant research into algorithms for generating two-tiered covers of WSNs within the past decade. Two-tiered WSNs have not only a set of sensors in a region, but also have a set of relay nodes that act as network gateways for sensors in a small region around them. These algorithms usually assume a random distribution of SNs in a given region. A layer of relay nodes is placed such that the relay layer forms a cover over the sensors. The SN's sole purpose is to gather information and forward it to a local RN. For many applications the SNs are designed to be built as cheaply as possible, and thus do not have the battery power and design parameters to transmit data long distances. The RNs collect data from the SNs within a small region and relay the data either directly to or through one another back to a BS. Each RN in a single hop two-tier WSN has a direct connection to the BS. An example single-hop two-tiered network is shown in Figure 1.

Single-hop two-tier WSNs only require that the SNs transmit to the RNs, and do not require that the RNs be able to transmit to one another. The only requirement on the set of RNs is that it forms a covering set of the SNs. It is assumed that either the data will be consolidated at the RN for later collection or each RN has some capability of transmitting its information back to

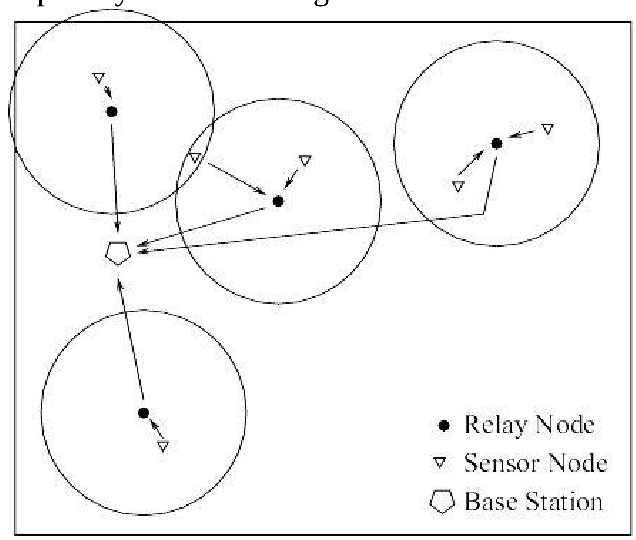

Fig. 1. A triangular grid of RNs

a BS. In the latter case, the RNs that are more distant from the BS will deplete their power more quickly than those further away [1].

Despite this drawback, this network architecture is still useful for networking environments where RNs have satellite uplinks, long range directional wireless communication or landline access. There is extensive study on this problem in terms of the minimum geometric disk cover (MGDC) and discrete unit disk cover (DUDC) problems.

2.1 Minimum Geometric Disk Cover

Given a region D containing a set P with n points, generate minimal covering set of unit disks C such that for each $p \in P$, $\exists c \in C$ such that $p \in c$.

2.2 Discrete Unit Disk Cover

Given a region D containing a set P with n points and a set of unit disks D, select a minimal covering set of unit disks $C \subseteq D$ such that for each $p \in P$, $\exists c \in C$ such that $p \in c$.

2.3 Comparison

The MGDC algorithm allows disks to be placed anywhere within the region, while the DUDC problem only allows disks to be placed in specific locations. Both the MGDC and DUDC problems have been proven to be NP-complete [2], but both also allow polynomial time approximation schemes. A PTAS generates a solution to an NP-Hard problem in polynomial time that is no more than some constant multiple of the optimal answer. A wide variety of PTAS have arisen for both of these problems. Many algorithms for DUDC have been proposed that generate solutions of no more than some constant multiple greater than one of the optimal solution in reasonable time [3, 4, 5, 6]. By comparison, algorithms for the MGDC problem generally require much longer runtimes, but can guarantee an arbitrary (1+O) level of accuracy to the optimal solution of disks placed anywhere in the region. Depending on the accuracy required and the algorithm used, MGDC and DUDC PTAS can be as fast as O(n^2) or slower than O(n^{100}). Some connected cover algorithms from the multiple-hop two-tiered problem, such as the 2CRNDC algorithm, are very similar to MGDC algorithms; only as a last step do they guarantee connectivity [7]. Some of the more recent work in this problem includes research by Liao and Hu [8], of which a modified algorithm is featured later in this work as a point of reference for the algorithm we present. Liao and Hu's algorithm build off of general set-based PTAS for approximating the MGDC [9].

3. Formulation of the Algorithm

We present an algorithm for generating a reasonably small unit disk cover of a set of points in O(n) time [11]. The approach for this algorithm relies on the uniformity of a triangular grid. Consider the problem of finding the most efficient cover of a large but finite plane using

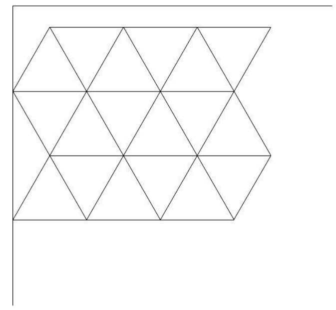

Fig. 2. A triangular grid of RNs.

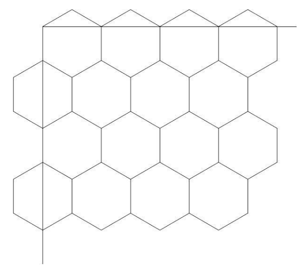

Fig. 3. A hexagonal cover of RNs.

disks of radius 1. Pompili et al. showed that the most efficient regular cover is a triangular grid of disks as in Figure 2, with a point-to-point transmission distance of ã3 [10]. However, our problem formulation does not require that we cover the entire region. We only need to provide a covering set for a set of n points in the region, representing SNs. We abstractly generate a cover of the region by overlaying a tessellation of hexagons of circumradius 1 with centers at each point on a triangular grid of edge length ã3. The hexagons in the region appear as in Figure 3. A hexagon of circumradius 1 is a regular hexagon inscribed in a circle of radius 1. Potential RN locations are only at the points on the triangular grid. Each RN only receives messages from SNs within the RN's corresponding hexagon. The algorithm iterates through the n SNs and adds the nearest point on the triangular grid to a solution set. By placing a unit disk at each point in the solution set, we produce an approximation to the minimal unit disk cover of the n points.

Problem Statement: given a region R containing a set P with n points, generate an approximation C to the minimal covering set of unit disks such that for each p ⌊ P, ∃ c ⌊ C such that p Ɑc.

Table 1 provides the notations used. Given a region filled with n SNs, we approximate a covering set. For the purpose of this formulation we will assume we are given a square region containing the SNs. In practical problems, the region would be defined as the minimal square that contains the set of SNs. Label this square region R with side length s. Our objective is to approximate the minimal cover of the SNs using a triangular grid of RNs.

The RN-SN transmission range r forms a convenient non-dimensional scaling for this problem. We call the non-dimensionalized region R_n, with side length s_n. In this region, the RN-SN transmission range is 1. By scaling all distances involved by r, the algorithm generates a cover for any size region efficiently.

We abstractly tessellate the non-dimensionalized region R_n with hexagons of circumradius 1. The RNs at the centers of the hexagons form a triangular grid. Each SN in R will be mapped to points in D_n via the transformation

$$\left(x(n,p), y(n,p)\right) = \left(\frac{x_p}{r}, \frac{y_p}{r}\right), \tag{1}$$

where (x_p, y_p) is the location os the SN p.

Given any square or rectangular region, we may orient the hexagons as in Figure 6, such that the hexagons fit neatly in the upper left corner of the region with minimal waste. We can then compute the coordinates of any RN. Tessellating the square region R_n this way requires no more than $\left\lceil \frac{s_n}{\sqrt{3}} \right\rceil + 1$ columns and $\left\lceil \frac{s_n}{1.5} \right\rceil + 1$ rows of hexagons of circumradius 1. Implementing the algorithm does not actually require the generation and storage in memory of the entire set of hexagons, merely the conceptual knowledge that we have overlaid it on the region.

We iterate through each SN and select the nearest hexagon in the grid. The regular nature of the tessellation makes finding the nearest RN a constant time arithmetic process. These RNs locations do not need to be pre-computed and then selected, as they can be computed on the fly using arithmetic and rounding.

Table 1. Variable definitions.

Variable	Description
R	The 2D region.
R_n	The non-dimensionalized 2D region.
s	The side length of R.
s_n	The side length of R_n.
r	RN ï SN transmission range.
C	An approximation to the minimal disk cover.
P	The number of sensor nodes.
n	The number of SNs.
x_p	The x-coordinate of sensor p.
y_p	The y-coordinate of sensor p.
$x_{(n,p)}$	The non-dimensionalized x-coordinate of sensor p.
$y_{(n,p)}$	The non-dimensionalized y-coordinate of sensor p.

The RNs, as they are set up in this problem, form a triangular grid. Each row on the triangular grid has the same y-coordinate. Every second row has an offset x-coordinate. We can immediately eliminate all but two potential candidates for the nearest RN to a SN by simply looking at the coordinates of the sensor. The sensor will fall between two rows of relays. Counting from the top, odd numbered rows will have a horizontal offset of ã3 / 2 from the even rows. Within each row, the x-coordinate of the SN will be closer to either the RN on its left or on its right. Each row then has a closest RN to the SN, and the closer of these two RNs is chosen to cover that SN.

We create a *selected relays* matrix M of booleans that stores whether or not the j^{th} RN in the i^{th} row of RNs must be selected to form a cover. This has the disadvantage of taking up a large block of memory by requiring a matrix of

$\left(\left\lceil \frac{s_n}{\sqrt{3}} \right\rceil + 1\right)\left(\left\lceil \frac{s_n}{\sqrt{1.5}} \right\rceil + 1\right)$ booleans, but avoids writing duplicate RNs to the solution set *C*. We then compute the location of the j^{th} RN in the i^{th} row for each true in *M*. These RNs make up *C*.

To map the locations of the RNs in R_n back to the region *R*, the corresponding location in *R* for an RN at x_n, y_n in R_n is

$$(x,y) = (rx_n, ry_n) \tag{2}$$

a set of points such that when we place an RN at each of these points, we have a cover of the SNs in *R*. Each SN will be within *r* units of distance of the nearest RN. This is an approximation to the minimal cover of disks of radius *r*.

Due to the properties of MGDC problem and the formulation of the triangular grid algorithm, there is no simple way to compare precisely how accurate of a solution the algorithm provides to the optimal solution. The MGDC problem is NP-complete, and so finding the optimal disk cover takes an unreasonable amount of time to compute for any random simulation. Additionally, the only mathematical bound the algorithm gives to the efficiency of its cover is that it is the most globally efficient layout of relays on the plane.

4. Analysis and Preliminary Experiments

The algorithm was run on a desktop machine using MATLAB R2011b. A set of points are generated using a uniform distribution in a square region of side length s = 10. This side length was chosen as it was large enough to see noticeable differences in solutions and computational times, as well as being small enough to compute in a reasonable time. The algorithm computes the locations of a triangular grid of unit disks that covers the region, then finds approximations to the minimal cover of unit disks on that grid.

Sensor densities of (n/s^2 = .1, .25, .5, .75, 1, 2) were considered to determine each the algorithm's response to networks of varying densities. An example of the output for n/s^2 = .5 is shown in Figure 4. Results of the simulation shows that the algorithm performed in O(n) time for simulations of all sizes of n. Memory usage was not a concern until the input data size grew to a very large n. Figure 5 shows the triangular grid algorithm across the trials we ran.

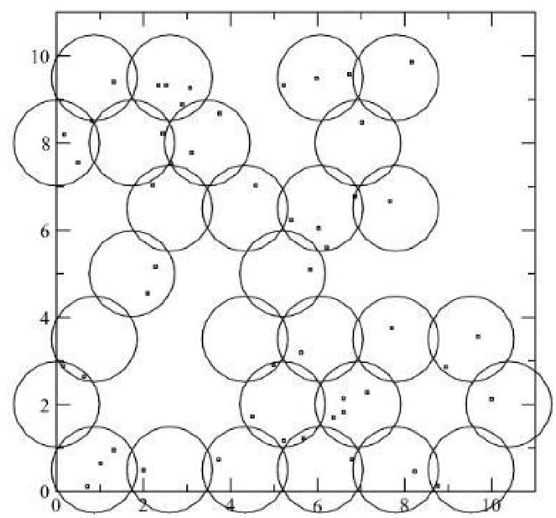

Figure 4. The triangular grid algorithm.

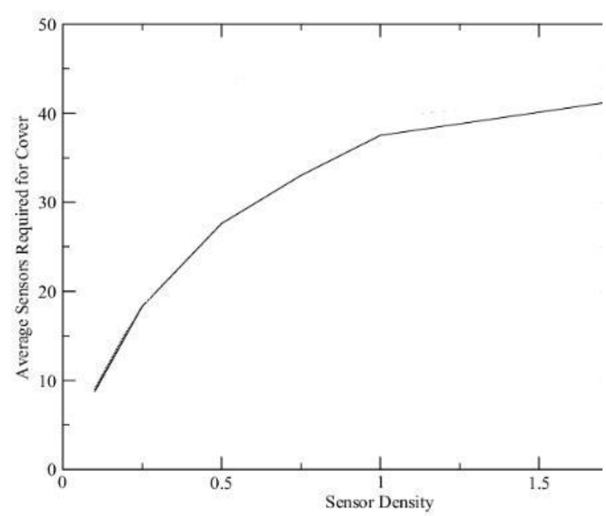

Figure 5. Average relays to cover sensor networks of various densities.

5. Conclusions and Future Work

We presented an O(n) triangular grid algorithm for approximating the minimum geometric disk cover of a set of points in a region. While the algorithm presented is not an epsilon-approximation, its speed and practical performance for generating approximations to an NP-Complete problem in linear time makes it suitable for some applications including guaranteeing coverage in dense or rapidly changing wireless sensor networks. For emergency situations and military applications, time is the primary issue for building an effective network, not cost. This algorithm provides a method for quickly generating covering sets of almost any size network.

The triangular grid algorithm could be improved. A better approximation could be found by culling relays from the grid using some simple enumerative techniques to identify unnecessary relays. Additionally, search techniques could be used to generate connected covers by finding unconnected spaces and connecting them with a shortest path of relays. While this would increase the runtime of the algorithm, it could provide fast solutions to the multiple-hop WSN problem. The runtime increase would likely be dominated by the runtime of the search algorithm. Common search algorithms, such as breadth-first and depth-first searches, are O(n^2). The algorithm would also generate an acceptable starting point for iterative methods for calculating a minimal cover.

Additionally, the algorithm deserves a comparison to an MGDC algorithm that is not restricted to a triangular grid. While the triangular grid algorithm provides adequate solutions on the grid, at this time we are looking at how the algorithm compares to a true MGDC algorithm in terms of RN-SN ratio.

6. References

[1] Xu, K., Hassanein, H.,Takahara, G. and Wang, Q., Relay node deployment strategies in heterogeneous wireless sensor networks. IEEE Transactions on Mobile Computing, 9(20:145-159, February 2010.

[2] Masuyama, S. and Ibaraki, T. and Hasegawa, T., The computational complexity of the m-center problems on the plane. *IEICE Transactions*, 1981, E64-E, pp. 57-64.

[3] Clinescu, G. and Mndoiu, I. and Wan, P. and Zelikovsky, A., Selection forwarding neighbors in wireless ad hoc networks. *Mobile Networks and Applications*, 9:101-111, 2004.

[4] Carmi, P. and Katz, M. and Lev-Tov, N., Covering points by unit disks of fixes location. In Takeshi Tokuyaa, editor, *Algorithms and Computation*, volume 4835 of Lecture Notes in Computer Science, pp. 644-655. Springer Berlin/Heidelberg, 2007.

[5] Claude, F. and Dorrigiv, R. and Durocher, S. and Fraser, R. and Lpez-Ortiz, A. and Salinger, A., Practical Discrete Unit Disk Cover Using an Exact Line-Separable Algorithm, *Algorithms and Computation*, vo, 5868 of Lecture Notes in Computer Science, pp. 45-54, Springer Berlin/Heidelberg, 2009.

[6] Das, G. and Fraser, R. and LÃ²pez-Ortiz, A. and Nickerson, B., On the discrete unit disk cover problem. Naoki Katoh and Amit Kumar, editors, WALCOM: *Algorithms and Computation*, vol. 6552, Lecture Notes in Computer Science, pp. 146-157. Springer Berlin/Heidelberg, 2011.

[7] Hao, B. and Tang, H and Xue, G., Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation. In *Workshop on High Performance Switching and Routing*, pp. 246-250, 2004.

[8] C. Liao and S. Hu, Polynomial time approximation schemes for minimum disk

cover problems., *Journal of Combinatorial Optimization*, 20:399-412, 2010.
[9] Nieberg, T. and Hurink, J. and Kern, W., Approximation schemes for wireless networks, *ACM Trans. Algorithms*, 4:49:1-49:17, August 2008.
[10] Pompili, D. and Melodia, T. and Akyildiz, I. F., Three-dimensional and to-dimensional deployment analysis for underwater acoustic sensor networks, Ad Hoc Networks, 7(4): 778-790, 2009.
[11] Martinez, A., A Geometric Tiling Algorithm for Approximating Minimal Covering Sets, a Master's thesis presented to the Graduate Faculty of The University of Akron, December 2011.