

Research Article

Linear Time Approximation Algorithms for the Relay Node Placement Problem in Wireless Sensor Networks with Hexagon Tessellation

Chi-Chang Chen,¹ Chi-Yu Chang,¹ and Po-Ying Chen²

¹Department of Information Engineering, I-Shou University, Kaohsiung 84001, Taiwan

²Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan

Correspondence should be addressed to Chi-Chang Chen; ccchen@isu.edu.tw

Received 26 June 2014; Revised 9 September 2014; Accepted 20 March 2015

Academic Editor: Chenzhong Li

Copyright © 2015 Chi-Chang Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The relay node placement problem in wireless sensor network (WSN) aims at deploying the minimum number of relay nodes over the network so that each sensor can communicate with at least one relay node. When the deployed relay nodes are homogeneous and their communication ranges are circular, one way to solve the WSN relay node placement problem is to solve the minimum geometric disk cover (MGDC) problem first and place the relay nodes at the centers of the covering disks and then, if necessary, deploy additional relay nodes to meet the connection requirement of relay nodes. It is known that the MGDC problem is NP-complete. A novel linear time approximation algorithm for the MGDC problem is proposed, which identifies covering disks using the regular hexagon tessellation of the plane with bounded area. The approximation ratio of the proposed algorithm is $(5 + \epsilon)$, where $0 < \epsilon \leq 15$. Experimental results show that the worst case is rare, and on average the proposed algorithm uses less than 1.7 times the optimal disks of the MGDC problem. In cases where quick deployment is necessary, this study provides a fast 7-approximation algorithm which uses on average less than twice the optimal number of relay nodes in the simulation.

1. Introduction

Given a set P of n points on the Euclidean plane and a prescribed radius r , disk d covers a point $p \in P$ if the distance between the center of d and p is not greater than r . The minimum geometric disk cover (MGDC) problem is to identify a set of disks with minimal cardinality covering all points in P . If radius r equals 1, the MGDC problem is called the minimum geometric unit disk cover (MGUDC) problem. To simplify calculations, this study focuses on solving the MGUDC problem. However, all MGUDC problem algorithms can easily be extended to the MGDC problem by changing the radius from 1 to any fixed real number r . Solutions to the MGDC problem can be used to solve the relay node placement problems in wireless sensor networks (WSNs).

A WSN consists of spatially distributed autonomous sensors that cooperatively monitor a region for physical

or environmental conditions, such as temperature, sound, vibration, pressure, motion, and pollution. The transmission power consumed by a wireless radio is proportional to the distance squared or even higher in the presence of obstacles. Thus, multihop routing (instead of direct communication) is usually used for sending collected data to the sink.

A method of prolonging the lifetime of a WSN and preserving network connectivity is to deploy a few costly, but powerful, relay nodes to communicate with the other sensors or relay nodes. The WSN relay node placement problem studies how to deploy the fewest relay nodes so that each sensor can communicate with at least one relay node. In this study, we assume that the relay nodes in the WSN are homogeneous and their communication ranges are circles with radius r . (If the communication range is irregular, the r can be taken as the minimum radius among the inscribed circles of the irregular ranges.) Therefore, the relay node placement problem of WSNs can be treated as solving

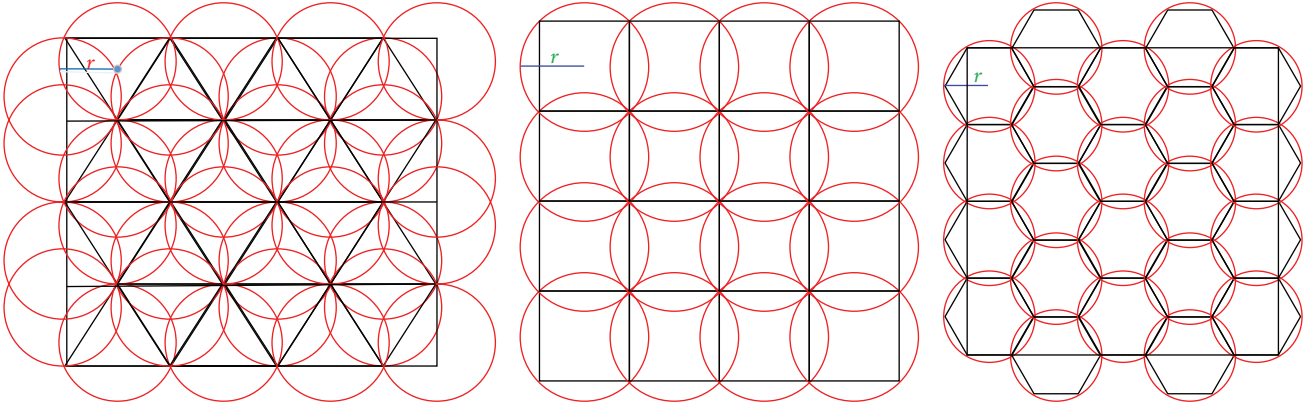


FIGURE 1: Three unit polygon (triangle, square, and hexagon) tessellations.

the MGDC problem and then placing the relay nodes at the centers of the chosen disks. In case the resulting relay nodes are not connected, additional relay nodes can be deployed to comply with the connection requirement among the relay nodes.

This study proposes two simple and effective approximation algorithms for the MGDC problem that identify covering disks using the regular hexagon tessellation of the plane. Tessellation is the process of creating a 2D plane by repeating a geometric shape with no overlaps or gaps. A regular tessellation is a tessellation that covers a 2D plane with regular polygons of the same shape and size. Only three regular tessellations (composed of the hexagon, square, and triangle) exist in the Euclidean plane [1].

A regular polygon enclosed by a unit disk (i.e., a circle with a radius of 1) is a unit polygon. If the polygons in a regular tessellation are all unit polygons, it is called a unit polygon tessellation. The three unit polygon tessellations are as follows: a unit triangle tessellation, unit square tessellation, and unit hexagon tessellation (Figure 1). Let us check how many unit polygons are needed to tessellate a given plane.

Suppose the area of the plane to be covered is F and the radius of the circumscribed circle of each regular polygon is r ; in order to cover the entire plane with regular tessellation, it needs at least $\lceil 4F/3\sqrt{3}r^2 \rceil$ triangles, $\lceil F/2r^2 \rceil$ squares, or $\lceil 2F/3\sqrt{3}r^2 \rceil$ regular hexagons (because the areas of each triangle, square, and hexagon are $(3\sqrt{3}/4)r^2$, $2r^2$, and $(3\sqrt{3}/2)r^2$, resp.). For example, suppose $F = 200 \times 200$ and $r = 10$; then in order to cover the area with regular tessellation, we need at least 308 triangles, 200 squares, or 154 regular hexagons. In general, if F is sufficiently large, then $\lceil 4F/3\sqrt{3}r^2 \rceil > \lceil F/2r^2 \rceil > \lceil 2F/3\sqrt{3}r^2 \rceil$. Therefore, we can use fewer hexagons than squares and triangles to cover the same area with the same circumscribed radius.

This paper is organized as follows: Section 2 provides the related research on the MGDC and WSN relay node placement problems. Section 3 presents the approximation algorithms and their analysis to the MGDC problem using hexagon tessellation. Section 4 provides the simulation results of the proposed approximation algorithms and its

application to the relay node placement problems in WSNs. Section 5 presents the conclusion.

2. Related Research

Given a set P of n points on the Euclidean plane, the MGDC problem is to identify the smallest set of congruent disks with radius r that covers all points in P . The MGDC problem has been shown to be NP-complete [2–4]. Hochbaum and Maass [2] developed a polynomial time approximation scheme (PTAS) for the MGDC problem. They used a divide-and-conquer approach to solve the MGDC problem and proposed the shifting lemma to calculate the approximation ratio. They obtained a family of algorithms with a worst-case approximation ratio of $1 + \epsilon$ (with $0 < \epsilon \leq 3$) and a running time $O(n^{4l^2+1})$, where n is the number of points and l ($l \geq 1$) is the shifting parameter. The time complexity becomes large as ϵ approaches zero, making this approach impractical for even a few input points.

Other polynomial approximation algorithms have been proposed, each of which provides suboptimal solutions within a constant approximation factor of the optimal solution [5–7]. These algorithms are based on the observation that the number of possible disk positions can be limited by assuming (without loss of generality) that any disk that covers at least two points has two of these points on its border. An exhaustive search of all possible disk positions leads to an optimal solution but requires an amount of time that is exponential in the number of points covered. By searching a subset of all possible disk positions, the running time of an algorithm becomes a polynomial of the number of points to cover, and the solution is guaranteed to be within a constant factor of the optimal one.

Franceschetti et al. [5, 8] used a different approach to approximate the MGDC problem. They solved the MGDC problem by locating the covering disks at the centers of a set of tessellated squares. Consider a square grid G . They examined how many disks with given radius r , centered at the vertices of G , were required to completely cover an arbitrary disk of radius r placed on the plane. They showed that this number

is an integer in the set $\{3, 4, 5, 6\}$, the value of which depends on the ratio of r to the grid spacing. If the disks are congruent unit disks (i.e., $r = 1$) that circumscribe the squares of grid G , six disks are required to cover an arbitrary disk of radius r in the worst case. In cooperation with shifting strategy, their approximation algorithm can solve the MGDC problem with no more than $6 + \epsilon$ times the optimal number of disks, where $0 < \epsilon \leq 18$.

In [9], Fu et al. developed an $O(n(\log n)^2(\log \log n)^2)$ deterministic time 2.8334-approximation algorithm for the MGDC problem. The authors used the concept of Borsuk number and developed some novel method to cover the points in the local region, which is roughly occupied by one disk. Instead of covering points with two fixed-size disks, they let two local regions share one disk in some cases. Although the approximation ratio of the algorithm in [9] is smaller than that of our proposed algorithms, their time complexity is higher than ours.

There are some interesting problems related to the MGDC problem studied in the literature, such as the discrete unit disk cover (DUDC) problem. Given a set P of n points and a set D of m unit disks on a 2-dimensional plane, the DUDC problem is (i) to check whether each point in P is covered by at least one disk in D and (ii) if so, then find a minimum cardinality subset $D^* \subseteq D$ such that unit disks in D^* cover all the points in P . In [10], the authors provide an algorithm with constant approximation factor 18. The running time of the proposed algorithm is $O(n \log n + m \log m + mn)$.

Lloyd and Xue [11] studied two versions of relay node placement problems. In the first version, they used a minimal number of relay nodes to create a connecting path between each pair of sensor nodes consisting of relay and sensor nodes. In the second version, they used a minimal number of relay nodes to create a connecting path between each pair of sensor nodes consisting only of relay nodes. They presented a polynomial time 7-approximation algorithm for the first problem and a polynomial time $(5+\epsilon)$ -approximation algorithm for the second problem, where $\epsilon > 0$ and can be any given constant. They claimed the complexity of their algorithms is bounded by $O(n \log n)$ plus the complexity of any efficient approximation algorithms for the minimum Steiner Tree Problem.

Tang et al. [12] studied the relay node placement problem in large-scale WSNs. They placed the fewest relay nodes in a sensor network playing field so that (1) each sensor node could communicate with at least one relay node and (2) the relay node network was connected. They treated relay node placement in WSNs as a variation of the MGDC problem. They developed two polynomial time approximation algorithms to solve the problem and proved that the ratio of relay nodes required by the approximation algorithm to the relay nodes required by the optimal algorithm is bounded by 8 for the first algorithm and 4.5 for the second. However, both algorithms have very high time complexities.

In [13], the authors proposed algorithms for the relay node placement problem in two-tiered WSNs with base stations, which aim to deploy a minimum number of relay nodes to achieve certain coverage and connectivity requirement. Under the assumption that the communication range of

the relay node is the same as sensor nodes, the authors present a $(5 + \epsilon)$ -approximation algorithm for the single cover single connectivity problem with time complexity of $O(n^{1/\epsilon^2} + n^3)$. In their paper, they also presented a $(20 + \epsilon)$ -approximation algorithm for the 2-cover 2-connectivity problem.

The authors in [14] studied the relay node placement problems of WSNs in a set of candidate locations with high energy harvesting potential. They proposed an algorithm that first builds an edge weighted hybrid communication graph (HCG) from the given network and then applies the existing algorithm for the Steiner Tree Problem (STP) to obtain the solution. They obtained a 12.4-approximation algorithm for the general case with a certain number of base stations and 10.85-approximation algorithm for the case without base stations. The complexity of their main algorithm is $O((n + m + l)^2 + T(A))$, where n is the number of sensor nodes, m is the number of base stations, l is the number of the candidate locations, and $T(A)$ is the time complexity of the approximation algorithm for STP.

Recent researches in [15–17] focused on the relay node placement problems in WSNs with extra constraints. In [15], the authors focused on fault tolerant relay node placement in two-tiered heterogeneous WSN with base stations. They proposed an $(18 + \epsilon)$ -approximation algorithm for the relay node placement problem that ensures every sensor node is able to communicate with at least two relay nodes and the induced relay node network is 2-connected. In [16], the authors focused on solving the relay node placement problem with the constraint of minimizing the overall communication cost. They proposed $O(n^2)$ algorithm based on spiral sequence generated for arbitrarily deployed sensor nodes. From their simulation results, the number of deployed relay nodes is very close to the optimal value. In [17], the authors proposed a relay node placement algorithm that the relay nodes are deployed in the center of grids that cover the monitor region and these grids are divided recursively depending on a precalculated threshold value of depth.

In this study, we propose two efficient algorithms for the relay node placement problems with low complexity. Our main result is a linear time $(5 + \epsilon)$ -approximation algorithm for the MGDC problem which can be applied directly to the single cover single connectivity relay node placement problem. For the connection requirement of relay nodes, the paper presents a simple method that first finds the connected components of the relay nodes and then, if necessary, places additional relay nodes among the components to make them connected. This simple method can be done in $O(m^2)$ time, where m is the number of deployed relay nodes generated by the MGDC algorithm. Note that the sensors in WSNs are usually densely deployed; thus the number of relay nodes is far smaller than that of sensor nodes in the networks.

3. Approximation Algorithms for the MGDC Problems with Hexagon Tessellation

To cover a sufficiently large area of a plane, a unit hexagon tessellation requires fewer polygons than a unit square tessellation. Using square tessellation to solve the MGDC problem

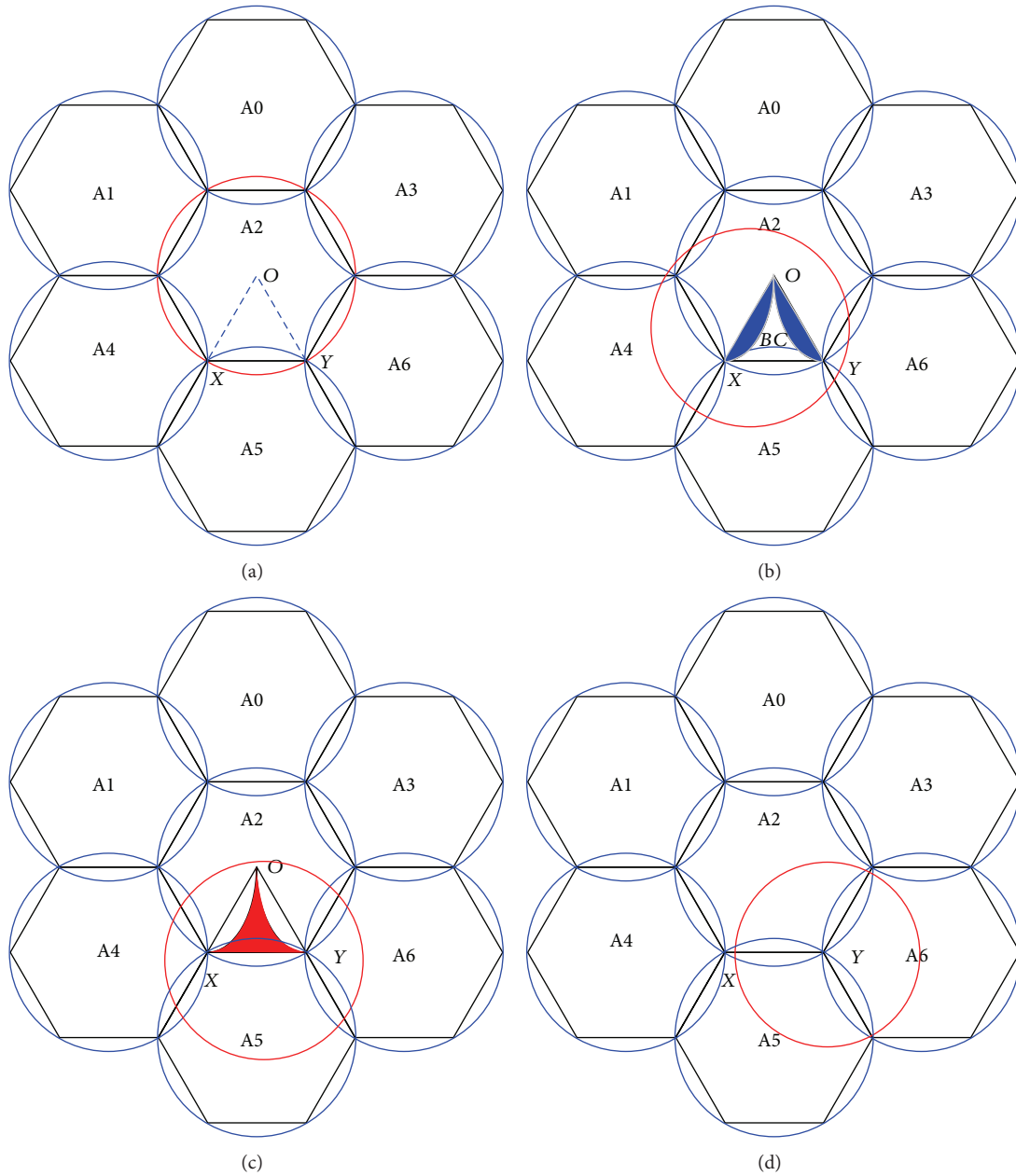


FIGURE 2: (a) The center of the unit disk is located at point O . (b) The center of the unit disk is located at the area of OBX . (c) The center of the unit disk is located in the area of OXY (red region). (d) The center of the unit disk is located at point Y .

has been studied in [8]. As far as we know, no studies on using hexagon tessellation to solve the MGDC problem have been published. This section presents such an approach and proposes an approximation algorithm which uses both the hexagon grid strategy and the shifting strategy.

3.1. Hexagon Grid Strategy. In [8], the authors showed that it is sufficient to cover an arbitrary unit disk in the plane by six grid unit disks in the worst case. This result can be improved if we use unit hexagon instead of unit grid. In this section, we show that it is sufficient to cover any unit disk in the plane by five circumscribed disks of tessellated unit hexagons.

Theorem 1. *A circle circumscribing a unit hexagon is called a hexagon disk. Five hexagon disks are sufficient to cover any unit disk D on a plane.*

Proof. Without loss of generality, assume that the center of unit disk D is located in a regular hexagon (e.g., $A2$ in Figure 2) surrounded by six neighboring regular hexagons. Because a regular hexagon can be divided into six congruent equilateral triangles, different cases were analyzed using one of the triangles (e.g., $\triangle OXY$ in Figure 2(a)). The analysis for the remaining triangles is the same.

Let C be the center of unit disk D . The following cases are discussed according to the location of C .

Case 1 (C is located at point O (Figure 2(a))). The circumscribed circle of hexagon A_2 completely covers unit disk D .

Case 2 (C is located in area OBX (or OCY) (Figure 2(b))). Arc OBX is derived from the unit circle centered at the intersection point of A_1 , A_2 , and A_4 . Arc OCY is derived from the unit circle centered at the intersection point of A_2 , A_3 , and A_6 . Five circumscribed circles of hexagons A_1 , A_2 , A_4 , A_5 , and A_6 (or A_2 , A_3 , A_4 , A_5 , and A_6 if the disk center is in OCY) completely cover unit disk D .

Case 3 (C is located in area OXY (the red region in Figure 2(c))). Four circumscribed circles of hexagons A_2 , A_4 , A_5 , and A_6 completely cover unit disk D .

Case 4 (the center of unit disk D is located at point Y (or X) (Figure 2(d))). Three circumscribed circles of hexagons A_2 , A_5 , and A_6 (or A_2 , A_4 , and A_5) completely cover unit disk D .

It is trivial to see that Cases 1 to 4 include the whole area of triangle OXY . Because the most circumscribed circles (i.e., Case 2) required to cover unit disk D are five, we conclude that it is sufficient to cover any unit disk on the plane with five hexagon disks. \square

Similar proof techniques can be applied to the other two regular tessellations (square and triangle). The proofs for the following theorems can be found in the Appendix. Similar to the definition in Theorem 1, a circle circumscribing a unit square is called a square disk and a circle circumscribing a unit triangle is called a triangle disk. Theorem 2 is due to a special case of Theorem 1 in [5].

Theorem 2. *Given a square tessellation, six square disks are sufficient to cover any unit disk D on a plane.*

Theorem 3. *Given a triangle tessellation, ten triangle disks are sufficient to cover any unit disk D on a plane.*

3.2. The shifting Strategy. The shifting strategy is a divide-and-conquer technique used to confine the error by applying it repetitively, choosing the single most favorable solution. Let P be a set of n given points on the plane enclosed in an area R . The goal is to cover these points with a minimum number of disks of prescribed radius r .

Suppose we have a local algorithm A for the given problem. The shifting algorithm $S(A)$ works as follows. We first divide the plane into vertical strips of width $D \geq 2r$. Let the shifting parameter be l . Consider the groups of l consecutive strips. Each group is itself a thicker strip of width $l \times D$. Let A be an algorithm that can deliver a solution within each group. We can find a feasible solution by applying algorithm A to each of the groups and then considering the union of all disks used. We then repeat the same strategy after shifting all the groups by the length D . Since each group is a strip $l \times D$ wide, we can repeat the shift a total of $l - 1$ times and then select the feasible solution of minimum cardinality as the final best covering.

Let Z_A be the value of the solution delivered by algorithm A . OPT denotes an optimal solution set and its size is $|OPT|$.

Let r_A be the performance ratio of the algorithm A . r_A is defined as the supremum of the ratio $Z_A/|OPT|$, over all problem instances. The following lemma is proven in [2].

Lemma 4 (the shifting lemma [2]). *Consider*

$$r_{S(A)} \leq r_A \left(1 + \frac{1}{l}\right). \quad (1)$$

We can apply the shifting strategy twice on the plane. We first cut the plane into vertical strips of width $l \times D$. Then, in order to cover the points in each such strip, we apply the shifting strategy in the other dimension. Thus, we cut the considered strip into regions of side length $l \times D$. Let B be the local algorithm that delivers a solution in each square region. By applying the shifting lemma twice, we have

$$r_{S(B)} \leq r_B \left(1 + \frac{1}{l}\right)^2. \quad (2)$$

3.3. The Proposed Approximation Algorithm. In this subsection, we present a linear time approximation algorithm (Algorithm 1) for the MGDC problem based on the hexagon grid strategy and the shifting strategy. The width of vertical strip D_1 is set to $3r$ and the height of horizontal strip D_2 is set to $2\sqrt{3}r$. Figure 3 shows an area divided by the square-like regions with size $D_1 \times D_2$.

Analysis of Algorithm Hexagon-1. The major step of the algorithm is Step 11, which is responsible for checking whether the current k hexagon disks can cover all of the points in the square-like region Q in question. Step 11 can be done in $O(n)$ steps by the following substeps.

Step 11(a). Identify the set of points $S \subseteq P$ which are located in region Q .

Step 11(b). Check whether each point u in S is located in at least one hexagon disk of the k hexagon disks by calculating the distances between u and the centers of the k hexagon disks.

Step 11(a) takes $O(n)$ steps to check each point in P . Step 11(b) takes $O(|S| \times k)$ steps. Since $|S|$ is bounded by n , Step 11(b) also takes $O(n)$ steps. Therefore, Step 11 takes $O(n)$ steps in total.

Now, let us compute the frequency of execution of Step 11 in the algorithm. In Step 11, all the combinations of the k hexagon disks are needed to be examined in the worst case, where k is from 0 to $4l^2$. Thus, the total number of examinations is $C(4l^2, 0) + C(4l^2, 1) + \dots + C(4l^2, 4l^2) = 2^{4l^2}$. Let the area of region R be M . Because the area of each hexagon is $3\sqrt{3}r^2/2$, there are $\lceil M/(4l^2 \times 3\sqrt{3}r^2/2) \rceil = \lceil M/6\sqrt{3}r^2l^2 \rceil$ square-like regions in R . Therefore, Step 11 needs to be performed $\lceil M/6\sqrt{3}r^2l^2 \rceil \times 2^{4l^2}$ times in the worst case for a shift. Since there are $(l - 1)^2$ shifts in the algorithm, the number of examinations in Step 11, in the worst case, is $\lceil M/6\sqrt{3}r^2l^2 \rceil \times 2^{4l^2} \times (l - 1)^2$, which is less than $\lceil M/6\sqrt{3}r^2 \rceil \times 2^{4l^2}$. Thus, the complexity of this algorithm is $O(Kn)$, where $K < \lceil M/6\sqrt{3}r^2 \rceil \times 2^{4l^2}$. If the area R is fixed, M is a constant.

Algorithm Hexagon-I**Input:** A set P of n points, and a two-dimensional region R .**Output:** A set of disks with radius r that covers P .

- (1) Let the width of the vertical strips be $l \times D_1$ and the height of the horizontal strips be $l \times D_2$, where $D_1 = 3r, D_2 = 2\sqrt{3}r$, and l is the shifting parameter (a positive integer).
- (2) Min-Cardinality = $+\infty$; Min-Set = $\{\}$ // Min-Set contains the minimum number // of hexagon disks that cover P found so far
- (3) For $i = 0$ to $l - 1$ do {
- (4) Set the starting position of vertical strips at $i \times D_1$
- (5) For $j = 0$ to $l - 1$ do {
- (6) Set the starting position of horizontal strips at $j \times D_2$
- (7) Let $H = \{\}$ // H contains the chosen hexagon disks that cover points in P
- (8) For each $(l \cdot D_1 \times l \cdot D_2)$ square-like region Q in R do
- (9) For $k = 0$ to $4 \times l^2$ do { // 0 is for the case of no points in Q
- (10) For each combination of k hexagon disks in Q do
- (11) Check whether the k hexagon disks cover all the points in Q
- (12) If yes then add the k hexagon disks to H and break the k -loop
- (13) }
- (14) If $|H| < \text{Min-Cardinality}$ then { Min-Set = H ; Min-Cardinality = $|H|$ }
- (15) }
- (16) }
- (17) Output Min-Set

ALGORITHM 1: The proposed approximation algorithm for the MGDC problem.

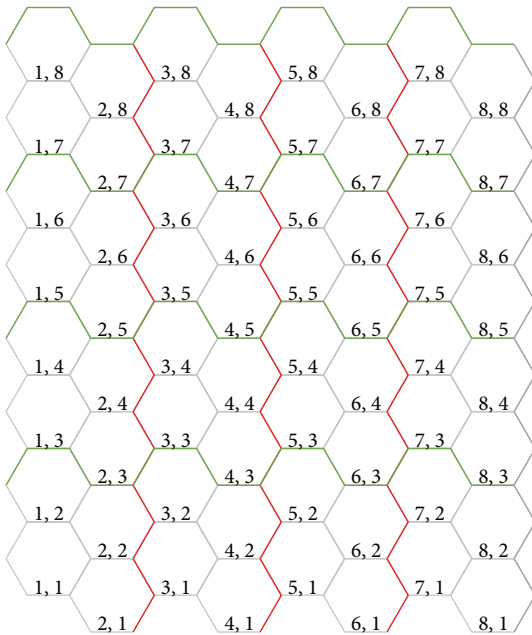


FIGURE 3: Square-like regions obtained by the vertical strips (divided by red lines) and the horizontal strips (divided by green lines).

Since both r and l are constants, K is a constant when the area R is fixed. Thus, the algorithm Hexagon-I in Figure 2 is linear, if the area of sensing region is fixed. (Note. If we assume that the region R is a bounded area, the input “a 2-dimensional region R ” in algorithm Hexagon-I should be removed and make a claim that the region R is a 2-dimensional bounded area in the body of the algorithm.)

Theorem 5. *There is a linear time approximation algorithm such that, for every given natural number $l \geq 1$, the algorithm delivers a cover of the n given points by disks of prescribed radius r in $O(Kn)$ steps with approximation factor $\leq 5(1+1/l)^2$, where K is a constant when the area of sensing region is fixed.*

Proof. Algorithm 1 is such a linear time algorithm that delivers a cover for the n given points in an area. The approximation ratio $\leq 5(1+1/l)^2$ is directly from Theorem 1 and the shifting lemma, and K is a constant bounded by $\lceil M/6\sqrt{3}r^2 \rceil \times 2^{4l^2}$ from the above analysis. \square

By Theorem 1 and Lemma 4, the performance ratio of Hexagon-I is less than or equal to $5 \times (1+1/l)^2$, where l is a natural number. Thus, the performance ratio of Hexagon-I is between $5 \times (1+1/\infty)^2 \cong 5$ and $5 \times (1+1/1)^2 = 20$. Therefore, the approximation ratio of algorithm Hexagon-I is $(5 + \epsilon)$, where $0 < \epsilon \leq 15$.

3.4. A Fast Approximation Algorithm. Theorem 1 shows that it is sufficient to cover the unit disk D with five hexagon disks in the worst case. Since there are six hexagons surrounding the hexagon which contains the center of unit disk D , it needs some computation to decide which hexagon disks can be used to cover the unit disk D . However, if we relax the constraint of having the least covering hexagon disks, we can always cover the unit disk D with at most seven hexagon disks (i.e., disks A0 to A6). Thus, we have the following corollary.

Corollary 6. *It is sufficient to cover any unit disk D with seven hexagon disks which surround disk D .*

Algorithm Hexagon-II**Input:** A set P of n points, and a two-dimensional region R .**Output:** A set of disks with radius r that covers P .**Step 1:**

Virtually tessellate the region R with hexagons of length r . Calculate and store the coordinate of the center and the corresponding index of each hexagon in the tessellation. (Each index contains two ranks (x_rank , y_rank) to x -axis and y -axis, respectively, as shown in Figure 5.)

Step 2:

For each point p with Cartesian coordinate (x, y) in P do

(1) $x_rank = \left\lfloor \frac{x}{3r/2} \right\rfloor$; // find the indices of the basic rectangle that contains p

If x_rank is even

$y_rank = \left\lfloor \frac{y}{\sqrt{3}r} \right\rfloor$

else

$y_rank = \left\lfloor \frac{y + (\sqrt{3}/2)r}{\sqrt{3}r} \right\rfloor$

(2) Find the hexagon containing p by comparing the distances from p to the centers of hexagons with indices (x_rank, y_rank) , $(x_rank + 1, y_rank)$, and $[(x_rank + 1, y_rank + 1)$ if x_rank is even or $(x_rank + 1, y_rank - 1)$ if x_rank is odd]. Ties are broken in favor of the marked hexagon, otherwise broken arbitrary.

(3) Mark the hexagon chosen in (2).

Step 3:

Output the circumscribed circles of the chosen hexagons.

ALGORITHM 2: A simple linear time approximation algorithm for the MGDC problem.

Inspired by the result of Corollary 6, we design a 7-approximation algorithm, as shown in Algorithm 2, for the MGDC problem. In the algorithm Hexagon-II, we simply check every point p in P to see in which hexagon the point p is located and then mark the hexagon. This algorithm is simple yet very efficient. The complexity of this algorithm is $O(n + |R|)$, where n is the number of points in P and $|R|$ represents the area of the tessellated region R .

Before we go into the detail of the 7-approximation algorithm Hexagon-II, let us introduce the technique of locating a point in the hexagonally tessellated plane in constant time.

Index each hexagon with length r in the hexagonally tessellated plane as in Figure 4. Suppose we divide the plane by rectangles with width $= (3/2)r$ and length $= \sqrt{3}r$ such that rectangles in even column are stacked from the x -axis and rectangles in odd column are stacked from the horizontal line which is $(\sqrt{3}/2)r$ above the x -axis; we can index each rectangle the same way as we index the hexagons in the tessellated plane.

It is much easier to locate a point in a rectangle than to locate a point in a hexagon directly. Therefore, in order to locate a point p in a hexagon of the tessellated plane, we first locate p in a basic rectangle. As shown in Figure 4, each basic rectangle contains three regions from three neighboring hexagons. Once we know the point p is located in some basic rectangle, we can determine in which hexagon the point p is located by comparing the distances from p to the centers of the three neighboring hexagons around the basic rectangle. The detail of the algorithm is shown in Algorithm 2 and the analysis of the time complexity is as follows.

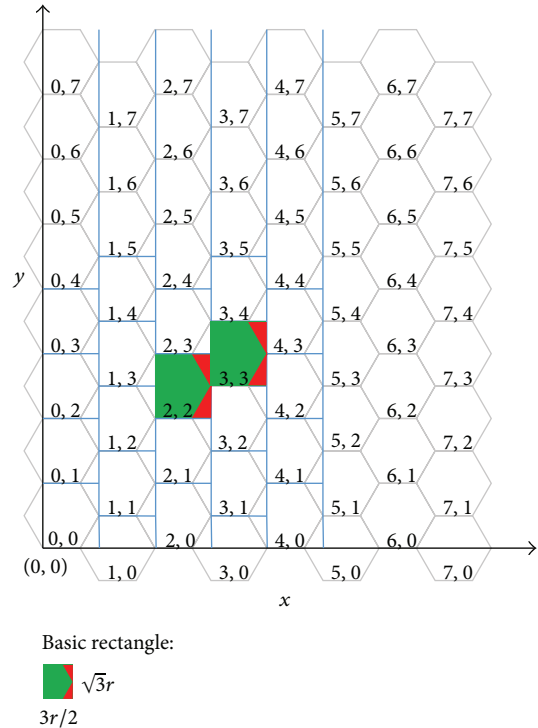


FIGURE 4: The hexagonal tessellation plane is indexed and divided by rectangles.

Because the number of hexagons in the tessellation relies on the area of the region R , Step 1 of the algorithm Hexagon-II requires $O(|R|)$ time to calculate and store the hexagon

centers and their indices. Because there are n points in P and it takes constant time to locate a point in a basic rectangle and check which hexagon the point is in, Step 2 requires $O(n)$ time. The complexity of Step 3 also relies on the number of hexagons, which is $O(|R|)$. Therefore, the overall complexity of the algorithm is $O(|R|) + O(n) + O(|R|)$, which is $O(n + |R|)$. If the area of region R is fixed, then the complexity of the algorithm is $O(n)$.

4. Simulation Results of the MGDC Problems and the Applications to the Relay Node Placement Problems

In this section, we first demonstrate how to derive the optimal solution to the MGDC problem using binary integer programming and solve the problem with MATLAB. We then simulate our proposed algorithms (Hexagon-I and Hexagon-II) and compare the results with the optimal solutions. The approximation algorithms of triangle and square tessellation are also simulated for comparison.

4.1. The Optimal Algorithm to the MGDC Problems Using Binary Integer Programming. In [18], Wong et al. use binary integer programming to calculate the optimal gateway placement under the assumption of known node positions and communication range. In the paper, the possible positions to place the gateways are found by identifying competitive regions (C Regions). A region here is defined as the overlapping communication ranges of a set of nodes. Thus if a gateway is placed in this region, it can communicate with these nodes in one hop. Therefore, solving the optimal MGDC problem is reduced to finding the minimum number of chosen C Regions that can cover all the nodes.

Suppose that n points in P are distributed on an $m \times m$ square region and the radius of the covering disk is r . The optimal solution to the MGDC problem can be calculated using the following binary integer programming canonical form:

$$\begin{aligned} & \text{Minimize} && I_1^T X \\ & \text{Subject to} && AX \geq I_2, \end{aligned} \quad (3)$$

where I_1^T is the transpose of a $C \times 1$ identity vector, in which C is the total number of C Regions, I_2 is an $n \times 1$ identity vector, X is a $C \times 1$ binary vector, where each entry denotes whether the corresponding C Region is chosen as an area to place the covering disk (1 for yes and 0 for no), and A is an $n \times C$ matrix where an element (i, j) is set to 1 if the distance between point i and the C Region j is less than or equal to r ; otherwise it is set to 0.

MATLAB binary integer programming function `bintprog()` was used to compute the optimal results.

4.2. Simulation Results. The simulation environment was set up as follows:

- (1) a square region of 200×200 ,
- (2) covering disk radius of 10,

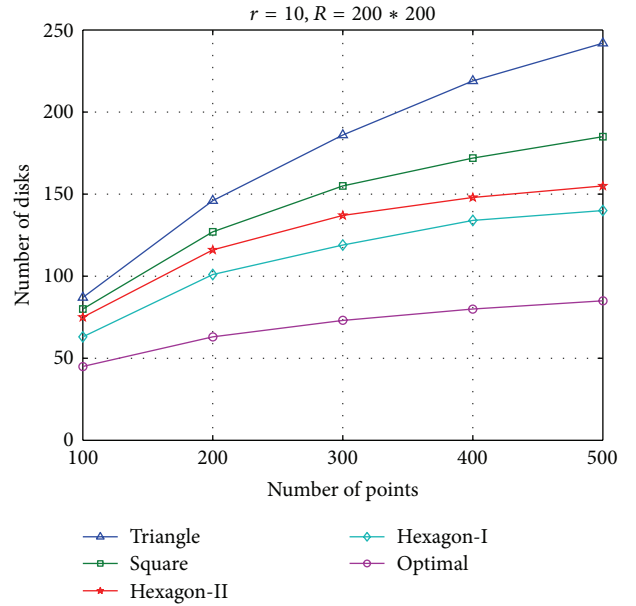


FIGURE 5: Performance comparison of the optimal solutions to the solutions of the proposed algorithms with triangle tessellation, square tessellation, and hexagon tessellation. (Hexagon-I and Hexagon-II use the algorithms in Figure 2 with $l = 2$ and Figure 3, resp.).

- (3) uniformly distributed points,
- (4) number of points ranging from 100 to 500,
- (5) each case running 50 tests and using the average of their values,
- (6) using an Acer PC with Intel core-i7 CPU and 4 Gb memory.

Figure 5 shows the simulation results. As expected, the hexagon tessellation performs better than the square and triangle tessellations. More importantly, the numbers of disks generated by algorithms Hexagon-I (with $l = 2$) and Hexagon-II are on average less than 1.7 and 2 times the optimal solution, respectively, for all the cases from 100 to 500 points. This indicates that the proposed approximation algorithms for the MGDC problem are practical for general use.

The execution times of Hexagon-I (with $l = 2$) and Hexagon-II are listed in Table 1. As shown in Table 1, the execution time of algorithm Hexagon-II is much faster than algorithm Hexagon-I. This is due to the fact that algorithm Hexagon-II only checks the input points once in the sensing area, whereas algorithm Hexagon-I needs to examine the combinations of hexagons in each square-like region.

4.3. Applications to the Relay Node Placement Problems in WSNs. The proposed approximation algorithms for the MGDC problem can be applied to the relay node placement problems in WSNs. The relay nodes are deployed at the centers of the chosen disks so that each sensor is covered by one relay node. In general, the relay nodes are responsible

TABLE 1: Comparison of the execution time (in seconds) of the two proposed algorithms.

Algorithm/number of points	100	200	300	400	500
Hexagon-I	396.87	3494.42	6314.50	9755.30	12923.42
Hexagon-II	0.37	0.40	0.43	0.45	0.47

TABLE 2: The average number of connected components and additional relay nodes needed for connectivity under various numbers of sensor nodes in a 200×200 monitor region. ($r = 10$, $R = 40$).

Number of sensor nodes	50	60	70	80	90	100	200	300	400	500
Number of relay nodes deployed using the Hexagon-II MGDC solution	42	50	56	63	68	75	114	137	148	155
Average number of connected components of relay nodes	4.9	3.9	3.1	1.9	1.5	1	1	1	1	1
Number of additional relay nodes needed for connectivity	10	4	2	1	1	0	0	0	0	0

for aggregating data from their dominating sensors and then forwarding the aggregated data to the base station by multihopping. To fulfill this task, the relay nodes themselves need to be connected. In this study, we assume that the relay nodes are homogeneous and with the same communication range R . R is normally much larger than the communication range r of sensors, and thus we adopt the assumption in [12] that $R \geq 4r$.

To establish the connectivity among relay nodes, we adopt the same approach taken in [19]. We model the relay nodes as a graph $G = (V, E)$, in which the vertex set V consists of all the relay nodes and, for any $u, v \in V$, there is an edge (u, v) in E if u is in the communication range of v . Table 2 lists the average number of connected components of 50 tests for various numbers of relay nodes deployed according to the solution of our proposed approximation algorithm Hexagon-II to the MGDC problem in a 200×200 monitored region.

As shown in Table 2, the number of components is equal to 1 while the amount of sensors is greater than 100, which means that the relay nodes are connected already. Since the density of sensor nodes in a WSN is normally high, it is very common that the number of sensors is greater than 100 in a 200×200 region. In case there is more than one connected component in G , we can use the following simple algorithm to add additional relay nodes to the network. The time complexity of the algorithm is $O(m^2)$, where m is the number of relay nodes.

Step 1. Find all the connected components in G .

Step 2. Calculate the distances between every two connected components. The distance of two connected components C_1 and C_2 is defined as the shortest distance between nodes in C_1 and nodes in C_2 .

Step 3. Denote the super graph formed by the connected components as H . Find a minimum spanning tree T of H .

Step 4. Add m relay nodes equidistantly to the edges in T whose lengths are greater than R , where m is equal to $\lceil \text{edge length}/R \rceil - 1$. Thus, the resulting graph is connected.

5. Conclusion and Discussions

This study proposes a linear time approximation algorithm for the MGDC problem, which can be used to solve the WSN relay node placement problems. The number of disks generated by the proposed algorithm is not more than $5 + \epsilon$ times the optimal solution, where $0 < \epsilon \leq 15$. Although the authors in [13] also provided a $(5 + \epsilon)$ -approximation algorithm for the relay node placement problems, in contrast to our linear time algorithm, the time complexity of their algorithm, $O(n^{1/\epsilon^2} + n^3)$, is quite large.

For the case where the locations of relay nodes need to be determined promptly, the study provides a fast 7-approximation algorithm for the WSN relay node placement problem having time complexity $O(n + |R|)$ with very low constant factor, where n is the number of points and $|R|$ is the area of the deployed region. If the area of the deployed region is fixed, then the time complexity of the algorithm is also linear.

Simulation showed that the number of disks generated by both of the proposed algorithms was on average less than twice the optimal solution. This allows the proposed algorithms to be practical for general use.

If we model the WSNs as a graph, the relay node placement problem in WSNs can be solved by applying the proposed MGDC approximation algorithms to find the disks that cover all the sensors and then place the relay nodes at the centers of the disks. Since the sensors of WSNs are usually densely deployed, the relay nodes placed by the result of the proposed algorithms are usually connected. In case that the relay nodes generated by the proposed algorithm are not connected, this study provides a simple algorithm that deploys additional relay nodes to comply with the connection requirement of relay nodes in a WSN.

In this study, we solve the relay node placement problem by first solving the MGDC problem and then placing additional relay nodes, if necessary, to preserve the connection requirement of the relay nodes. The other approach to solve the relay node placement problem is to consider the connection requirement of relay nodes while solving the MGDC problem. It is highly likely that this approach can

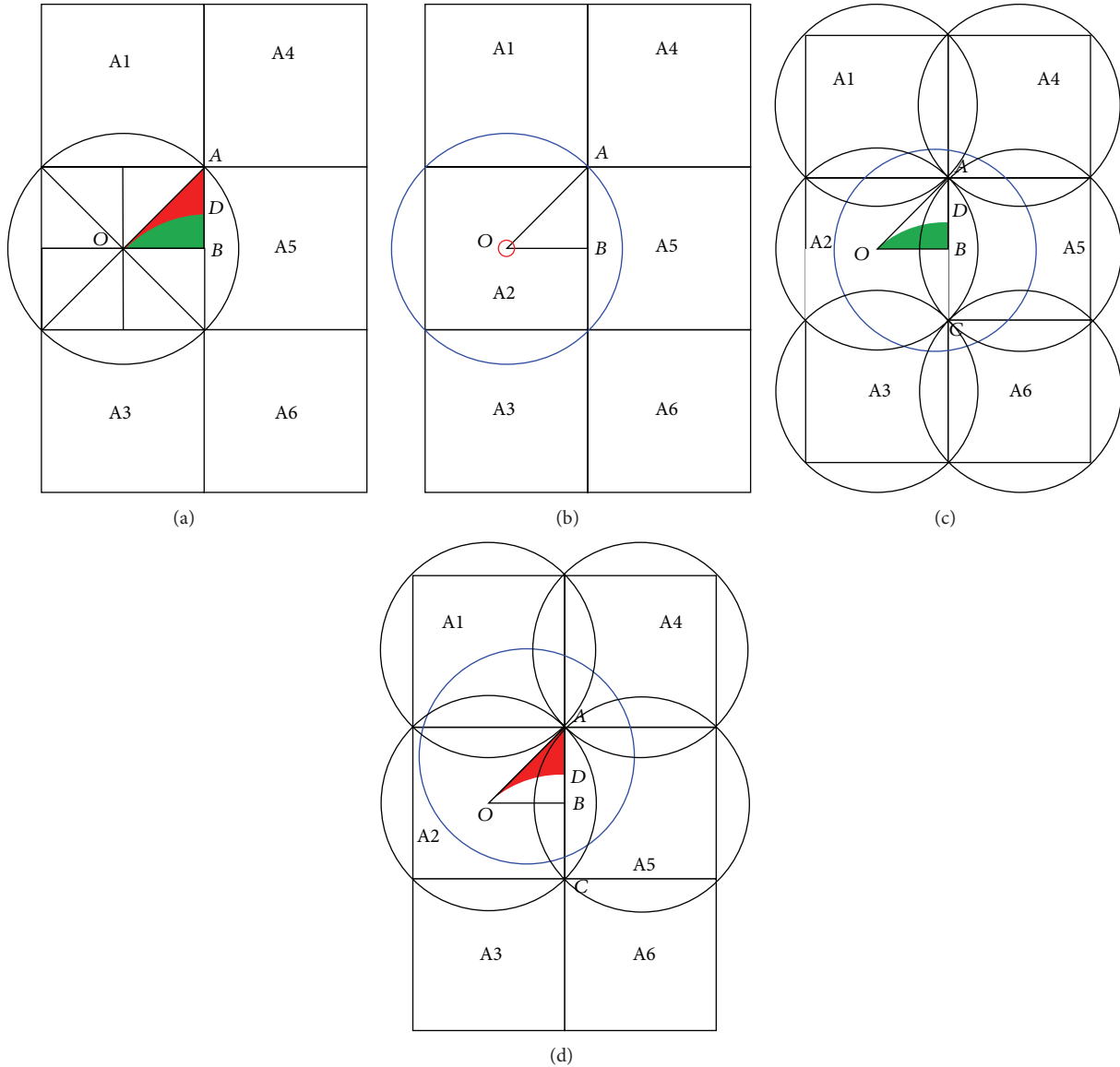


FIGURE 6: (a) A regular square can be divided into 8 congruent right triangles. (b) The center of the unit disk is located at point O . (c) The center of the unit disk is located in the area of OBD (green region). (d) The center of the unit disk is located in the area of ODA (red region).

reduce the number of relay nodes needed to cover the sensor nodes. However, it may increase the time complexity of solving the MGDC problem.

Appendix

Theorem A.1. *Given a square tessellation, six square disks are sufficient to cover any unit disk D on a plane.*

Proof. A circle circumscribing a unit square is called a *square disk*. We show that it is sufficient to cover any unit disk δ on a plane with 6 square disks.

Without loss of generality, assume that the center of unit disk U is located in a regular square $A2$ in Figure 6. Since a regular square can be divided into 8 congruent right triangles

(Figure 6(a)), we analyze different cases using one of the right triangles (e.g., OBA in Figure 6(a)). The analysis of the remaining triangles is the same.

Let ϕ be the center of unit disk δ . The following cases are discussed according to the location of C .

Case 1 (ϕ is located at point O , where O is the center of square $A2$. (Figure 6(b))). The unit disk δ can be completely covered by the circumscribed circle of square $A2$.

Case 2 (ϕ is located in area OBD (the green region in Figure 6(c))). The arc OD is derived from the unit circle centered at the intersection point C of $A2$, $A3$, $A5$, and $A6$. The unit disk δ can be completely covered by 6 circumscribed circles of squares $A1$, $A2$, $A3$, $A4$, $A5$, and $A6$.

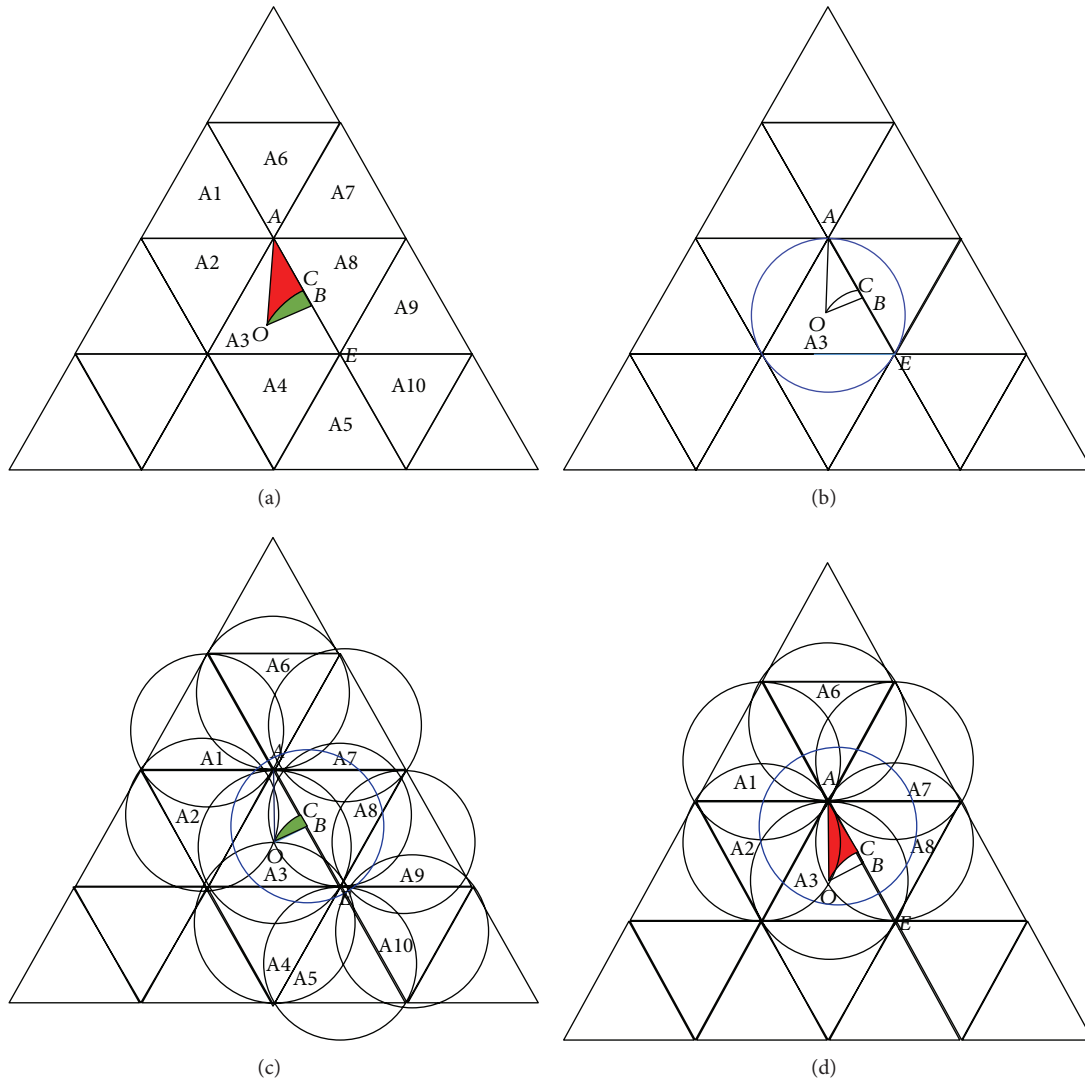


FIGURE 7: (a) A regular triangle can be divided into 6 congruent right triangles. $\triangle OBA$ is one of the right triangles, where O is the center of A_3 . (b) The center of the unit disk is located at point O . (c) The center of the unit disk is located in the area of OBC (green region). (d) The center of the unit disk is located in the area of OCA (red region).

Case 3 (ϕ is located in area ODA (the red region in Figure 6(d))). The unit disk δ can be completely covered by 4 circumscribed circles of squares $A_1, A_2, A_4,$ and A_5 .

It is trivial to see that Cases 1 to 3 include the whole area of triangle OBA . Since the most circumscribed circles (i.e., Case 2) required to cover unit disk δ are 6, we conclude that it is sufficient to cover any unit disk on a plane with 6 square disks. Therefore, suppose the optimal number of unit disks to cover P is k . Since each unit disk can be covered by at most 6 square disks, not more than $6 \times k$ squares disks are required to cover P . Thus, using the proposed algorithm, the number of output unit disks is not more than 6 times the optimal number of unit disks. \square

Theorem A.2. Given a triangle tessellation, ten triangle disks are sufficient to cover any unit disk D on a plane.

Proof. A circle circumscribing a unit triangle is called a *triangle disk*. We show that it is sufficient to cover any unit disk δ on a plane with 10 triangle disks.

Without loss of generality, assume that the center of unit disk δ is located in the regular triangle A_3 in Figure 7(a). Since a regular triangle can be divided into 6 congruent right triangles, we only analyze one of the triangles (e.g., $\triangle OBA$ in Figure 7(a)). The analysis of the remaining triangles is the same.

Let ϕ be the center of unit disk δ . The following cases are discussed according to the location of C .

Case 1 (ϕ is located at point O , which is the center of regular triangle A_3 (Figure 7(b))). The unit disk δ can be completely covered by the circumscribed circle of triangle A_3 .

Case 2 (ϕ is located in area OBC (the green region in Figure 7(c))). The arc OC is derived from the unit circle centered at the intersection point E of triangles A_3 , A_4 , A_5 , A_8 , A_9 , and A_{10} . The unit disk δ can be completely covered by 10 circumscribed circles of triangles A_1 to A_{10} .

Case 3 (ϕ is located in area OCA (the red region in Figure 7(d))). In this case, the unit disk δ can be completely covered by 6 circumscribed circles of triangles A_1 , A_2 , A_3 , A_6 , A_7 , and A_8 .

It is trivial to see that Cases 1 to 3 include the whole area of triangle OBA . Since the most circumscribed circles (i.e., Case 2) needed to cover unit disk δ are 10, we conclude that it is sufficient to cover any unit disk on the plane with 10 triangle disks. Therefore, suppose the optimal number of unit disks to cover P is k . Since each unit disk can be covered by at most 10 triangle disks, not more than $10 \times k$ triangles disks are required to cover P . Thus, with the proposed algorithm, the number of output unit disks is not more than 10 times the optimal number of unit disks. \square

Disclosure

Po-Ying Chen used to work for Department of Information Engineering, I-Shou University. However, he moved to National Chin-Yi University of Technology in August 2014, during the time in which this paper was under review.

Conflict of Interests

The authors declare no conflict of interests.

Acknowledgments

This work was partially supported by the Ministry of Science and Technology of Taiwan under Contract MOST103-2221-E-214-033. The authors would like to thank all the anonymous reviewers for their precious suggestions and Professor Yukon Chang for the improvement of English writing of this paper.

References

- [1] W. W. R. Ball, *Mathematical Recreations and Essays*, Dover Publications, New York, NY, USA, 13th edition, 2010.
- [2] D. S. Hochbaum and W. Maass, "Approximation schemes for covering and packing problems in image processing and VLSI," *Journal of the Association for Computing Machinery*, vol. 32, no. 1, pp. 130–136, 1985.
- [3] D. S. Johnson, "The NP-completeness column: an ongoing guide," *Journal of Algorithms*, vol. 3, no. 2, pp. 182–195, 1982.
- [4] R. J. Fowler, M. S. Paterson, and S. L. Tanimoto, "Optimal packing and covering in the plane are NP-complete," *Information Processing Letters*, vol. 12, no. 3, pp. 133–137, 1981.
- [5] M. Franceschetti, M. Cook, and J. Bruck, "A geometric theorem for approximate disc covering algorithms," Electronic Technical Report ETR035, 2001, <http://resolver.caltech.edu/CaltechPARADISE:2001.ETR035>.
- [6] H. Brönnimann and M. T. Goodrich, "Almost optimal set covers in finite VC-dimension," *Discrete & Computational Geometry*, vol. 14, no. 4, pp. 463–479, 1995.
- [7] T. F. Gonzalez, "Covering a set of points in multidimensional space," *Information Processing Letters*, vol. 40, no. 4, pp. 181–188, 1991.
- [8] M. Franceschetti, M. Cook, and J. Bruck, "A geometric theorem for network design," *IEEE Transactions on Computers*, vol. 53, no. 4, pp. 483–489, 2004.
- [9] B. Fu, Z. Chen, and M. Abdelguerfi, "An almost linear time 2.8334-approximation algorithm for the disc covering problem," in *Algorithmic Aspects in Information and Management*, vol. 4508 of *Lecture Notes in Computer Science*, pp. 317–326, Springer, Berlin, Germany, 2007.
- [10] G. K. Das, R. Fraser, A. López-Ortiz, and B. G. Nickerson, "On the discrete unit disk cover problem," in *WALCOM: Algorithms and Computation: 5th International Workshop, WALCOM 2011, New Delhi, India, February 18–20, 2011. Proceedings*, N. Katoh and A. Kumar, Eds., vol. 6552 of *Lecture Notes in Computer Science*, pp. 146–157, Springer, Berlin, Germany, 2011.
- [11] E. L. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *IEEE Transactions on Computers*, vol. 56, no. 1, pp. 134–138, 2007.
- [12] J. Tang, B. Hao, and A. Sen, "Relay node placement in large scale wireless sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 490–501, 2006.
- [13] G. Chen and S. Cui, "Relay node placement in two-tiered wireless sensor networks with base stations," *Journal of Combinatorial Optimization*, vol. 26, no. 3, pp. 499–508, 2013.
- [14] S. Misra, N. E. Majd, and H. Huang, "Approximation algorithms for constrained relay node placement in energy harvesting wireless sensor networks," *IEEE Transactions on Computers*, vol. 63, no. 12, pp. 2933–2947, 2014.
- [15] Z. Gao, K. Chen, and X. Qiu, "Relay node placement with base stations in wireless sensor networks fault-tolerant," *Chinese Journal of Electronics*, vol. 23, no. 4, pp. 794–800, 2014.
- [16] N. Kumar and P. K. Jana, "Relay node placement algorithm in wireless sensor network," in *Proceedings of the 4th IEEE International Advance Computing Conference (IACC '14)*, pp. 220–225, February 2014.
- [17] H. Barati, M. Sedighi, A. Movaghar, and I. Attarzadeh, "Proposing a novel algorithm for fault-tolerant relay node placement in wireless sensor networks," *Journal of Advances in Computer Research*, vol. 5, no. 1, pp. 53–60, 2014.
- [18] J. L. Wong, R. Jafari, and M. Potkonjak, "Gateway placement for latency and energy efficient data aggregation," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, pp. 490–497, November 2004.
- [19] C.-Y. Chang, C.-C. Chen, and C.-C. Liu, "A novel approximation algorithm for minimum geometric disk cover problem with hexagon tessellation," in *Advances in Intelligent Systems and Applications—Volume 1: Proceedings of the International Computer Symposium ICS 2012 Held at Hualien, Taiwan, December 12–14, 2012*, R.-S. Chang, L. C. Jain, and S.-L. Peng, Eds., pp. 157–166, Springer, Heidelberg, Germany, 2013.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

