MULTIPLE SINK POSITIONING IN SENSOR NETWORKS

Shahab Mihandoust

A thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montréal, Québec, Canada

January 2008

Your file Votre référence
ISBN: 978-0-494-40948-0
Our file Notre référence
ISBN: 978-0-494-40948-0

Abstract

Multiple Sink Positioning in Sensor Networks

Shahab Mihandoust

We study the problem of positioning multiple sinks, or data collection stops for a mobile sink, in a sensor network field. Given a sensor network represented by a unit disc graph $G = (V, E)$, we say a set of points U (sink node locations) is an h-hop covering set for G if every node in G is at most h hops away from some point in U. Placing sink nodes at the points of a covering set guarantees that every sensor node has a short path to some sink node. This can increase network lifetime, reduce the occurrence of errors, and reduce latency. We also study variations of the problem where the sink locations are restricted to be at points of a regular lattice (lattice-based covering set), or at network nodes (graph-based covering set).

We give the first *polynomial time approximation scheme (PTAS)* for the h-hop covering set problem, the h-hop lattice-based covering set problem, and the h-hop graph-based covering set problem. We give a new PTAS for the lattice-based disc cover problem, based on a new approach deriving from recent results on dominating sets in unit disc graphs. We show that this gives a $(3+\epsilon)$-approximation algorithm for the disc cover problem, and gives the first distributed algorithm for this problem. We give a $(5+\epsilon)$-approximation algorithm for the h-hop covering set problem in unit disc graphs, that does not require a geometric representation of the graph. Finally, we give a $(3+\epsilon)$-approximation algorithm for the h-hop covering set problem for unit disc graphs that runs in time quadratic in the number of nodes

in the graph, for any constant ϵ and h. In addition to showing how well a lattice-based approach for a disc cover problem approximates the optimal solution, we prove a geometric theorem that gives an exact relationship between the side of a triangular lattice and the number of lattice discs that are necessary and sufficient to cover an arbitrary disc on the plane.

Acknowledgments

"The important thing is not to stop questioning."

"Do not worry about your difficulties in Mathematics. I can assure you mine are still greater."

"The hardest thing in the world to understand is the income tax."

Albert Einstein (1879 - 1955)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless sensor networks have been an active area of research for the last several years since the desire for wireless connectivity has grown drastically. According to MIT's *Technology Review* magazine, sensor networks are "one of the ten technologies that will change the word". The important recent technological developments which influence the advancement of wireless sensor networks, and make research on these networks more important, are firstly the growth in semiconductor technology, which causes smaller and cheaper computing units, secondly the miniaturization of energy capacity, and fast recharging capabilities, and finally the *system-on-a-chip* (SoC) integration technology that enables micro sensors, on board processors, and wireless interfaces to be integrated at a very small scale.

The applications envisioned for wireless sensor networks range from industrial control and monitoring, home automation and consumer electronics, security and military sensing, to health monitoring. An example of wireless industrial control could be the control of commercial lighting, where a flexible wireless system can deliver a strong control on turning on and off the lights in industrial units or controls the dimming of them. Another example of industrial monitoring is the heating, ventilating, and air conditioning (HVAC) of buildings.

A wireless HVAC system, using wireless thermostats and humidistats (sensors), monitors the heat load generated by people in a building (which is a very dynamic parameter), and provides a balanced heating and air conditioning in the area. A home version of a HVAC system can also be an example of a home automation application as well as using wireless sensor networks for other controlling tasks such as light and sound. Locating and identifying targets for a potential attack is an example of a military application. In addition, athletic performance monitoring and at-home health monitoring such as daily blood sugar check are two classes of health monitoring applications. There exist other applications for wireless sensor networks such as intelligent agriculture and environmental sensing or asset tracking and supply chain management. For each of these applications there are several examples, some of which are listed and explained briefly in [5].

The devices used in sensor networks perform both a *sensing* function, and a *data collection* function. In the proposed architectures for sensor networks, sensor nodes must self-organize to form a multi-hop network, and all collected and relevant data must be sent to a *sink node*, which is connected to a wired backbone network. This gives an overview on the functions and elements of a networks which should meet the requirements of the applications mentioned above. These requirements of specific applications may also dictate that sensor networks should have some features such as low power consumption, low cost, ease of availability, mobility, and security.

Some of the criteria to evaluate the performance of sensor networks include message latency and data throughput. Depending on the application, each of these features may become an important factor. Our assumption is that sensor nodes are expected to be deployed in the thousands, and thus need to be very low cost devices. This means that they operate on limited battery power, which can translate to both limited computational power, and low transmission power. Since replacing the batteries may not be feasible or cheap in many applications, increasing the lifetime of the sensor network is one of the most important research challenges in the area.

1.1 Lifetime of Sensor Networks

Sensor nodes of sensor network are expected to be deployed in the thousands; they may be deployed to an area which is not accessible, or recharging the batteries of the sensor may not be cost effective. Therefore, extending the lifetime of a sensor network becomes an important area of research when the goal is to prolong the functionality of such a network. There exist different definitions for the lifetime of a network. One definition is the time that the first sensor node in the network fails because it is out of energy. The other definition extends the previous definition; lifetime of a network is the time for which the network can keep its connectivity.

Saving the energy in a sensor network can be discussed from different points of view. Clearly, the choice of the *network communication protocols* to be used influences the overall energy dissipation in a network. These protocols can be designed to minimize energy consumption, achieve fault tolerance, and perform local collaboration to reduce bandwidth requirements. For example, in comparison with conventional communication protocols, *LEACH* [12] provides evenly distributed energy load by means of localized coordination of subareas in the network (clusters) and local compression to reduce global communication . The other example is *MAP* (Medial Axis Based Protocol) which captures both geometric and topological features of a sensor network and provides an appropriate geometric abstraction of the network whose consequence is an excellent load balance [4].

Another mechanism to reduce energy requirement are the sleep and wake up schemes, which are implemented on top of the MAC protocol. Different schemes such as *on demand* [31], *scheduled rendezvous* [1], and *asynchronous schemes* [28] can be applied in different situation. For example, in on demand scheme, a node wakes up only when it receives a packet from a neighbor. In scheduled rendezvous, all the neighboring nodes wake up at a same time, periodically, for any potential communication. Unlike the first two schemes in asynchronous schemes nodes act independently to wake up for communication.

The topology of the network also affects the issue of energy saving. Two factors are quite relevant to contributing to reduce the lifetime in a sensor network. First, the energy consumed in routing to the sink is proportional to the number of hops in the route. Thus, nodes that are far away from the sink node, deplete energy at all the intermediate nodes in the path to the sink. Secondly, nodes that are close to the sink node have to forward packets on behalf of all other nodes in the network, and are therefore at risk of running out of battery power sooner than other nodes. One proposal to increase the lifetime of these networks is therefore to increase the density of nodes around the sink node; some percentage of these nodes close to the sink would be in sleep mode, waiting to take over when the other nodes die, and thereby keeping the network operational. This, however, does not address the first issue of long paths causing energy to be spent at many nodes in the network. Another proposal to increase the lifetime of these networks is to create multiple sink nodes, which can either be static or mobile. The second approach of using multiple sinks is of our interest in this work and we will explain it in more detail.

In such an architecture with multiple or mobile sinks , each sensor node would then send data only to the nearest sink node. Thus the distance from a sensor node to its preferred sink node would be smaller than in the single-sink case. Additionally, the burden of the nodes close to the sink node would be shared by many more nodes. [9, 30, 20, 27, 16, 32] explore this idea, and we look at some these approaches in the next sections. Apart from increasing the lifetime of the network, using multiple sinks or mobile sinks can reduce the error rate, and latency since they reduce the number of hops to be traveled by each packet of data, and can handle a sparse and otherwise disconnected network [16].

1.2 Mobile Sinks

We consider the lifetime as an important issue in wireless sensor networks; therefore, mobility of the sink in such a network is proposed as a solution to save more energy in terms of multi-hop routing and to prolong the network lifetime.

For the first time Gandham et al. [9] developed the idea of having a mobile sink. Since the location of a mobile sink influences the multi-hop communication of sensor nodes and the mobile sink, they proposed an integer linear programming model to determine the location of mobile sinks which aims at minimizing the energy consumption per node plus minimizing the total energy consumption in a given time. Wang et al. in [30] also look at this problem with a different objective function which concerns the overall network lifetime directly instead of indirectly from minimization of energy consumption at each node.

Mobility can be applied to not only a sink but also *relays*. *Relays* are optional elements in a sensor network that can facilitate the task of multi-hop routing. Relays are limited in number and they can be recharged and therefore their energy can be considered unlimited in calculations [29]. They can be injected to a sensor network, then they move around in the area where the network is deployed, and they carry the data collected by sensor nodes. The consequence is that sensor nodes will be less involved in multi-hop routing. Mobility is so beneficial in such an extension that even a two-dimensional random walk for mobility of MULEs (Mobile Ubiquitous LAN Extensions), such as cars and animals (this can be an example of relays) increases the network performance and lifetime where the idea is to save energy by having single-hop communication from a sensor to mule which is passing by [27].

There are some drawbacks in using mobile sinks. For example, all the nodes in the networks have to be aware of the position of the sink since the goal of the network is to deliver the data to the sink. In contrast, a mobile relay is only responsible for multi hop routing, and therefore, other sensor nodes do not require to know its place. When a mobile relay passes by a sensor, it can carry the data which is available in that sensor and deliver it to a sensor which is closer to the sink. Thus, sensors between the source and destination of the relay do not get involved in multi hop routing. The other drawback of using mobile sinks is that it may not be possible for the sink to communicate with the backbone all the time, while the relay does not have such a task at all. Wang et al. in [29] believe that the approach which uses mobile relays instead of mobile sinks is more robust since a relay failure does affect the lifetime but not the functionality of the network while a sink is the user of the data provided by sensors, and its failure is a malfunction.

Although there are some disadvantages coupled with mobile sinks, Wang et al. in [29] compare the lifetime for different approaches to show that using a mobile sink outperforms all other approaches including mobile relays. In addition to the lifetime improvement as the premier advantage of using a mobile sink, mobility has other advantages according to A. Kansal *et al.* in [16]. The second advantage is the data fidelity. It is obvious that when the number of hops in multi-hop routing increases, the probability of error increases. In addition, static nodes spend less energy since the retransmission required due to errors is reduced when the error rate is smaller. The third advantage is that in some situations the data rate can be increased via decreasing latency. This does not mean that data travels faster over the links but the capacity of the sensor network will be increased because of carrying data physically in a mobile node. This advantage may be more applicable for mobile relays rather than mobile sinks. The fourth advantage is handling disconnectivity of sensor networks. As a result, static sensor nodes can reduce their transmission range to a lower value, and still communicate through smaller connected networks. Some other advantages of using mobile elements are finer time synchronization, security enhancement, and calibrating a localization system. Two issues about which to be concerned with regard to mobility are the motion control of mobile elements and the influence of speed on data collection, and each of them can be discussed extensively but are beyond the scope of this work.

In this thesis our goal is to find an optimal set of positions for a mobile sink to stop in the sensor network as a means to optimize the multi-hop routing in the network. Calculating the duration of each stop, or the routes or velocities between stops is beyond the scope of this work.

1.3 Multiple Sink Positioning

The basic motivation for developing the system architecture of a multiple sink sensor network is very similar to the motivation of having mobile sinks. Two main drawbacks of a static single sink sensor network, which are the bottleneck problem and the energy consumption in multi-hop routing, are the main reasons for looking at a multiple sink architecture, but there are also other issues to be considered. In [9] the problem of locating multiple sinks in the network in order to maximize the lifetime of the network while the number of multiple sinks is minimized was initially proposed. We look through the problems around multiple sinks in this section more closely.

Given a single sink sensor network, it is possible that the sink fails in the network.

A multiple sink network solves this shortcoming if such a situation is a possible scenario. Although the efficiency of a multiple sink architecture is very much the same as the mobile single sink in terms of the bottleneck and multi-hop routing problems, latency is the point of difference. Given a single mobile sink in a network, sensor nodes have to collect and store the data till the mobile sink moves to the closest stop point to the node. In fact not only can data delivery be delayed in this situation, but there is also the possibility of loss of data because of buffer overflow.

In fact, multiple sinks are beneficial in the above scenarios, but the possibility of having such an architecture depends on the application. An agricultural scenario is an example where using multiple sinks is applicable, since the network can be considered as a large scale network, and it is better to divide the network to a number of clusters to obtain a scalable network [26]. To deal with the routing in such a network different routing algorithms are also available such as ELBR and PBR which attempt to balance the energy consumption of sensor nodes [22].

If we use the multiple sink sensor network architecture, the issues that should be considered include finding the best sink locations. In the scenario that there exist a fixed number of sinks, clustering algorithms to find the efficient clustering of the sensor nodes can solve the problem. The method of *k-mean clustering* in [14] which is classified as a non-hierarchical clustering method or the generic method of self organizing maps, which is a general purpose unsupervised learning algorithm [17], are different alternatives. But these solutions all take the number of clusters or the number of multiple sinks as a parameter.

If we consider the number of multiple sinks as a variable, then the natural question which arises is to find the minimum number of sinks to achieve a desired lifetime for the network. Sometimes there exists a specific period during which the network should be functional.

For example, in an agricultural application the field must be monitored until the harvest. Assuming that there exist such a knowledge about the lifetime of the network, it is possible to start with one sink and estimate the network lifetime, then increment the number of sinks until the desired lifetime is achieved. We recall that if we know the number of sinks in the network it is possible to use the clustering method as an alternative to find their best locations.

However, it may not be possible to estimate accurately the lifetime of the sensor network in advance. Also, rather than achieving a specific lifetime, the goal may be to maximize the lifetime as much as possible. In other words, it is possible to have a scenario where there is not any prior knowledge about the number of sinks or the lifetime of the network. Then the question in such a scenario is to find the minimum number of sinks in the network while maximizing the network lifetime. Oymen has categorized the problems mentioned above in [26] and introduces the solutions existing in the literature for each of them.

In this thesis we are interested in question of finding the minimum number of sinks to maximize the lifetime. In multi-hop routing, if we limit the maximum number of the hops that a piece of data should travel to reach a sink, it can yield a significant saving in energy. The question we pose therefore, is to find the minimum number of sinks and their locations on the sensor network field, so that every packet has to travel at most h hops to reach a sink.

1.4 Problem Statement

In this work, we are interested in the problem of where to position the multiple sink nodes in order to further the goal of lifetime maximization. More specifically, suppose we want to ensure that the distance between a sensor node to its nearest sink is never more than some specified number of hops, say h. How many sink nodes should we place in the sensor field to guarantee this, and where should we place them? In the case of a mobile sink, we could think of the problem as finding the number and positions of "data collection stops" to be made by the mobile sink so that every sensor node is within h hops of a data collection stop.

For a sensor network which is scattered in an area I, there exists a link between two sensors, if each of them stays in the transmission range of the other. We assume there is only one type of sensor in the network, and therefore the transmission range of all sensors is the same; thus, if one sensor stays in the transmission range of another sensor, then there is a link between the two sensors. This means that the network can be represented as a *unit disc graph* where nodes are the sensors on the plane and node u is connected to node v by an edge if u lies in the unit disc centered at v. The unit disc represents the transmission range of v.

We define the *h-hop covering set problem* as follows: given a unit disc graph $G = (V, E)$, where V represents the set of sensor node locations, find the minimum-sized set of new points U (sink node locations) such that every node in V can reach a point in U using at most h hops. Apart from the edges in G, a sensor node can also reach in one hop a sink node which is within Euclidean distance one from it. Another way of describing the same problem is: given a unit disc graph $G = (V, E)$, find the minimum-sized set of new vertices/points U, such that U would be an h-hop dominating set in G', the unit disc graph obtained by augmenting G with U in the natural way (any pair of nodes in the augmented graph at distance less than 1 from each other is connected by an edge).

Depending on the actual application, it may not be feasible to place a sink node at any arbitrary position in the field. So we also consider two constrained versions of the problem.

In the first version, which we call the lattice-based covering set problem, we constrain the sink node positions to be at the points of a triangular lattice. The other constrained version of the problem investigated here, which we call the graph-based covering set, requires sink nodes to be placed at the nodes of G.

Given a set V of n points in the plane, a *disc cover* for V is a set of points U such that every point in V falls inside a unit disc centered at some point in U. In other words, every point in V is within distance one of some point in U. A disc cover whose elements are restricted to be a subset of points on a regular lattice is called a *lattice-based cover.*

Given a set of points V, the *disc cover problem* is to find the minimum sized disc cover for V. The *lattice-based disc cover problem* is to find a minimum sized lattice-based disc cover for V. We denote the size of an optimal disc cover for V by $|DC(V)|$ and an optimal lattice-based disc cover by $|LDC(V)|$. The disc cover problem has been studied extensively in [21, 11, 13, 3, 2]. The lattice-based disc cover problem was explained in [21]. We present these ideas in detail in chapter 2.

Recall that a unit disc graph is a graph $G = (V, E)$ where each element of V can be mapped to a point in the plane in such a way that $(u, v) \in E$ if and only if $||u - v|| \leq 1$. A *1-hop covering set* for the unit disc graph $G = (V, E)$ is a set of points U in the plane such that every element of V is within Euclidean distance 1 of some element of U. Put another way, U is a dominating set in the graph $G' = (V', E')$ where $V' = V \cup U$, and $E' = E \cup \{(x, u) \mid x \in V, u \in U, ||u - v|| \leq 1\}$. An *h-hop covering set* for G is a set of points U such that for every node $v \in V$, there exists a $v' \in V$ such that v is within $h - 1$ hops of v', which in turn is within Euclidean distance 1 of an element of U. Once again, U can be seen as an h-hop dominating set in the graph G augmented by the set U.

A covering set whose elements are constrained to be the points of a lattice is called a *lattice-based covering set.* A covering set for a graph G whose elements are constrained to be the vertices of the graph itself is called a *graph-based covering set* which is exactly the third version of the problem where the constraint is put the sinks on the same places as sensors.

In summary, given a unit disc graph G, the *covering set problem* is to find a minimum sized covering set for G. The *lattice-based covering set problem* is to find a minimum sized lattice-based covering set for G. The *graph-based covering set problem* is to find a minimum sized graph-based covering set for G. We denote the size of an optimal h-hop covering set of G by $|CS[h](G)|$, the size of an optimal h-hop lattice-based covering set for G by $|LCS[h](G)|$, and the size of an optimal h-hop graph-based covering set for a graph G by $|GCS[h](G)|$.

A 1-hop covering set for the unit disc graph $G = (V, E)$ is the same as a disc cover for V. In [15] this is shown to be NP-hard. Therefore, the h-hop covering set problem is also NP-complete. Since the 1-hop graph-based covering set problem is the same as the minimum dominating set problem, it is also NP-complete. In the lattice-based covering set problem, the number of lattice points is independent of the number of points to be covered. Since the covering discs are centered on lattice points, it is possible to find an optimal lattice-based 1-hop covering set in time linear in n and h-hop covering set in time quadratic in n. However, the constant factors for these optimal algorithms involve terms that are exponential in the size of the sensor network field. Therefore, we are interested in approximation algorithms for these problems to make the algorithm independent of the size of the field.

Given an approximation algorithm A for the h-hop covering set problem let $C_A(G)$ be the size of the cover produced by A on the graph G. Then the *performance ratio* of algorithm A is given by $max_G \frac{C_A(G)}{CS[h](G)}$. The approximation ratios of algorithms for the lattice-based and graph-based covering sets are defined similarly. Certain problems admit approximation algorithms that run in polynomial time and that have a performance ratio of $1 + \epsilon$ for any $\epsilon > 0$.

Definition 1 *[**Polynomial Time Approximation Scheme** (PTAS)] [7]*
An approximation scheme for an optimization problem is an approximation algorithm that takes as input not only an instance of the problem but also a value $\epsilon > 0$ such that for any fixed ϵ, the scheme is an approximation algorithm with relative error bound ϵ. We say that an approximation scheme is a polynomial-time approximation scheme if for any fixed $\epsilon > 0$ the scheme runs in time polynomial in the size n of its input instance.

1.5 Summary of Contributions

In the previous section we introduced the problems we study in this thesis. In this section we list all of our results and they will be discussed in detail in the following sections.

1. We give the first PTAS for the h-hop covering set problem, the h-hop lattice-based covering set problem, and the h-hop graph-based covering set problem.

2. We give a new PTAS for the lattice-based disc cover problem, based on a new approach derived from recent results on dominating sets in unit disc graphs. We show that this gives a $(3 + \epsilon)$-approximation algorithm for the disc cover problem, and give the first distributed algorithm for this problem.

3. We prove that the PTAS for the h-hop graph-based covering set problem can be used to derive a $(5 + \epsilon)$-approximation for the h-hop covering set problem. This algorithm

	Algorithm	Approximation	Time Complexity
1 hop	(shifting strategy) [13]	$(1+\frac{1}{\ell})^2$	$O(\ell^2 \lceil \ell\sqrt{2} \rceil^2 n^{2\lceil \ell\sqrt{2} \rceil^2+1})$
1-hop	(based on dominating set) [24] ∇	$5(1+\epsilon)$	$O(n^{c^2})$, $c = O(\frac{1}{\epsilon}\log\frac{1}{\epsilon})$
1-hop	(based on PTAS for lattice-based problem)[21]	$3(1+\epsilon)$	$O(c_1 n)$
1-hop	(based on PTAS for lattice-based problem)* ∇	$3(1+\epsilon)$	$O(c_2 n)$
h-hop	(Algorithm 1, with $\ell > h$)*	$(1+\frac{h}{\ell})^2$	$O(\ell^2 max(\lceil \ell\sqrt{2} \rceil^2, n) n^{2\lceil \ell\sqrt{2} \rceil^2+1})$
h-hop	(Algorithm 2) *	$(1+\frac{1}{\ell})^2$	$O(\ell^2 max(\lceil h\ell\sqrt{2} \rceil^2, n) n^{2\lceil h\ell\sqrt{2} \rceil^2+1})$
h-hop	(based on PTAS for graph-based problem)* ∇	$5(1+\epsilon)$	$O(n^{c_3})$
h-hop	(based on PTAS for lattice-based problem)* ∇	$3(1+\epsilon)$	$O(c_4 n^2)$

Table 1: Summary of results for h-hop covering set problem for unit disc graphs. c_1 and c_2 are functions of ϵ, while c_3 and c_4 are functions of ϵ and h; none of them depends on n, the number of nodes in the graph. * refers to results obtained in this thesis. ∇ refers to algorithms that have a distributed implementation

does not require knowledge of the geometric representation of the unit disc graph, unlike all other algorithms for the covering set problem.

4. We prove that the PTAS for the h-hop lattice-based covering set problem can be used to derive a $(3+\epsilon)$-approximation for the h-hop covering set problem. This algorithm runs in time $O(c_3 n^2)$ where c_3 is a function of ϵ and h, but does not depend on the number of nodes in the graph.

Our results for the covering set problem, along with previously known results are summarized in Table 2. As is shown in the table, the better approximation ratio we achieve, the worse time complexity is forced.

1.6 Outline of Thesis

In Chapter 2, we review the literature on the disc cover and dominating set problems. In Chapter 3, we describe our algorithm for the h-hop covering set problem and analyze the performance ratios and running times. Chapter 4 explains the geometric theorem which is used in chapter 3, and finally, Chapter 5 talks about the conclusions of this thesis and points to directions for future work.

Chapter 2

Related Work

In Chapter 1, minimizing the lifetime of a sensor network was introduced as an important problem in the area of sensor networks. Different approaches were introduced to increase the lifetime of a sensor network. These approaches are different according to the application, size, and cost of a sensor network. Among these approaches we are interested in using multiple or mobile sinks since having multiple or mobile sinks saves energy in terms of multi-hop routing, as well as avoiding a bottleneck at the sink node.

As mentioned in Section 1.4, the 1-hop version of our problem has already been studied as the disc cover problem, in which the goal is to find a set of covering discs on the plane for a given set of points. Also, the 1-hop graph-based covering set is identical to the dominating set, which is a well known graph-theoretic problem, which has been extensively studied. Our results originate from some of these known results and techniques, which we present in some detail in this chapter.

Hochbaum and Maass present a unified and powerful approach for the disc cover problem called the *shifting strategy* in [13]. In Section 2.1, their approach is introduced and discussed. Franceschetti, Cook and Bruck introduce a lattice-based solution to this problem in [21] to reduce the running time of the approach of [13]. Since the work in [21] is a solution for the 1-hop lattice-based multiple sink positioning problem (lattice-based disc cover problem), which we discuss in Section 3.2.1 , Section 2.2 contains some details of this work. Next, the graph-based version of the problem is the same as the dominating set problem; in section 2.3 we describe the algorithm of Nieberg and Hurink [24] for dominating sets in unit disc graphs, and we explain how it can be implemented in a distributed manner [19].

2.1 Covering and Packing Problem

Given a set of points in the plane, identifying a minimally-sized set of discs that cover all the points is a strongly NP-complete problem, as stated in Garey and Johnson's comprehensive review of this concept [10]. For the first time, [13] presented an algorithm with bounded approximation ratio for the disc cover problem. This means for a fixed $\delta > 1$, the ratio of the size of the solution delivered by the algorithm, and the size of an optimal solution for the problem does not exceed δ. Assuming that $\delta = 1 + \epsilon$ for $\epsilon > 0$ the functional dependence of the running time of the algorithm on the size of the input and $\frac{1}{\epsilon}$ is polynomial. Therefore, their scheme is said to be *fully polynomial.*

The fundamental technique of the algorithm in [13] is the *shifting strategy.* Section 2.1.1 describes this technique and the conditions for its applicability. This section also explains how this technique delivers a δ-approximation ratio. Section 2.1.2 explains the PTAS for the disc cover problem based on the shifting strategy. Section 2.1.3 explains the limitations of the shifting strategy which leads to another approach which is discussed in Section 2.2.

2.1.1 Shifting Strategy

The shifting strategy is a divide-and-conquer approach that bounds the approximation ratio for the disc cover problem. This strategy approximates the optimal solution; the difference of the size of the solutions with an optimal one is known as the error. For a given set of points the algorithm provides a number of solutions with different errors and the best solution should be selected among them.

We assume that we are given n points in a set N that are scattered in an enclosed area I on a plane. This gives a two dimensional input to the shifting strategy. [1] The goal is to cover all the points in N with a minimum number of discs of diameter D. Let ℓ be called the shifting parameter. The approximation ratio and the running time of the algorithm using the shifting strategy both depend on parameter ℓ. In the first step, the area I should be subdivided into vertical strips of width D which are left closed and right open. Observe that ℓ consecutive strips of width D make a larger strip of width $\ell \times D$ (Figure 1). As we consider these groups of strips of size $\ell \times D$ we can make ℓ different partitions on I such that in each of these ℓ partitions, I is divided into groups of width $\ell \times D$. Each partition can be derived from the previous partition by shifting all boxes of width $\ell \times D$ in the previous partition to the right (or left) by a distance D. After shifting ℓ times we will end up with the first partition that we started from. This means that we can make ℓ distinct partitions on I which are denoted as $S_1, S_2, \cdots, S_\ell$.

Assume that A is an algorithm that delivers a solution for the disc cover problem when we apply it on a bounded area such as any strip of width D, and Z^A denotes the size of the solution delivered by A. The value r_A denotes the ratio of Z^A to $|OPT|$ where $|OPT|$ is the size of an optimal solution for the problem.

[1]Although this strategy is not limited to 2-dimensional metrics on a plane, we do not discuss higher dimensions as it is out of our context.

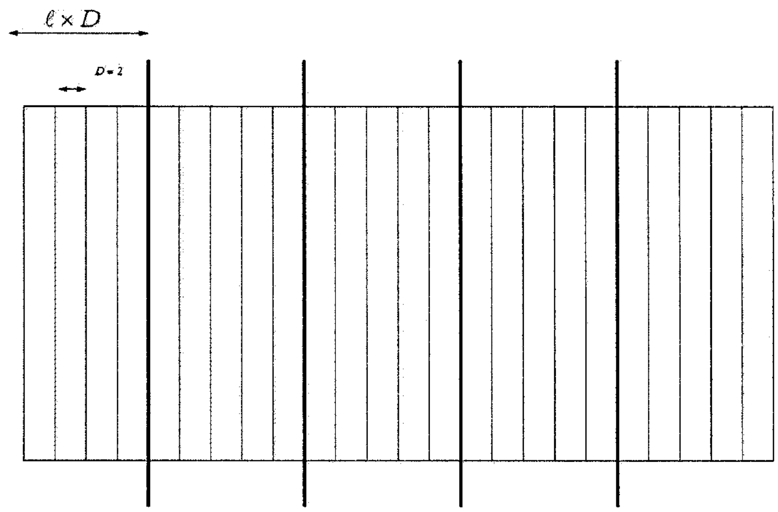

Figure 1: ℓ consecutive strips of width D make a larger strip of width $\ell \times D$

For a given partition S_i among ℓ partitions, $A(S_i)$ is the algorithm that applies algorithm A separately to each of the strips in S_i and then delivers the union of all discs as a solution to the entire problem. Although this may not be an optimal solution, it is a feasible solution to the disc cover problem which is defined on I. In fact, we can find such a feasible solution for the problem ℓ times as we have ℓ partitions; therefore, we have a set of solutions of disc cover sets which are delivered by $A(S_1), A(S_2), \cdots, A(S_\ell)$. The solution given by shifting strategy, called S_A, is the minimum-sized solution in the set $A(S_1), A(S_2), \cdots, A(S_\ell)$. Now we can provide the *Shifting Lemma* by Hochbaum and Maass , which shows that the performance ratio of the solution delivered by the shifting strategy for the disc cover problem is $(1 + \frac{1}{\ell})$.

Lemma 1 *(Shifting Lemma) [13]*

$$r_{S_A} \leq r_A(1 + \frac{1}{\ell}) \tag{1}$$

where A is a local algorithm, and ℓ is the shifting parameter.

This lemma and its proof with details are provided in [13]. This lemma explains the fact that the approximation ratio of a shifting algorithm which is applied to find a disc cover set, is at most $(1 + \frac{1}{\ell})$ worse than the best solution which is delivered by algorithm A on all partitions. If we assume that A delivers an optimal solution then we could prove that the approximation ratio of the shifting strategy is not worse that $(1 + \frac{1}{\ell})$. In other words, if A delivers an optimal solution, it means that $r_A = 1$, and therefore $r_{S_A} \leq 1 + \frac{1}{\ell}$ or $Z^{S_A} \leq (1 + \frac{1}{\ell})|OPT|$.

2.1.2 PTAS for disc cover problem

In the previous section the shifting lemma is discussed and it is shown that the approximation ratio of the shifting strategy to find a disc cover set is at most $(1 + \frac{1}{\ell})$. In this section it is shown that this approximation ratio can be achieved in polynomial time, and therefore the shifting strategy is a PTAS to solve the disc cover problem. This work is done in [13] for arbitrary dimensions.

Theorem 1 *[13] For any finite dimension $d \geq 1$, there is a polynomial time approximation scheme H^d such that for every given natural number $\ell \geq 1$, which is the shifting parameter, the algorithm H^d_ℓ delivers a cover of n given points in d-dimensional Euclidean space by d-dimensional balls of given diameter in $O(\ell^d \left\lceil \ell.\sqrt{d} \right\rceil^d 2n^{d\lceil \ell\sqrt{d}\rceil^d+1})$ steps with $(1 + 1/\ell)^d$ approximation ratio.*

Since our algorithm for h-hop covering sets described in Section 3.1 uses the shifting strategy, and our proof uses similar ideas to the proof of Theorem 1, we proceed to describe how the shifting strategy is a PTAS for the case $d = 2$. To show this, we apply the shifting strategy explained in 2.1.1 in two dimensions, vertically and horizontally. This means that we cut the plane into strips of width $\ell \times D$ vertically, and in the next step we cut it into strips of length $\ell \times D$ horizontally. Therefore, we will end up with squares of side $\ell \times D$

on the plane. The key point to prove the theorem is that we can find a disc cover set in such a square in time polynomial in $\hat{n}$, when $\hat{n}$ is the number of points in a square of side $\ell \times D$. The approach in [13] uses an exhaustive search in such a square which needs at most $\lceil \ell.\sqrt{2} \rceil^2$ discs of diameter D to be covered compactly. This is an upper bound on the size of any disc cover set in this square. One important observation is that any disc that covers at least two points has two of them on its border. This fact helps to count all the possibilities of disc cover sets in such a square since any disc which covers a set of points can be re-drawn such that at least two of the points lie on the border of the disc and other points of the set still remain covered. In fact, there are only two ways of drawing a disc when two points are on its border. Therefore, we need to check $2.\binom{\hat{n}}{2}$ to put the first disc, and then we check the possibility of disc covers of size 1 to $\lceil \ell.\sqrt{2} \rceil^2$. Thus, we have to check at most $O(\hat{n}^{2.\lceil \ell.\sqrt{2} \rceil^2})$ arrangements of discs. The next step is to check for each point whether it is in the range of any disc or not. This can be checked in $O(\ell^2.\hat{n})$ time for each of the arrangements. Consequently, in $O(\ell^2 \lceil \ell\sqrt{2} \rceil^2 n^{2\lceil \ell\sqrt{2} \rceil^2+1})$ steps we can find an optimal disc cover for a square of side $\ell \times D$, and since this is the same for all partitions, the running time of the shifting strategy on a plane (two dimensions) is polynomial. Because we use the shifting strategy in two dimensions, the approximation ratio of the result will be $(1+\frac{1}{\ell})^2$. Expanding it to $(1+\frac{1}{\ell}^2+\frac{2}{\ell})$, we can rewrite this approximation ratio in form of $1+\epsilon$ where $\epsilon \leq 3$ for $\ell \geq 1$ as the shifting parameter.

In Section 3.1 we show how to extend this work to the more general h-hop covering problem. Finally, it should be mentioned that the idea of the shifting strategy is originally proposed by Baker [2] and what we summarized was an adaptation for geometric disc cover problem.

2.1.3 Limitations of Shifting Strategy

The shifting strategy presented in Sections 2.1.1 and 2.1.2, has a $(1+\epsilon)$-approximation ratio which is equal to 4 in the worst case when the shifting parameter is 1 or in other words $\epsilon = 3$. The running time of the algorithm is $O(\ell^2 \lceil \ell\sqrt{2} \rceil^2 n^{2\lceil \ell\sqrt{2}\rceil^2+1})$, and for $\epsilon = 3$ it is $O(n^9)$. The degree of the polynomial for the running time increases as ϵ decreases. The sequence $9, 19, 51, \cdots, \lceil \ell\sqrt{2} \rceil^2 + 1$ gives this degree for $\ell = 1, 2, \cdots$. This is an important shortcoming of this strategy and makes it inefficient for even a small number of points since the running time is very expensive.

Some efforts to reduce the running time to find a solution for this problem are summarized in Table 2. A better approximation ratio of $1+\epsilon$ or $1+\frac{1}{\ell}$ where $\epsilon = \frac{1}{\ell}$ is provided in [8] by Feder and Greene and also in [11] by Gonzalez. Although the approximation ratio is better than what Hochbaum and Mass had, but the sequence of values which shows the polynomial degree of the running time starts with 13 and only grows slower than the former approach. The last row of the table shows an approach with linear running time which is introduced by Franceschetti and Cook. In the next section we describe this approach [21], which introduces the *grid strategy* to the disc cover problem.

	Author	Approximation	Running time	Year
1	Houchbaum and Mass [13]	$(1+\frac{1}{\ell})^2$	$O(\ell^2 \lceil \ell\sqrt{2} \rceil^2 n^{2\lceil \ell\sqrt{2}\rceil^2+1})$	1985
2	Gonzalez / Feder and Greene [11]	$(1+\frac{1}{\ell})$	$O(6\ell \lceil \ell\sqrt{2} \rceil n^{6\lceil \ell\sqrt{2}\rceil+1})$	1991
3	Gonzalez / Feder and Greene [11]	8	$O(n + n\log S)$	1991
4	Bronnimann and Goodrich [3]	$O(1)$	$O(n^3 \log n)$	1995
5	Franceschetti and Cook [21]	$\alpha(1+\frac{1}{\ell})^2$	$O(Kn)$	2000

Table 2: Approximation algorithms for disc cover problem.

2.2 Lattice-Based Geometric Disc Cover Problem

Franceschetti and Cook in [21] introduce another approach for the disc cover problem. They introduce the *grid strategy* to the disc cover problem which yields a linear running time at the cost of a worse approximation ratio. Using the grid strategy, the disc cover centers are limited to be selected only from a set of grid points. We call this the *lattice-based disc cover problem.* The algorithm in [21] combines both the shifting and grid strategies to solve this problem and provides an $\alpha(1+\frac{1}{\ell})^2$-approximation ratio for $\alpha = 3, 4, 5$, or 6. This section explains the grid strategy briefly, and elaborates on how we can get a linear running time using this strategy.

Franceschetti and Cook in [21] present a basic theorem in combinatorial geometry that helps to solve the disc cover problem. They use a square lattice to introduce their grid strategy and combine it with the shifting strategy presented in Section 2.1.1. The algorithm is very much the same as the PTAS discussed in Section 2.1.2. The same steps exist for dividing the plane into strips of width $\ell \times D$, then finding the set of disc covers locally in squares of side $\ell \times D$, but the difference is that the set of disc cover centers inside the square are only selected from a set of lattice points. If we cover the whole area of the plane with a square lattice, in any sub-area there exist a set of grid points as a group of candidates for the disc cover centers. In the geometric theorem that they present, they prove that if we take the disc cover centers from only the set of square lattice points, the approximation ratio of the algorithm which solves the disc cover problem is at most 3 times worse than the case that we do not restrict the location of the disc cover centers. Therefore, we can get a $3(1+\epsilon)$-approximation ratio. Although this is a worse performance ratio than the performance ratio of the PTAS in Section 2.1.2 (shifting strategy), it has a linear running time since it restricts the candidates for disc covers. In Section 2.2.1 we present the geometric theorem and then in Section 2.2.2 we discuss the linear running time.

2.2.1 Geometric Theorem

Theorem 2 *[21] Consider a square lattice where the distance between two neighboring lattice vertices is one. Call a disc of fixed radius r, centered at a lattice vertex, a grid disc. The number N of grid discs that are necessary and sufficient to cover any disc of radius r placed on the plane is given by:*

- *CASE 1. For $r < \frac{\sqrt{2}}{2}$, N does not exist.*
- *CASE 2. For $\frac{\sqrt{2}}{2} \leq r < \frac{\sqrt{10}}{4}$, $N = 6$.*
- *CASE 3. For $\frac{\sqrt{10}}{4} \leq r < 1$, $N = 5$.*
- *CASE 4. For $1 \leq r < \frac{5\sqrt{2}}{4}$, $N = 4$.*
- *CASE 5. For $r \geq \frac{5\sqrt{2}}{4}$, $N = 3$.*

The proof of this theorem is discussed in detail in [21] with a proof for the necessary and sufficient conditions for each case. The consequence of this theorem is that if we want to apply a lattice-based strategy, we are able to achieve an approximation ratio of 3 for a disc cover problem by using a square lattice. Therefore, when Franceschetti and Cook [21] combine this strategy with shifting strategy, which has $(1+\epsilon)$ approximation ratio, the resulting approximation ratio will be $3(1+\epsilon)$.

2.2.2 Linear Time Approximation Algorithm

Recalling from the shifting strategy that we restrict our exhaustive search to a square of side $\ell \times D$, Franceschetti and Cook [21] prove that the running time of the lattice-based covering problem is linear since it is possible to find the lattice-based disc cover inside the squares of side $\ell \times D$ in linear time. In such a square there exists p lattice vertices. Define K as a function of ℓ (shifting parameter) and p, such that $K_{(\ell,p)} = \ell^2 \Sigma_{i=1}^{\lceil \ell\sqrt{2} \rceil^2 - 1} \binom{p}{i} i$. Observe that $K_{(\ell,p)}$ does not depend on n, the number of points. They provide the following theorem:

Theorem 3 *[21] Let p, and $K_{(\ell,p)}$ be as defined before, and $\alpha \in \{3,4,5,6\}$. There is a* linear time approximation algorithm A_1 *such that for every given natural number $\ell \geq 1$, the algorithm A_1 delivers a cover of n points by discs of diameter D in $O(K_{(\ell,p)}n)$ steps, with approximation factor which is less or equal to $\alpha(1+\frac{1}{\ell})^2$.*

Because this linear time is the key feature of the lattice-based strategy, we will rephrase some parts of the proof here. The point is that with fixed ℓ and p then K is also fixed. But we need to explore what this K is. Restricting our search for a disc cover to a square of side $\ell \times D$ there exist p lattice vertices as the candidates. We recall that in such a square we need at most $\lceil \ell\sqrt{2} \rceil^2$ discs of diameter D to cover the square compactly. Therefore, we never need to check arrangements of more that $\lceil \ell\sqrt{2} \rceil^2 - 1$ grid discs. Since we need to find the distance of $\hat{n}$ given points in the square to grid centers in a potential solution we need $\left(\Sigma_{i=1}^{\lceil \ell\sqrt{2} \rceil^2 - 1} \binom{p}{i} i \right) \hat{n}$ steps to find a lattice-based cover in the square. Consequently, Franceschetti and Cook could come up with a linear time approximation algorithm that still has a reasonable approximation ratio of $3(1+\epsilon)$.

2.3 Dominating Sets in Unit Disc Graphs

In Chapter 1, we categorized the disc cover problem to three different versions where one of these versions is the *graph-based* disc cover problem. Here the goal is to cover all the nodes in a unit disc graph with a set of discs which are centered on the same points as the graph vertices. Thus, solving the graph-based disc cover problem is the same as solving the dominating set problem on the graph. In this section we describe the algorithm given by Nieberg and Hurink for the problem in [24]. The main reason for looking through their approach to solve the dominating set is the potentiality of their approach to be applied locally and in a distributed manner and the fact that it does not require a geometric representation of the graph. The former attribute is helpful in the h-hop version of graph-based disc cover problem and also in our approach for the lattice-based disc cover problem. In our lattice-based approach we preferred to combine the technique of Nieberg and Hurink in [24] with the grid strategy instead of the combination of the shifting and grid strategies because of the possibility to implement this technique in a distributed manner. This section discusses this technique from [24] and its adaptation to a distributed implementation from [19].

First we provide the definitions and preliminaries which are used in [24], and later we will describe their technique to find *local dominating sets* to generate the minimum dominating set of a unit disc graph with performance ratio of $(1 + \epsilon)$. Finally, it will be explained how this approach can be implemented a polynomial running time.

2.3.1 Definitions and Preliminaries

For $G = (V, E)$ where V is a set of vertices and E is a set of edges, a set $D \subset V$ is a *dominating set* for V if for every vertex $v \in V$, v is either in D or has a 1-hop neighbor in D. A *minimum dominating set* is a dominating set with minimum cardinality.

Another important definition is that of a neighborhood of a vertex in a graph, which can be defined for different distances. $N(v)$ denotes the immediate neighborhood of vertex v, or in other words, $N(v) := \{u \in V | (u, v) \in E\} \cup \{v\}$. The hop distance of two vertices

v and u is denoted by $d(u, v)$, and the r-th neighborhood of a vertex v, which is denoted by $N^r(v)$, includes the set of vertices such as u such that $d(u, v) \leq r$. We define $N(\phi) := \phi$ and $N^1(v) := N(v)$, and finally the r-th neighborhood of v is defined recursively $N^r(v) := N(N^{r-1}(v))$ for any $r \in N$. In addition, a neighborhood of a subgraph W is defined by $N(W) := \bigcup_{w \in W} N(w)$. Finally, D is defined as an operation on all possible subsets of vertices, which returns a minimum dominating set of such a subset. For example $D(W)$ dominates W where $W \subset V$. An important observation which is illustrated in Figure 2 is that $W \subset N(D(W))$ and $D(W) \subset N(W)$.

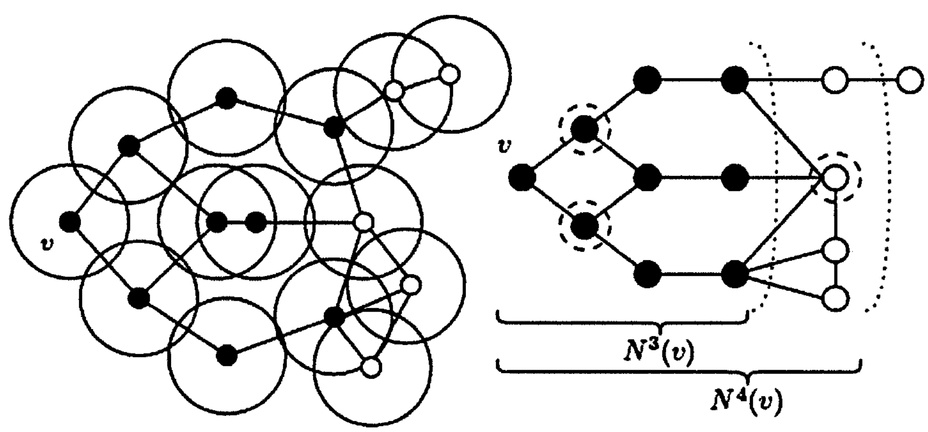

Figure 2: $D(W) \subset N(W)$, where W is $N^3(v)$, so $D(W)$ is in $N^4(v)$

2.3.2 Finding the Minimum Dominating Set

In this section we describe the technique of Nieberg and Hurink to find the minimum dominating set of a unit disc graph with $(1 + \epsilon)$ performance ratio.

This technique is based on finding dominating sets on subgraphs of G. For that, [24] introduces the concept of a *2-separated collection* of subsets, $S = \{S_1, \cdots, S_k\}$ which is defined as a collection of subsets of vertices $S_i \subset V$ for $i = 1, \cdots, k$ with the following property:

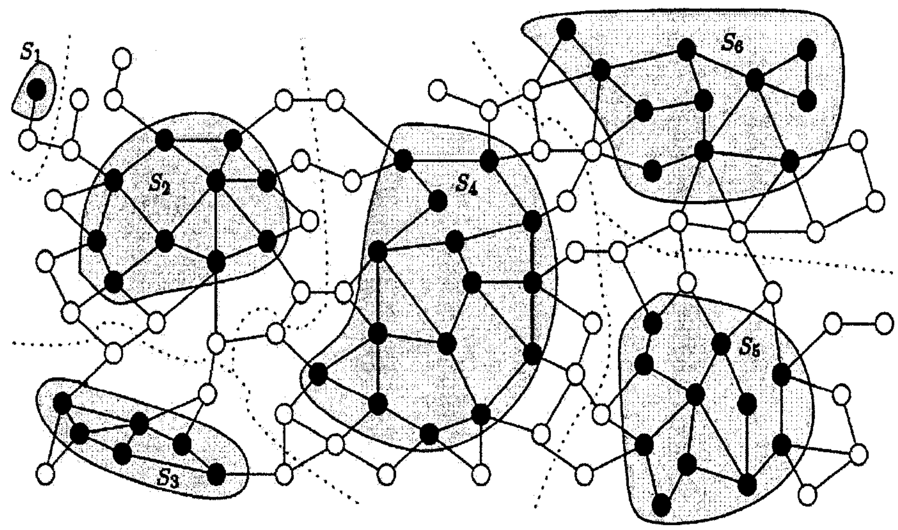

Figure 3: A 2-separated collection $S = S_1, \cdots, S_k$ in a graph $G = (V, E)$

- For any two vertices s and $\hat{s}$ such that $s \in S_i$ and $\hat{s} \in S_j$ and $i \neq j$, we have $d(s, \hat{s}) > 2$.

A key property of the subgraphs in a 2-separated collection, which is illustrated in Figure 3, is that since all the vertices in one subgraph have a hop distance greater than two to all the vertices in another subgraph, the dominating set of one of these two does not overlap with the other one since $D(W) \subset N(W)$ for any subgraph W. Therefore, it would be possible to prove the following lemma.

Lemma 2 *[24] For a 2-separated collection $S = S_1, \cdots, S_k$ in a graph $G = (V, E)$, we have*

$$|D(V)| \geq \sum_{i=1}^{k} |D(S_i)| \tag{2}$$

The proof of this lemma is provided in [24], but the consequence is important since it provides a lower bound on the minimum dominating set of the graph $G = (V, E)$. A

2-separated collection also helps to find an approximate solution. The idea behind this is to enlarge the subgraphs S_i to obtain another subgraph T_i such that the size of the minimum dominating set of T_i, which is $|D(T_i)|$, is smaller than $(1+\epsilon).|D(S_i)|$. Therefore we can conclude that $\sum_{i=1}^{k} D(T_i) \leq (1+\epsilon)\sum_{i=1}^{k} D(S_i) \leq (1+\epsilon)|D(V)|$, where the second inequality follows from Lemma 2. Thus, if we choose S_i and T_i such that $\bigcup_{i=k}^{k} D(T_i)$ dominates G completely, we could get a dominating set which is at most $(1+\epsilon)$ times the size of the minimum dominating set. This is proved as a corollary in [24] which says:

Corollary 1 *[24] Let $S = S_1, \cdots, S_k$ be a 2-separated collection in $G = (V, E)$, and let $T_1, \cdots, T_k$ be subsets of V with $S_i \subset T_i$ for all $i = 1, \cdots, k$. If there exists a bound $\epsilon \geq 0$ such that*

$$|D(T_i)| \leq (1+\epsilon).|D(S_i)|$$

holds for all $i = 1, \cdots, k$ and if $\bigcup_{i=k}^{k} D(T_i)$ forms a dominating set in G, the set $\bigcup_{i=k}^{k} D(T_i)$ is a $(1+\epsilon)$-approximation of and an MDS in G.

2.3.3 Construction of the solution

We need to know how to find subgraphs S_i and T_i such that they fulfill the conditions that are needed to construct a dominating set for G. Nieberg and Hurink [24] describe the algorithm to construct subgraphs S_i and T_i. The algorithm starts with an arbitrary vertex $v \in V$, and in the simplest scenario, it finds the dominating set of $N^r(v)$ for $r = 1, 2, \cdots$ using an exhaustive search, until the following inequality is violated:

$$|D(N^{r+2}(v))| \geq \rho |D(N^r(v))|$$

Lemma 3 *[24] There exists a constant $c = c(\rho)$ such that $\hat{r} \leq c$ that is, the largest neighborhood to be considered during the iteration of the algorithm is bounded by a constant.*

They call the smallest value that violates the inequality above $\hat{r}$. Then $N^{\hat{r}}(v)$ stands for the subgraph S_i, and $N^{\hat{r}+2}(v)$ for T_i. In the next step, the algorithm finds another arbitrary vertex v' in $G - N^{\hat{r}+2}(v)$ to construct another S_i and T_i, and does the previous steps iteratively until no vertex remains in G to be covered. It is clear from the construction that:

a: The S_i subgraphs form a 2-separated collection.

b: For each i, $|D(T_i)| \leq (1+\epsilon)|D(S_i)|$

c: The union of $D(T_i)$ for all i is a dominating set for G.

2.3.4 Polynomial Running Time

The algorithm in the previous section does not have any precondition of using a unit disc graph as the input, but this fact is used to prove the polynomial running time of the algorithm. In fact, in a unit disc graph, no matter how many immediate neighbors a vertex has, the maximum number of independent neighbors of any vertex does not exceed 5. This fact helps to conclude that the size of any independent set in any neighborhood of a vertex like v is bounded. In [24], the following lemma indicates this bound:

Lemma 4 *[24] Let $G = (V, E)$ be a UDG. Any independent set I^r of $N^r(v)$, when $v \in V$, satisfies:*

$$|I^r| \leq (2r+1)^2 = O(r^2).$$

Since the minimum dominating set is also an independent set, the consequence of this lemma is that even in the simplest scenario of exhaustive search to find the dominating set in S_i or T_i, it is possible to find the minimum dominating set in $O(n^{\vartheta})$, for some constant ϑ, which is a polynomial running time.

To summarize, we looked at the dominating set problem for unit disc graphs. We briefly explained the work in [24] which has two significant parts. The first thing is the algorithm that provides a $(1 + \epsilon)$-approximate solution for the minimum dominating set problem. Beside, it proves that this algorithm can be run in polynomial time. The result of these two is a PTAS for the dominating set problem in unit disc graphs.

2.3.5 Distributed Implementation

Two important advantages of the algorithm of Nieberg and Hurink in [24] are that first, it does not require a geometric representation of a graph G as an input, and second it can be implemented locally and in a distributed manner. In this section we explain the distributed implementation of their algorithm to find the minimum dominating set for a graph G. The distributed implementation is discussed in detail in [19]. We use the technique in [19] to implement the h-hop graph-based covering set problem in Section 3.3.1 in a local and distributed manner, and adapt this technique to implement the h-hop lattice based covering set problem in a distributed manner.

The distributed algorithm to find the minimum dominating set in [19] works for all *polynomially growth-bounded graphs*, which are defined as graphs such that there exists a polynomial function $f(r)$ such that for every $v \in V$ the size of the largest independent set in an r-neighborhood of v is at most $f(r)$. The algorithm has a pre-processing phase, which consists of clustering and coloring of a given unit disc graph $G = (V, E)$. To find the cluster graph $\overline{G} = (\overline{V}, \overline{E})$, where $\overline{V}$ is the set of cluster heads and $(u, v) \in \overline{E}$ if and only if $d_G(u, v) \leq c$ for a constant c, the algorithm constructs a maximal independent set, I, on G in a distributed and local manner using the algorithm in [18] and then assigns $\overline{V} = I$. The constant c is the same as that given in Lemma 3, and has the property that the largest neighborhood of a vertex to be considered when computing local dominating sets when starting with a vertex v is $N^c(v)$. The maximum degree of the resulting cluster graph is bounded by a constant since G is a polynomially growth-bounded graph. Therefore, $\overline{G}$ can be colored by an efficient and local coloring algorithm [6] using $\Delta_{\overline{G}} + 1$ colors, where $\Delta_{\overline{G}}$ is the maximum degree of $\overline{G}$.

After the pre-processing phase, for every vertex $v \in \overline{V}$ with color k, the algorithm applies the technique in [24] to find the minimum dominating set locally with the performance ratio of $1+\epsilon$. The key observation is that all the nodes with the same color can start the process at the same time, since the c-neighborhood of a vertex v with a given color is completely disjoint from the c-neighborhood of another vertex of the same color and $N^c(v)$ is the largest neighborhood of v that is involved during the computation of the dominating set while starting with v. After the computation for vertices with color k is finished, the vertices with color $k+1$ can start their computation. The performance ratio of $(1+\epsilon)$ as guaranteed by the centralized algorithm is also achieved by the distributed version.

Chapter 3

Covering Set Problem

In Chapter 1, we introduced the problem of multiple sink positioning or finding the stop points on a plane for a mobile sink, which can be rephrased as a covering set problem. We provide the definition of the h-hop covering set problem again:

Covering Set Problem: Given a unit disc graph $G = (V, E)$, where V represents a set of sensor node locations, find the minimum sized set of points U (sink node locations) such that every node in V can reach a point in U using at most h hops.

In this chapter we look at this problem in detail. Different algorithms are proposed for both 1-hop and h-hop versions of this problem. The performance ratio and time complexity of these algorithms will be discussed, and finally we compare the different approaches which are introduced for this problem.

3.1 PTAS for h-hop Covering Set Problem

Recall that the 1-hop covering set problem is the same as the disc cover problem, for which Hochbaum and Maass proposed a PTAS using the shifting strategy. This algorithm was described in detail in Section 2.1. We propose two algorithms for the h-hop covering set problem, which are extensions of the algorithm in Section 2.1, in two different ways.

The algorithm above can be extended for the h-hop problem in two different ways. We explore these two extensions in the next two sections.

3.1.1 Shifting Strategy for h-Hop Covering Set Problem: Algorithm 1

In our first solution for the h-hop covering set problem, we use the algorithm described above, except for the fact that at the bottom of the recursion, we look for optimal h-hop covering sets rather than disc covers. However, the analysis of our algorithm is different and we provide an out.

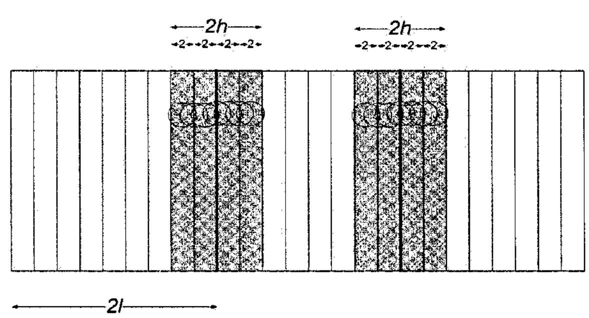

Figure 4: The set OPT^i is the shaded area. It includes all discs that are within h hops of the strip boundaries.

We choose $\ell > h$, and let $S_1, S_2, \ldots, S_\ell$ be the distinct shift partitions. Fix a specific partition S_i. We denote the optimal solution by OPT and let OPT^i be the set of discs in OPT which cover vertices in two adjacent 2ℓ-width strips given by the partition S_i. It is easy to see that all such discs are within Euclidean distance h of the 2ℓ-width strip boundaries corresponding to the partition S_i (see Figure 4). For j a specific 2ℓ-width strip in the partition S_i, let $|OPT_j|$ be the number of discs in an optimal solution for strip j. Then the above observation indicates that:

$$\Sigma_{j \in S_i} |OPT_j| \leq |OPT| + |OPT^i|$$

Consider a specific disc Φ in the set OPT. It is not difficult to see that Φ can be within h hops of the strip boundaries for at most h partitions. That is, it can belong to at most h sets OPT^i. This implies that

$$\Sigma_{1 \leq i \leq \ell} |OPT^i| \leq h|OPT|$$

Thus

$$min_{i=1,\ldots,\ell}\{\Sigma_{j \in S_i} |OPT_j|\} \leq (1 + h/\ell)|OPT|$$

Repeating the application of the algorithm to the second dimension, as in [13], we divide the strips into squares of side $2\ell \times 2\ell$. In each square, we find the optimal h-hop covering set using exhaustive enumeration. This algorithm results in a solution that is $(1 + \frac{h}{\ell})^2$ approximation to the optimal solution.

To analyze the complexity of the algorithm, note that as in [13], at most $\lceil \ell\sqrt{2} \rceil^2$ discs are needed to cover all the nodes in the $2\ell \times 2\ell$ square, and there are $O(n^2)$ positions for the disc centers to consider. Thus at most $O(n^{2\lceil \ell\sqrt{2} \rceil^2})$ arrangements of discs have to be considered. In order to check if an arrangement of discs is actually a cover, we can perform a breadth-first search for up to h levels from each of the points in the candidate cover, an operation that takes $O(n^2 + n\lceil \ell\sqrt{2} \rceil^2)$ time. Since there are ℓ shifts in each dimension, the total amount of time taken is $O(\ell^2 max(\lceil \ell\sqrt{2} \rceil^2, n) n^{2\lceil \ell\sqrt{2} \rceil^2 + 1})$ where $\ell > h$.

The above discussion yields the following theorem:

Theorem 4 *The h-hop covering set problem admits a PTAS.*

3.1.2 Shifting strategy for h-Hop Covering Set Problem: Algorithm 2

Another algorithm for the same problem is described briefly here. In this algorithm we divide the area into strips of width $2h$ instead of 2. Let $S_1, S_2, \ldots, S_\ell$ be the distinct shift partitions which are made by groups of ℓ consecutive strips of width $2h$. Let OPT^i be the discs in OPT which cover vertices in two adjacent $2h\ell$-width strips given by the partition S_i. Then any disc in OPT^i must be within Euclidean distance h of the $2h\ell$-width strip boundaries given by partition S_i (see Figure 5). Thus any disc in OPT can appear in at most one such set OPT^i, and it follows that

$$\Sigma_{1\leq i\leq \ell}|OPT^i| \leq |OPT|$$

This implies that

$$min_{i=1,\ldots,\ell}\{\Sigma_{j\in S_i}|OPT_j|\} \leq (1+1/\ell)|OPT|$$

Thus, Algorithm 2 has a performance ratio of $(1 + 1/\ell)^2$. However, this improved performance ratio comes at the cost of a more expensive computation at the base of the recursion, where optimal h-hop covering sets must be found in squares of side $2h\ell$. In such a square the maximum number of discs that are needed to cover all the nodes in the square is $\lceil h\ell\sqrt{2}\rceil^2$ which means that at most $O(n^{2\lceil h\ell\sqrt{2}\rceil^2})$ arrangements of discs have to be considered. Therefore, the total time taken is $O(\ell^2 max(\lceil h\ell\sqrt{2}\rceil^2, n)n^{2\lceil h\ell\sqrt{2}\rceil^2+1})$.

3.1.3 Comparison between the two algorithms

Algorithm 1's performance ratio depends on h, while Algorithm 2's does not. On the other hand, Algorithm 2's running time depends on h while Algorithm 1's does not. Tables 3 and 4 compare the performance ratio and the time complexity of the two algorithms discussed above for specific values of ℓ and h. For $h \geq 2$, for any value of shifting parameter, Algorithm

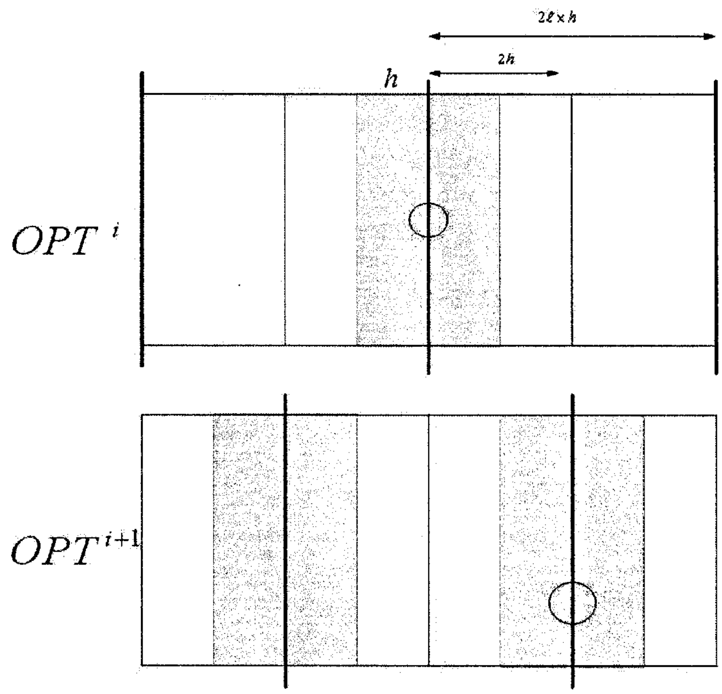

Figure 5: Each disc in OPT can appear in at most one set OPT^i

2 yields a better performance ratio. However, for $h > 1$, the time complexity of Algorithm 2 is much worse than that of Algorithm 1. For both algorithms, Table 4 confirms that the running time is a polynomial of very high degree in n for both algorithms.

	Algorithm 1							Algorithm 2
	$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$	$h = 10$	$h = 20$	for all h
$\ell = 5$	1.44	1.96	2.56	3.24	N/A	N/A	N/A	1.44
$\ell = 20$	1.10	1.21	1.32	1.44	1.56	2.25	N/A	1.10
$\ell = 50$	1.04	1.08	1.12	1.16	1.21	1.44	1.96	1.04

Table 3: The performance ratios of Algorithm 1 and Algorithm 2 for different values of ℓ and h.

	Algorithm 1	Algorithm 2						
	for all h	$h=1$	$h=2$	$h=3$	$h=4$	$h=5$	$h=10$	$h=20$
$\ell=5$	129	129	393	883	1509	N/A	N/A	N/A
$\ell=20$	1569	1569	6273	14113	25089	39201	156801	N/A
$\ell=50$	9801	9801	39201	44101	156801	245001	980001	3920001

Table 4: The time complexity of Algorithms 1 and 2. The exponent of n for different values of ℓ and h is shown.

3.2 $(3+\epsilon)$-Approximation Algorithm for Covering Set Problem

In the previous section, a polynomial time approximation scheme was explained for the covering set problem. The running time of Algorithm 1 was $O(\ell^2 max(\lceil \ell\sqrt{2}\rceil^2, n) n^{2\lceil \ell\sqrt{2}\rceil^2+1})$ where the approximation ratio is $(1+\frac{h}{\ell})$. Considering a sequence of values for ℓ, $\{1, 2, 3, \cdots, 10, \cdots\}$, the sequence of the degree of n is $\{5, 17, \cdots, 401, \cdots\}$, which means that it grows very fast. This is the main motivation to look for a trade-off between the running time and approximation ratio. In this section we explore the idea of superimposing a regular lattice on the plane to achieve this trade-off. This means that we consider only lattice-based disc covers, ie. we restrict the covering disc to center only on lattice points. This will affect the approximation ratio of the solution that we propose for the covering set problem, but a linear time will be achieved.

In Section 3.2.1 we propose a PTAS for the lattice-based disc cover set problem. In Section 3.2.2 this algorithm will be generalized for the h-hop version of the problem. We also prove that these two algorithms deliver a $(3+\epsilon)$-approximation ratio solution for the covering set problem, and finally, Section 3.2.3 presents the first distributed algorithm for the lattice-based disc cover set problem.

3.2.1 PTAS for Lattice-based Disc Cover Problem

In this section, we describe a PTAS to find a lattice-based disc cover for a given set of points V. We adapt and extend the ideas used in [24] to find the minimum dominating set of a unit disc graph. For the algorithm to work, a triangular lattice of side at most $\sqrt{3}$ or a square lattice of side at most $\sqrt{2}$ is required. However, in this section, we assume that the triangular lattice has side $2/\sqrt{7}$; the approximation bounds given here depend on this assumption. In Chapter 4, we will discuss the relationship between the lattice size and the performance ratio in general.

The infinite triangular lattice can be seen as tiling the plane with hexagons, such that each lattice point u is the center of a unique hexagon $H(u)$ (see Figure 6). Given a finite set of lattice points U, the set of hexagons associated with U is $\{H(u) \mid u \in U\}$. We say that two hexagons are *connected* if they share a side. A set of hexagons is said to be connected, if for any two hexagons H_i and H_j in the set, there is a *sequence* of hexagons from the set connecting them.

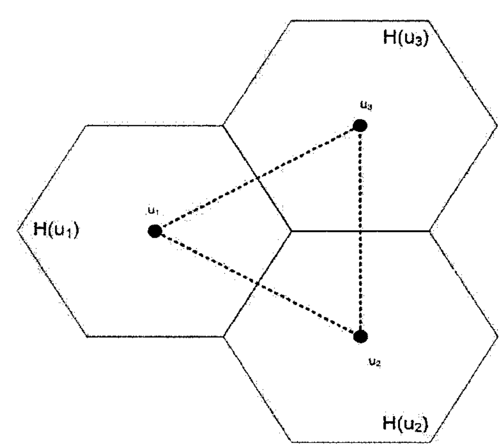

Figure 6: Each lattice point u is the center of a unique hexagon $H(u)$

Now that we have the definitions which are related to the triangular lattice, we recall

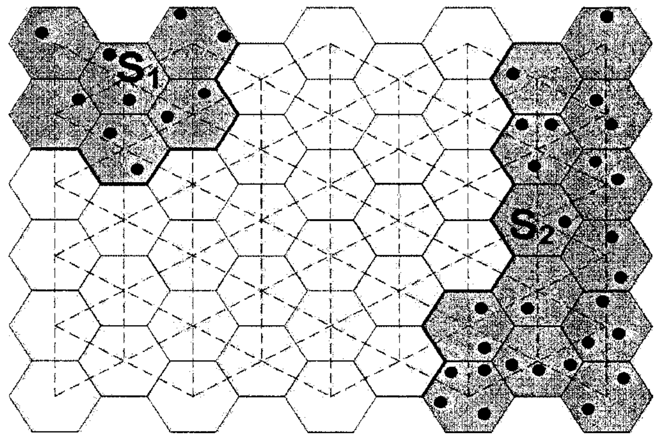

Figure 7: $\{S_1, S_2\}$ is a 4-separated collection of hexagons.

the definition of lattice-based disc cover set problem given in Chapter 1.

Definition 2 $S = \{S_1, S_2, ..., S_k\}$ *is a 4-separated collection of hexagons if*

1. S_i *is a set of connected hexagons, for all* i, $1 \leq i \leq k$.
2. S_i *and* S_j *are disjoint for all* $i \neq j$.
3. *if* $H \in S_i$ *and* $H' \in S_j$ *where* $i \neq j$, *then there are at least four hexagons between* H *and* H'.

See Figure 7 for an illustration of a 4-separated collection of hexagons. The motivation behind defining a 4-separated collection of hexagons is that the disc covers of S_i and S_j are completely disjoint for any distinct S_i and S_j in such a collection, as shown in Lemma 5.

Let u be a lattice point, and S be a set of hexagons. Then we define the neighborhood of H, denoted $N(H)$, to be the set of all hexagons which share a side with H. Similarly, the neighborhood of S, denoted $N(S)$, is $\bigcup_{H \in S} N(H)$. This definition can be extended to an r-neighborhood in a natural way:

$$N^r(H) = \begin{cases} N(N^{r-1}(H)) & \text{if } r > 1 \\ N(H) & \text{if } r = 1 \end{cases}$$

We use V_A to denote the subset of V that is contained in a region A.

The idea behind the four-separated collection of hexagons is that the lattice-based disc covers of each pair S_i, S_j in such a collection have to be disjoint, as shown in Lemma 5. This implies that the sum of the sizes of the lattice-based disc covers for all members of a four-separated collection forms a lower bound on a lattice-based disc cover for V. Our algorithm then expands the sets S_i in a controlled way to find a bounded-size lattice-based disc cover for the entire set V.

Lemma 5 *For any 4-separated collection of hexagons $S = \{S_1, S_2, ..., S_k\}$:*

$$\sum_{i=1}^{i=k} |LDC(V_{S_i})| \leq |LDC(V)|$$

Proof. Fix an S_i. As shown in Figure 8, the point u_1 is a lattice point in $N^3(S_i) - N^2(S_i)$ with minimum distance to any point inside the S_i, and therefore to any element of V_{S_i}. Since the lattice distance is $d = 2/\sqrt{7}$, the minimum distance from u_1 to S_i can be easily seen to be $\sqrt{52/21} > 1$ from triangle $\triangle u_1 mn$. This means a lattice disc (with radius 1) centered at any lattice point in $N^3(S_i) - N^2(S_i)$ cannot possibly cover any point in S_i. Since S is a 4-separated collection, if $i \neq j$, then $LDC(V_{S_i}) \cap LDC(V_{S_j}) = \emptyset$. The lemma follows. □

Algorithm Given a 4-separated collection, we would now like to expand each S_i to T_i such that the union of disc covers for V_{T_i} form a disc cover for all of V, and the disc covers

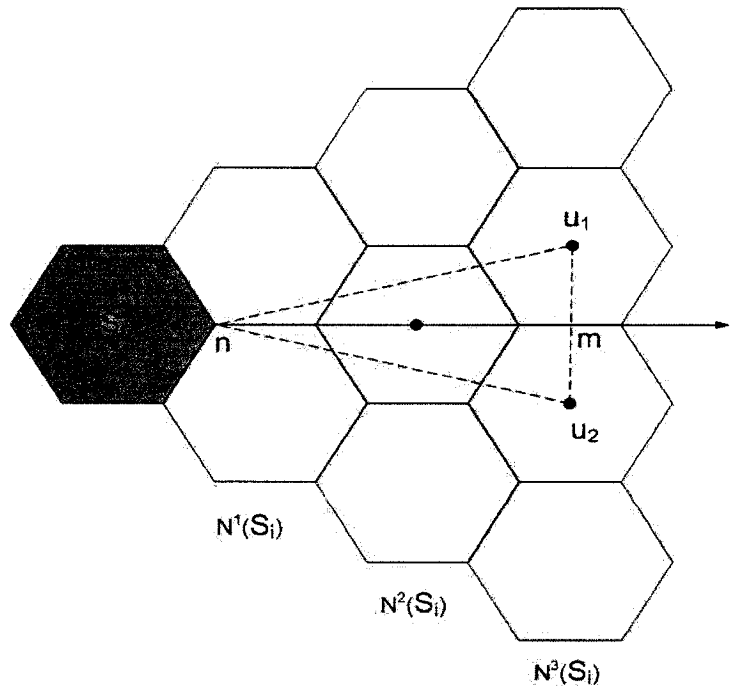

Figure 8: The minimum distance of any lattice point inside $N^3(S_i)$ is greater or equal than r, which is $\sqrt{52/21} > 1$

for each T_i is at most $(1+\epsilon)$ times the size of that for the corresponding S_i. In this manner, the union of all the disc covers in T_i will form a disc cover which is at most $(1+\epsilon)$ times the optimal disc cover for V. Given a set of points V, let R be the subset of hexagons chosen from the hexagonal tiling of the plane, that each contain at least one point from V. Let $i = 1$. We initialize H_i to be an arbitrary hexagon from R. Next, define $S_i = N^r(H_i)$ and $T_i = N^{r+4}(H_i)$ where r is the smallest integer such that $|LDC(V_{N^{r+4}(H_i)})| \leq (1+\epsilon)|LDC(V_{N^r(H_i)})|$. Having done this, we remove T_i from R, increment i and continue until there is no hexagon left in R [1]. Let the number of iterations for R to become empty be k. The disc cover returned by the algorithm is $\bigcup_{1\leq i\leq k} LDC(T_i)$, where $LDC(T_i)$ is an optimal lattice-based disc cover for the set T_i.

[1] We point out here that the neighborhoods above are formed using the infinite hexagonal tiling, and not only the hexagons in R.

It is clear from the description above that $\{S_1, S_2, \ldots, S_k\}$ as formed by the algorithm above is a 4-separated collection of hexagons. It is also straightforward to see that $\cup_{1 \leq i \leq k} V_{T_i} = V$. Therefore, $\cup_{1 \leq i \leq k} LDC(V_{T_i})$ is indeed a lattice-based disc cover for V. The following lemma shows that it is in fact a $(1+\epsilon)$ approximation of the optimal lattice-based disc cover for V.

Lemma 6

$$\Sigma_{i=1}^{k} |LDC(V_{T_i})| \leq (1+\epsilon)|LDC(V)|$$

Proof. By design, $|LDC(V_{T_i})| \leq (1+\epsilon)|LDC(V_{S_i})|$ for every i such that $1 \leq i \leq k$. Therefore,

$$\Sigma_{i=1}^{k} |LDC(V_{T_i})| \leq (1+\epsilon)\Sigma_{i=1}^{k} |LDC(V_{S_i})| \leq (1+\epsilon)|LDC(V)|$$

where the second inequality follows from Lemma 5. □

It remains to show that the algorithm described above runs in polynomial time. We do this in two steps: first we show that for any constant r, and for any hexagon H_i, an optimal lattice-based disc cover $LDC(V_{N^r(H_i)})$ can be found in polynomial time. Secondly, we show that there is a constant c such that the largest neighborhood to be considered in the process of finding the sets S_i and T_i is always bounded by c.

For any fixed r, and any hexagon H_i in the hexagonal tiling, the number of hexagons in $N^r(H_i)$ is given by $(\Sigma_{i=1}^{r} 6i) + 1 = 3r^2 + 3r + 1 = O(r^2)$. Therefore, $LDC(V_{N^r(H_i)})$ can be found simply by enumerating all the possibilities. We start with all singleton subsets of lattice points in $N^{r+2}(H_i)$ and look at progressively larger subsets until a disc cover is found. There are at most $O(2^{O(r^2)})$ such subsets which are candidate covers; for each such candidate cover, it can be verified in $O(\widetilde{n})$ time whether it is indeed a cover, where

$\widetilde{n}= |V_{N^r(H_i)}|$. Therefore the total time taken for this exhaustive search is $O(2^{O(r^2)}\, \widetilde{n})$, which is linear in $\widetilde{n}$, provided r is constant. If $S_i = N^r(H_i)$, then, starting with H_i, the time taken to identify the sets S_i and T_i as well as find the optimal lattice-based disc covers for them is at most r times the time taken to find $LDC(N^r(H_i))$, which is still linear in the number of nodes in T_i for constant r. Since the sets V_{T_i} for $1 \le i \le k$ are disjoint, the total time taken by the algorithm is linear in n, the total number of nodes.

Lemma 7 *There exists a constant $c = c(\epsilon)$ such that the largest neighborhood to be considered during the process to find S_i and T_i for any i is bounded by that constant.*

Proof. Let c be the largest neighborhood considered while calculating S_i and T_i for some value of i. Recall that if $S_i = N^r(H_i)$, then $T_i = N^{r+4}(H_i)$, and that in finding the dominating set for S, we need to look at all lattice points in $N^2(S)$. Then for any $r < c$,

$$
\begin{aligned}
3(r+2)^2 + 3(r+2) + 1 &\ge |LDC(N^r(H_i))| \\
&> (1+\epsilon)|LDC(N^{r-4}(H_i))| \\
&> (1+\epsilon)^2|LDC(N^{r-8}(H_i))| \\
&> \dots \\
&> (1+\epsilon)^{\lfloor r/4 \rfloor}|LDC(N^{r \bmod 4}(H_i))| \\
&> (1+\epsilon)^{\lfloor r/4 \rfloor}
\end{aligned}
$$

where the last inequality holds since $1 \le |LDC(N^{r \bmod 4}(H_i))| \le \alpha$ for some constant α. It is clear that the inequality $3(r+2)^2 + 3(r+2) + 1 > (1+\epsilon)^{\lfloor r/4 \rfloor}$ will be violated eventually, for a value of r that depends only on ϵ and not on n, the number of points in V. □

The following theorem is immediate:

Theorem 5 *The lattice-based disc cover problem admits a PTAS.*

The above theorem, together with Theorem 8 (from Chapter 4) can be used to give an approximation algorithm for the disc cover problem.

Corollary 2 *For any fixed ϵ, and for any set of points V on the plane, there is a linear time algorithm that computes a disc cover of V that is at most $3(1+\epsilon)$ times the size of the optimal disc cover.*

Proof. Let $S(V)$ be the disc cover given by the PTAS of Theorem 5. Then,

$$|S(V)| \leq (1+\epsilon)|LDC(V)| \leq (1+\epsilon)(3|DC(V)|)$$

where the second inequality follows from Theorem 8 (Chapter 4), which implies that any disc on the plane which is not collocated with a lattice point, can be substituted with at least 3 lattice discs. □

3.2.2 h-Hop Lattice-based Covering Sets

As stated earlier, a 1-hop lattice-based covering set for a graph $G = (V, E)$ is the same as a lattice-based disc cover for V. Therefore, for any unit disc graph, the algorithm in Section 3.2.1 gives a PTAS for the lattice-based covering set problem, and a $3(1 + \epsilon)$-approximation algorithm for the covering set problem.

Corollary 3 *For any fixed $\epsilon > 0$, and for any unit disc graph G, there is a linear time algorithm that computes a 1-hop covering set for G that is at most $3(1 + \epsilon)$ times the size of the optimal 1-hop covering set.*

To find the h-hop lattice-based covering set for a unit disc graph G, we use similar ideas to the algorithm in Section 3.2.1. Essentially, we create a $2m$-separated collection of hexagons $\{S_1, S_2, \ldots, S_k\}$, where m is given by the following:

$$m = \begin{cases} 2i & \text{if } 6i - 2 \leq \sqrt{21}h < \sqrt{36i^2 + 12i + 4} \\ 2i + 1 & \text{if } \sqrt{36i^2 + 12i + 4} \leq \sqrt{21}h < 6(i+1) - 2 \end{cases} \quad (3)$$

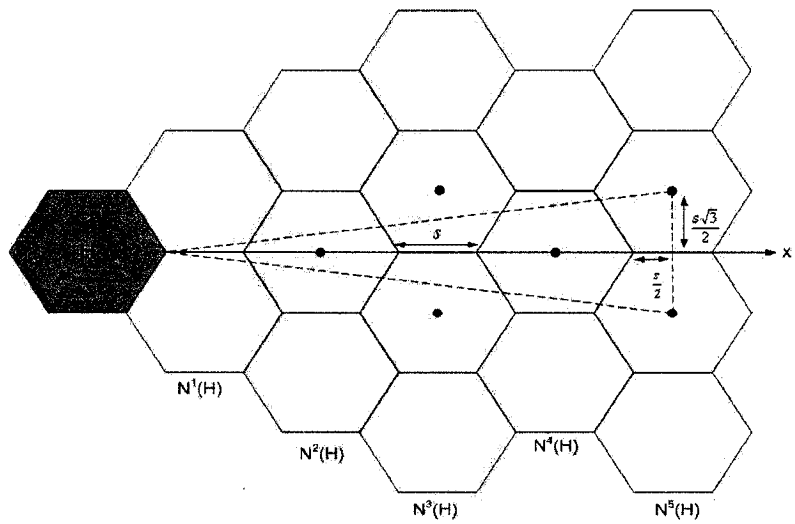

Figure 9: Finding the minimal m such that no lattice point in $N^{m+1}(H)$ can cover any point in H. The value $s = d/\sqrt{3} = 2/\sqrt{21}$.

Claim 1 *The value m defined in Equation 3 is the minimum value such that no lattice point in $N^{m+1}(H)$ can be within h hops from any node located inside the hexagon H.*

Proof. Observe that when the lattice distance $d = 2/\sqrt{7}$, the side of the hexagon is $d/\sqrt{3} = 2/\sqrt{21}$. From Figure 9, it is easy to see that among all the lattice points in $N^{2i}(H)$, the one that lies on the x-axis minimizes the distance to the hexagon H. The minimum distance from H to a lattice point in $N^{2i}(H)$ is therefore $(3i-1)2/\sqrt{21}$. Similarly, among all lattice points in $N^{2i+1}(H)$, the ones that are just above or just below the x-axis minimize the distance to the hexagon H. The minimum distance from H to a lattice point in $N^{2i+1}(H)$ can be calculated to be $\sqrt{(36i^2 + 12i + 4)/21}$. Since in one hop, at most distance 1 can be covered, for a lattice point u in $N^j(H)$ to cover a point in H, it must be that the distance between u and H is not greater than h. The lemma follows. □

The separation between S_i and S_j for $i \neq j$ implies that $LCS[h](G_{S_i}) \cap LCS[h](G_{S_j}) = \emptyset$, where G_A is the graph induced by the vertices of G lying in region A. Once again, we choose a hexagon H_i with a vertex that has not yet been covered, and assign $S_i = N^r(H_i)$, and $T_i = N^{r+2m}(H_i)$ where r is chosen such that $|LCS[h](G_{T_i})| \leq (1+\epsilon)|LCS[h](G_{S_i})|$. Assuming that after i such rounds, all vertices have been covered, we can use similar arguments to the 1-hop case to show that $\cup_{1 \leq i \leq k} LCS[h](G_{T_i})$ is a $(1+\epsilon)$-approximation to the optimal lattice-based h-hop covering set. The following lemma establishes that as in the one-hop case, there is a constant c such that the size of the largest neighborhood to be considered during the process to find S_i and T_i is bounded by c.

Lemma 8 *There exists a constant $c = c(h, \epsilon)$ such that the largest neighborhood to be considered during the process to find S_i and T_i for any i is bounded by that constant.*

Proof. Let c be the largest neighborhood considered while calculating S_i and T_i for some value of i. Recall that if $S_i = N^r(H_i)$, then $T_i = N^{r+2m}(H_i)$, that in finding the dominating set for a set of hexagons S, we need to look at all lattice points in $N^m(S)$, and that the number of lattice points in $N^i(H)$ for a hexagon H is $3i^2 + 3i + 1$. Then for any $r < c$,

$$\begin{aligned}
3(r+m)^2 + 3(r+m) + 1 &\geq |LCS[h](N^r(H_i))| \\
&> (1+\epsilon)|LCS[h](N^{r-2m}(H_i))| \\
&> (1+\epsilon)^2|LCS[h](N^{r-4m}(H_i))| \\
&> \cdots \\
&> (1+\epsilon)^{\lfloor r/2m \rfloor}|LDC(N^{r \bmod 2m}(H_i))| \\
&> (1+\epsilon)^{\lfloor r/2m \rfloor}
\end{aligned}$$

where the last inequality holds since $1 \leq |LDC(N^{r \bmod 2m}(H_i))| \leq \alpha(m)$ for some function of m. It is clear that the inequality $3(r+m)^2 + 3(r+m) + 1 > (1+\epsilon)^{\lfloor r/2m \rfloor}$ will be violated eventually, for a value of r that depends only on ϵ and m (which is a function of h) and not on n, the number of points in V. □

To analyze the time complexity, we observe that the difference between the h-hop case and the 1-hop case is that it takes $O(n^2)$ time to verify if a given candidate set of lattice points is in fact an h-hop covering set. This is because we perform a breadth-first search for upto h levels from each of the points in the candidate cover to verify that all vertices in G are covered. Since the number of candidate sets is independent of n, the total time taken is $O(c_2 n^2)$ where c_2 is a function of ϵ and h.

Theorem 6 *The h-hop lattice-based covering set problem on unit disc graphs admits a PTAS.*

Corollary 4 *For any ϵ, and for any unit disc graph G, there is a quadratic time algorithm that computes a h-hop covering set for G that is at most $3(1+\epsilon)$ times the size of the optimal h-hop covering set.*

Proof. Take any point u in the optimal h-hop covering set. Then the disc centered at u can be covered by three lattice discs as shown in Chapter 4. Any vertex of the graph that is at most h hops away from u is also at most h hops away from at least one of three lattice disc centers that cover the disc centered at u. So if $S(G)$ is the solution given by the PTAS of Theorem 6, then

$$|S(G| \leq (1+\epsilon)|LCS[h](G)| \leq (1+\epsilon)(3|CS[h](G)|)$$

□

3.2.3 Distributed Algorithm for Lattice-based Disc Cover Problem

The algorithms in Sections 3.2.1 and 3.2.2 have one important property. These algorithms can be implemented in a distributed manner. As far as we know, this is the first distributed algorithm for the disc cover problem. The distributed implementation of the algorithms retains the same performance ratio as the centralized version. In this section, we show how to implement the algorithm for lattice-based disc cover problem given above in a distributed manner.

The main idea of the distributed implementation of the algorithm is to use a pre-defined coloring of the hexagonal lattice, which satisfies the constraint that the *reuse distance* is $2c + 1$. Here c is the constant derived in Lemma 7 in Section 3.2.1, which is the size of the largest neighborhood to be considered during the process to find S_i and T_i for any i. For that, we first define the reuse distance here:

Definition 3 *In a valid coloring with reuse distance k, any two vertices at distance $\leq k$ must be assigned different colors.*

Such a coloring can be obtained by partitioning the tiling into identical *clusters* (see Figure 10), such that every hexagon in a cluster has a different color, and corresponding hexagons in different clusters are assigned the same color. These ideas have previously been studied in the context of channel assignment in cellular networks [23]. The mechanism to form the clusters is simple: the cluster corresponding to a hexagon is simply its c-neighborhood. It is straightforward to identify a subset of hexagons from the tiling such that their clusters partition the entire tiling. The number of colors needed for the coloring is therefore the size of a c-neighborhood, that is $3c^2 + 3c + 1$. Finally, we observe that the color given to a particular hexagon is derivable simply by the knowledge of the coordinates.

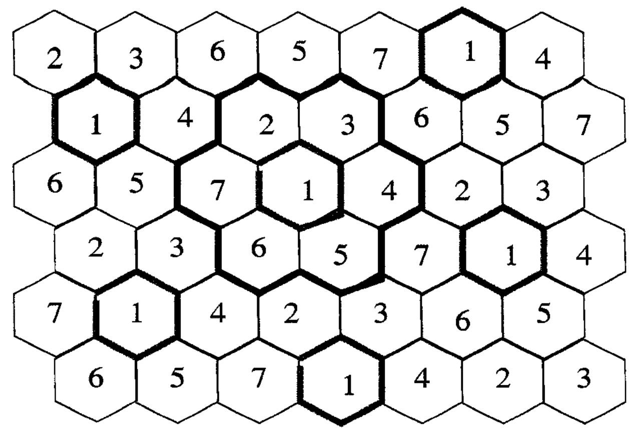

Figure 10: Clustering using a 1-neighborhood

Now, the lattice-based covering set will be derived in stages, defined by the coloring. Let $H^i = \{H_{i_1}, H_{i_2}, \ldots, H_{i_j}, \ldots, H_{i_k}\}$ be the set of hexagons with color i. In the i-th stage, all the hexagons in H^i participate. In particular, H_{i_j} is used as a seed to create the sets S_{i_j} and T_{i_j} as outlined in the previous section. The key observation is that all hexagons in H^i can start the process simultaneously: any two hexagons H_{i_j} and H_{i_l} in H^i are of the same color, and are therefore at least $2c+1$-distance apart. By the definition of c, the neighborhoods considered in the construction of T_{i_j} and T_{i_l} are completely disjoint. Then, the vertices that are covered are removed from consideration, and we proceed with hexagons of color $i+1$.

An outline of the distributed implementation follows. In the i-th stage, for every j, all nodes (if any) in the hexagon H_{i_j} participate in a local leader election to elect one representative (at most) from H_{i_j}. This leader node now broadcasts a request for information about the positions of all nodes within its c-hexagon neighborhood, where c is the constant defined in Lemma 7. Next, the leader node locally computes the S_{i_j} and T_{i_j} sets as well as the optimal h-hop covering sets for $G_{S_{i_j}}$ and $G_{T_{i_j}}$ as described earlier, and broadcasts the locations of the lattice points in these optimal covering sets. Any node that is covered by the announced lattice points now pulls itself out for the purpose of future rounds and will not respond to requests for information from leader nodes of hexagons of future colors. This finishes the i-th stage, and now nodes in hexagons of color $i + 1$ start their leader election.

The distributed implementation of the algorithm still yields the $(1 + \epsilon)$-approximation ratio for lattice-based disc cover problem, and the $(3 + \epsilon)$-approximation ratio for the disc cover problem. It is straightforward to see that the h-hop covering set and h-hop lattice-based covering set problems can be implemented in a distributed manner in the same way.

3.3 $(5 + \epsilon)$-Approximation Algorithm for Covering Set Problem

In the first two sections of this chapter we showed two different algorithms which deliver $(1+\epsilon)$ and $(3+\epsilon)$ approximation ratios for the covering set problem. In the first one, there was no restriction to locate the elements of the covering set, but in the second approach, which was the lattice-based approach, we restricted the covering set to be on the lattice points. This restriction caused a worse approximation ratio of $(3 + \epsilon)$, but it also provides a better running time which is discussed in Section 3.2.1. In this section another constraint will be introduced to our covering set problem. Here we restrict the positions of the covering disc to be on the vertices of the graph. In other words, we study the graph-based covering set problem; that is, we only consider points corresponding to nodes in the graph as candidates for inclusion into the covering set. A 1-hop graph-based covering set for a unit disc graph is the same as a dominating set in the graph, for which a PTAS is given in [24]. We go on to show a PTAS for the graph-based h-hop covering set problem. One important property of this algorithm is that it does not require a geometric representation of the unit disc graph.

Finally we will show that this algorithm which is a PTAS, delivers a $(5+\epsilon)$-approximation ratio for the covering set problem.

3.3.1 PTAS for h-Hop Graph-based Covering Set Problem

Recall that a covering set for graph $G = (V, E)$ whose elements are constrained to be the vertices of the graph itself is called a *graph-based covering set.* The algorithm proposed for the h-hop graph-based covering set is a straightforward extension of the 1-hop case [24], which is explained in Section 2.3. Like the algorithm in [24], our algorithm does not require a geometric representation of the unit disc graph.

In this algorithm, we use the fact that an h-hop graph-based covering set in a unit disc graph G is a dominating set in the graph G^h. We run Neiberg and Hurink's algorithm on G^h, since the algorithm itself is not specific to any type of graph. The proof of the polynomial run time for unit disc graphs depends on the fact that the size of the maximal independent set in any r-neighborhood of the graph is at most $(2rh+1)^2$ which is $O(r^2)$. However, the size of an MIS in G^h for G a unit disc graph is at most the size of an MIS in G.

To analyze the running time, observe that an optimal one-hop covering set in G^h is a minimum dominating set in G^h. Let $N^r(G, v)$ be the graph induced by the set of vertices in G that are at graph distance at most r from v in the graph G. Then:

$$|GCS[1](N^r(G^h, v)| \leq |MIS(N^r(G^h, v))| \leq |MIS(N^{rh}(G, v))| \leq (2rh+1)^2 \qquad (4)$$

where the last inequality bounding the size of an independent set in a unit disc graph is from [25]. Thus, to find the minimum dominating set in $N^r(G^h, v)$, we can do an exhaustive search in time $O(n^{O((rh)^2)})$. Furthermore, as in [24], we can show that the size of the maximum sized neighborhood examined by the algorithm is bounded by a constant c.

Lemma 9 *There exists a constant $c = c(h, \epsilon)$ such that the largest neighborhood to be considered during the process to find S_i and T_i for any i is bounded by that constant.*

Proof. Let c be the largest neighborhood considered while calculating S_i and T_i for some value of i. Then using an analysis exactly as in [24], and the bound on $|GCS[1](N^r(G^h, v))|$ obtained in Equation 4, for any $r < c$,

$$\begin{aligned}
(2rh+1)^2 &\geq |GCS[1](N^r(G^h, v))| \\
&> (1+\epsilon)|GCS[1](N^{r-2}(G^h, v))| \\
&> (1+\epsilon)^2|GCS[1](N^{r-4}(G^h, v))| \\
&> \ldots \\
&> (1+\epsilon)^{\lfloor r/2 \rfloor}|GCS[1](N^{r \bmod 2}(G^h, v))| \\
&= (1+\epsilon)^{\lfloor r/2 \rfloor}
\end{aligned}$$

where the last equality is a consequence of the fact that $|GCS[1](N^0(G^h, v))| = |GCS[1](N^1(G^h, v))| = 1$. It is clear that the inequality $(2rh+1)^2 > (1+\epsilon)^{\lfloor r/2 \rfloor}$ will be violated eventually, for a value of r that depends only on ϵ and h and not on n, the number of points in V. □

The following theorem is immediate:

Theorem 7 *The h-hop graph-based covering set for unit disc graphs admits a PTAS.*

Next, we show a relationship between the optimal graph-based covering set and the optimal covering set for a unit disc graph.

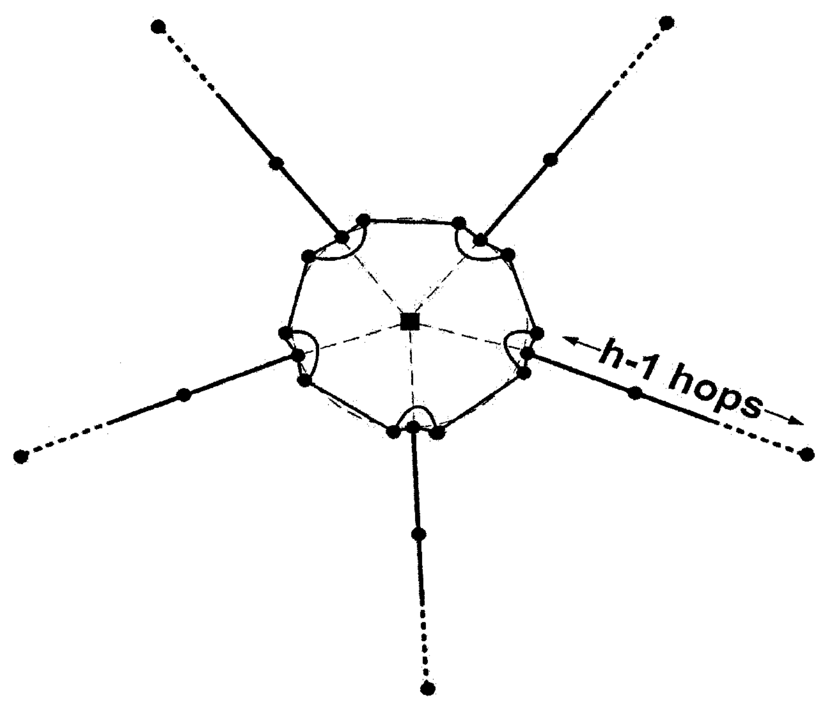

Figure 11: There are five discs in the optimal graph-based covering set, while the optimal covering set has only one disc (at the point in the middle of the circle.)

Lemma 10 *For every unit disc graph G, $|GCS[h](G)| \leq 5|CS[h](G)|$*

Proof. Let u be a point in the optimal h-hop covering set of G. Let $A(u)$ be the set of vertices in G that are within h hops of u. Consider the set of vertices of G that are contained in the unit disc centered at u, and let $B(u)$ be a maximal independent set of the subgraph of G induced by these vertices. Then $B(u)$ is also a dominating set of this subgraph, and every node in $A(u)$ is within h hops of some node in $B(u)$. Furthermore, $|B(u)| \leq 5$, since G is a unit disc graph, and the size of the maximal independent set in any neighborhood of a vertex in a unit disc graph is 5. This implies that we can obtain an h-hop graph-based covering set for G, by simply replacing every vertex u in the optimal covering set with an MIS of the subgraph induced by the disc centered at u, and the size of this covering set is at most five times the size of the optimal covering set, thereby proving the lemma. □

The bound on the approximation is tight, as shown by the example in Figure 11. The following corollary is a consequence of Theorem 7, and Lemma 10.

Corollary 5 *For any ϵ, for any integer $h > 0$, and for any unit disc graph G, there is a polynomial time algorithm that computes a h-hop covering set for G that is at most $5(1+\epsilon)$ times the size of the optimal h-hop covering set.*

Even though this algorithm has a worse performance ratio as well as worse running time than the algorithm in Section 3.2.2, the advantage of this algorithm is that it does not require a geometric representation for the graph.

In Section 2.3.5, we described the distributed implementation of the algorithm of Nieberg and Hurink for dominating set problem given in [19]. The results of [19] apply to all polynomially growth-bounded graphs. Since our algorithm for h-hop covering sets consists of running Nieberg and Hurink's algorithm on G^h, and G^h is a polynomially growth-bounded graph, it follows that the results of [19] also give a distributed implementation for our h-hop graph-based covering set algorithm.

Chapter 4

Geometric Theorem

In this section, we prove a result relating the size of an optimal disc cover to a lattice-based disc cover. This relationship depends on the side of the lattice d. However, regardless of the value of d, the minimum number of lattice discs required to cover an arbitrary disc is three, as shown below:

Lemma 11 *Any disc on the plane which is not co-located with a lattice disc needs at least three lattice discs to cover it.*

Proof. Assume for the purpose of contradiction that there is a non-lattice disc Φ with center c such that it is covered entirely by two lattice discs centered at A and B. If c lies on the line segment connecting the centers of A and B, then Figure 12 shows that there is a point at distance exactly one from c on the line segment perpendicular to $\overline{AB}$ that is at distance greater than 1 from the centers of both A and B. Similarly, if c is not on the line segment connecting A and B, the same argument applies. □

The following theorem gives a precise specification of the number of lattice discs that can be guaranteed to cover an arbitrary disc, for all possible values of the lattice side length.

Theorem 8 *Let $\mathcal{L}$ be the infinite triangular lattice where the distance between two neighboring vertices is d. The number N of lattice discs that are necessary and sufficient to cover an arbitrary unit disc placed on the plane is given by:*

Case 1: $d > \sqrt{3}$	*N does not exist*
Case 2: $1 < d \leq \sqrt{3}$	$N = 5$
Case 3: $\frac{2\sqrt{7}}{7} < d \leq 1$	$N = 4$
Case 4: $d \leq \frac{2\sqrt{7}}{7}$	$N = 3$

Proof. We examine each case separately, and prove that in the specified range of d, the given number of lattice discs N is sufficient to cover any disc on the plane (sufficient condition), and at the same time, there exists a disc Φ (more precisely, a disc *placement*) such that N lattice discs would be *required* to cover Φ (necessary condition).

4.1 Case 1

For $d > \sqrt{3}$ the lattice discs do not cover the plane completely (see Figure 13). In particular, the point M is at distance more than one from each of A, B, and C. Thus, there exists Φ (for example centered at M),such that there does not exist any integer N where N lattice discs could cover Φ.

4.2 Case 2: $1 < d \leq \sqrt{3}$

Next we consider the case when $1 < d \leq \sqrt{3}$. We show that five discs are necessary and sufficient to cover an arbitrary disc that is not collocated with a lattice point.

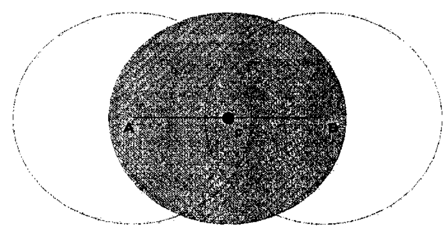

Figure 12: Two lattice discs cannot cover a disc placed on the plane with a center not located on a lattice point.

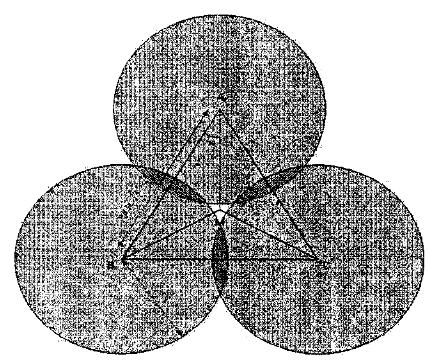

Figure 13: If $d > \sqrt{3}$, then N does not exist.

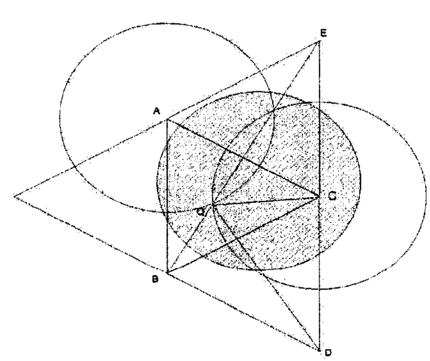

Figure 14: If $1 < d \leq \sqrt{3}$ then the three closest lattice discs must be in any solution.

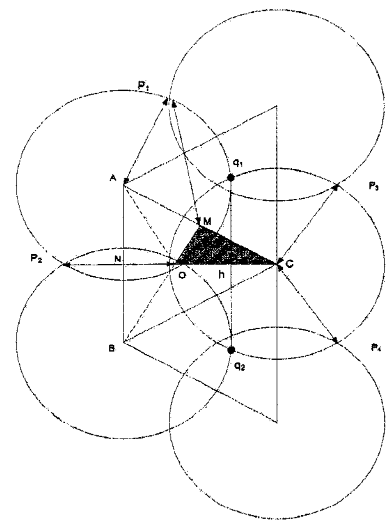

Figure 15: Five lattice discs are necessary when $1 < d \leq \sqrt{3}$.

Necessary condition

Let Φ be an arbitrary disc with center located inside the triangle ABC (see Figure 14). We claim that any lattice disc cover of Φ must contain the lattice discs centered at A, B, and C. Suppose instead that there is a lattice disc cover that contains only the lattice discs centered at A and C (for example). Then consider the point Q which is the intersection of the discs centered at A and C lying inside $\triangle ABC$. This point (and a small area around it not contained in the discs centered at A and C) must be contained in the disc Φ. From Table 5 in Appendix A (items 1 and 2), it can be seen that for $1 < d \leq \sqrt{3}$, it is at distance greater than 1 from lattice discs centered at D and E and all other lattice discs other than the one centered at B are even further away. Thus any lattice disc cover for Φ must contain all three lattice discs centered at A, B, and C. The cases when the lattice disc cover does not contain A or C can be argued similarly.

Now we prove that there exists a disc which cannot be covered with 4 lattice discs, and therefore at least five lattice discs will be needed. Consider the specific disc Φ centered at

h, the intersection of line segment $\overline{OC}$ and $\overline{q_1q_2}$, where o is the circumcenter of $\triangle ABC$, q_1 (respectively q_2) is the intersection of the lattice discs A and C (respectively B and C) falling outside $\triangle ABC$ (see Figure 15) . Since $\|C - q_1\| = 1$, $\|h - q_1\| < 1$. This shows that there is a region around q_1 which cannot be covered by discs A or C, and similarly a region around q_2 which cannot be covered by discs B or C. It is not difficult to see that two more lattice discs are needed to cover these regions. This completes the proof of the necessary condition for this case.

Sufficient condition

To prove the sufficient condition, we show that any disc Φ in the plane can be completely covered with at most 5 lattice discs when $1 < d \leq \sqrt{3}$. Consider a disc Φ with center c in the triangle ABC of the triangular lattice, as shown in Figure 15. Let T_1, T_2, T_3 be the three triangles that share an edge with triangle ABC. We can consider the triangle ABC to be divided into six similar sub-triangles as shown in Figure 15. If the center of Φ lies in the sub-triangle OMC, where O is the circum-center of the triangle ABC, and M is the midpoint of $\overline{AC}$, then we claim that the five lattice discs that cover Φ are centered at A, B, C, and the third vertex of the two triangles in T_1, T_2, and T_3, that include C. The cases where the center of Φ lies in one of the other sub-triangles are identical and will not be discussed further here. To show that the five lattice discs given above completely cover the disc Φ, we argue that the distance from each of the three points O, M, C to the outside of the area covered by the five discs is at least 1. Let R be the closed curve containing the union of the five lattice discs. It is easy to see from Figure 15 that C, being the center of a lattice disc, is at distance 1 from R, and that the closest points from M to the boundary R are P_1 and P_3. It is straightforward to see that $\|M - P_1\| > \|A - P_1\| = 1$. Similarly,

$\|M - P_3\| > \|C - P_3\| = 1$. Finally, we consider point O, for which the closest point to the boundary is P_2. It is straightforward to verify from Table 5 that $\|O - P_2\| \geq 1$ for $1 < d \leq \sqrt{3}$. This concludes the proof that the five given lattice discs completely cover the disc Φ.

4.3 Case 3: $\frac{2\sqrt{7}}{7} < d \leq 1$

When $\frac{2\sqrt{7}}{7} < d \leq 1$, we show that four discs are necessary and sufficient to cover an arbitrary disc which is not co-located with a lattice point.

Necessary Condition

Consider the specific disc Φ centered at M in Figure 16, where M is the midpoint of $\overline{AB}$. In Figure 16, j and k are the two points on $\overline{ST}$ at distance 1 from M. Similary, i and z are also the two points at distance 1 from M, which are on the perpendicular bisector of $\overline{ST}$. Let Ψ be the set of fourteen lattice points shown in Figure 16.

Claim 2 *1. Any lattice-based disc cover of Φ must include a lattice disc centered in the set $\{A, B, C, E\}$.*

2. There exists an optimal disc cover consisting solely of discs centered in the set Ψ.

Proof. To see (1) observe that the lattice discs centered at $\{A, B, C, E\}$ are the only lattice discs that cover the point M for $d > 2/\sqrt{7}$. To see (2), notice that for any lattice disc with center $\alpha \notin \Psi$, there exists a disc with center $\beta \in \Psi$ such that $disc(\alpha) \cap \Phi \subseteq disc(\beta) \cap \Phi$. □

It follows from the above claim that to prove the necessary condition, it is enough to consider 3-subsets of Ψ that contain either A or B (or both) and those that contain C or D (but not A or B). In each case we show using a proof by contradiction that it would be impossible to cover Φ. Suppose $U \subseteq \Psi$ is a set of three lattice points such that the lattice discs centered at them covers disc Φ. For convenience, we say U covers Φ.

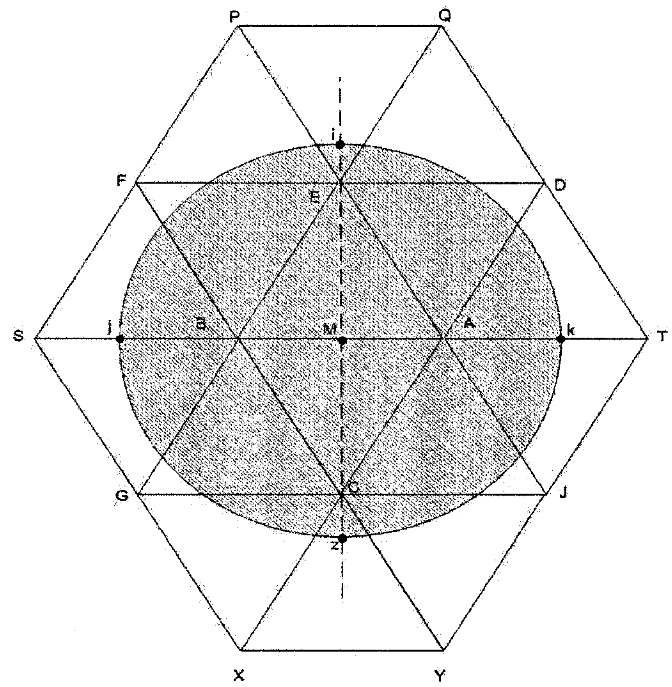

Figure 16: The fourteen closest lattice points to M, which is the midpoint of $\overline{AB}$

- **Case when** $(A \in U)$ **or** $(B \in U)$

Assume that A is included in U (the case when $B \in U$ is symmetric and the proof is identical and omitted). We systematically consider the possibilities for the second disc in U. First notice that if the second disk is any lattice disc with center on the line segment $\overline{ST}$ then neither of the first two discs can cover either of the points i and z. Since $\|i - z\| = 2$, no single third lattice disc can cover both of them. This means none of the lattice discs centered at S, B, or T can be part of a three-disc cover U that contains A. Next we consider all points in Ψ *above* the line $\overline{ST}$ to be the second disc in U. The cases when the second disc in U is below the line $\overline{ST}$ are symmetric and not considered here. For each of the possibilities of the second disc, we show that no matter what is chosen as the third disc, U cannot cover Φ completely.

$A \in U$ and $Q \in U$: Observe that if any of the discs with center on or above the line segment $\overline{ST}$ (*i.e.* P, F, E, D, T, B, or S) is the third disc in U, then the point z cannot be covered, regardless of the value of d, simply because $\|M - z\| = 1$, and M is closer to z than any of the discs in this set. Similarly, if the third disc is centered on or to the right of the line $\overline{iz}$, then the point j cannot be covered. This leaves only discs centered at G or X as candidates for the third disc. Suppose $U = \{G, A, Q\}$. It follows that the discs centered at G and Q must have a non-empty intersection. However, this implies that $\|G - Q\| = 3d < 2$, or $d < 2/3$. This contradicts the assumption that $d > 2/\sqrt{7}$. Since $\|X - Q\| > \|G - Q\|$, it follows that X cannot be the third disc in U as well. Thus, if $A \in U$ and $Q \in U$ (and similarly, if $A \in U$ and $Y \in U$), then U cannot cover the disc Φ.

$A \in U$ and $D \in U$: Using exactly similar arguments to the previous case, we can see that all lattice points in Ψ except for G and X are excluded from being centers for the third disc. But D, A, and X are collinear, which means U cannot cover Φ. It remains to consider the case $U = \{A, D, G\}$. But since G, M, and D are collinear, there is a point at distance 1 from M on the line perpendicular to $\overline{DG}$ which is not covered by any of the discs in U. Thus, if $A \in U$ and $D \in U$ (and similarly if $A \in U$ and $J \in U$), then U cannot cover the disc Φ.

$A \in U$ and $F \in U$: After excluding candidates on and above the line $\overline{ST}$ and the discs centered at J and Y, for which we have just provided arguments for exclusion, the only candidates for the third disc in U are the discs centered at G, X, and C. Let q be the

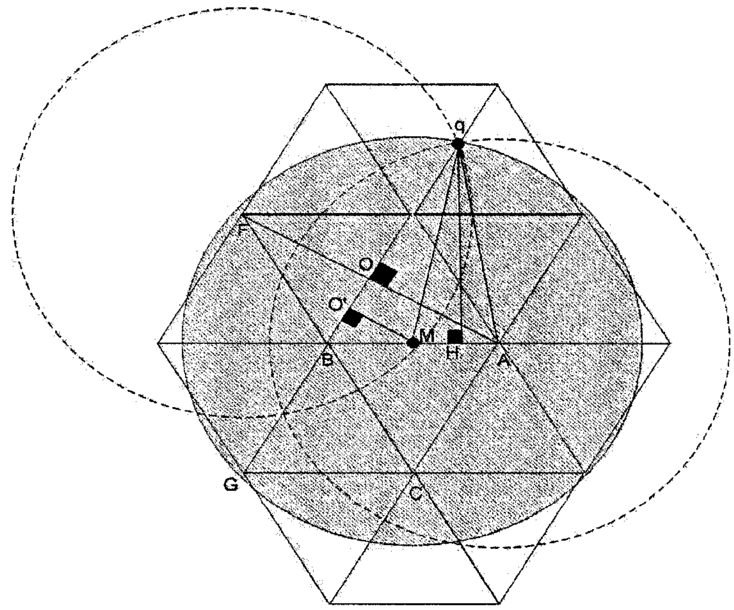

Figure 17: The case when $A \in U$ and $F \in U$. The point q is not covered by any of the discs centered at G, C, or X, yet is at distance < 1 from M.

intersection of discs centered at A and F such that the disc centered at G does not contain Q (see Figure 17). It is easy to verify that q is also not contained in the discs centered at X or C. From Table 5, it is easy to verify that when $d > 2/\sqrt{7}$, then $\|M - q\| < 1$, which implies that there is an area of the disc Φ which is not covered by the lattice discs with centers at any of A, F, G, C, X. This shows that if $A \in U$ and $F \in U$, (and similarly if $A \in U$ and $G \in U$) then U cannot cover Φ.

$A \in U$ **and** $P \in U$**:** After excluding candidates on and above the line $\overline{ST}$ and the discs centered at J, Y, and G, for which we have just provided arguments for exclusion, the only candidates for the third disc in U are the discs centered at X and C. But from Table 5, it is easy to verify that $\|P - j\| = \|X - j\| > 1$, for $d > 2/\sqrt{7}$. Further clearly $|C - j| > |M - j| = 1$. Thus none of the discs centered at A, P, X, C can cover the point j. This shows that if $A \in U$ and $P \in U$ (and similarly if $A \in U$ and $X \in U$), then U cannot cover Φ.

$A \in U$ **and** $E \in U$**:** The only remaining candidate for the third disc is the disc centered at E. However, since C and E are collinear with M, it is clear that $U = \{A, C, E\}$ cannot cover the point $j \in M$.

We have shown that if $A \in U$, then U cannot cover Φ. Symmetrical arguments apply to $B \in U$ as well. We go on to consider the cases excluding both these lattice discs.

- **Case when** $(A \notin U)$ **and** $(B \notin U)$ **and** $(C \in U$ **or** $E \in U)$

Assume that U is a three-disc cover for Φ that includes the disc centered at C (the case when $E \in U$ is symmetrical and omitted). As before, we consider the possibilities for the second disc in U. First, observe that if $E \in U$ as well, since neither C nor E covers either of the points j and k, and $\|j - k\| = 2$, no third lattice disc can cover both of them. We consider all the elements to the right of the line $\overline{CE}$ to be candidates for the second disc and show that in each case, U cannot cover Φ. The arguments for the elements of Ψ to the left of the line $\overline{CE}$ are symmetrical and will be omitted.

$C \in U$ **and** $Q \in U$**:** Observe that if any of the discs with center on or to the right of the line EC (D, T, J, Y) is the third disc in U, then the point j cannot be covered, regardless of the value of d, simply because $\|M - j\| = 1$, and M is closer to j than any of the discs in this set. However, if the third disc is centered to the left of the line CE, then the point k cannot be covered since $\|M - k\| = 1$, and M is closer to j than any of the discs in this set, and $\|Q - k\| = \|X - j\| > 1$ for $d > 2/\sqrt{7}$ as can be verified from Table 5. Thus, if $C \in U$ and $Q \in U$ (or if $C \in U$ and $P \in U$), then U cannot cover Φ.

$C \in U$ **and** $D \in U$**:** After excluding all lattice points in Ψ on or to the right of line $\overline{EC}$, as well as the point P for which we just provided arguments for exclusion, the only candidates remaining for the third element in U are F, S, G, and X. Let q be the intersection of discs centered at C and D that is nearest to the point k (see Figure 18). It is not difficult to verify that the discs centered at F, S, G, and X do not contain the point q. On the other hand, from Table 5, it is straightforward to verify that $\|M - Q\| < 1$ when $d > 2/\sqrt{7}$ which shows that there is an area of Φ which is not covered by any of the discs with centers in $\{C, D, S, G, F, X\}$. Thus, when $C \in U$ and $D \in U$ (or when $C \in U$ and $F \in U$), then U does not cover Φ.

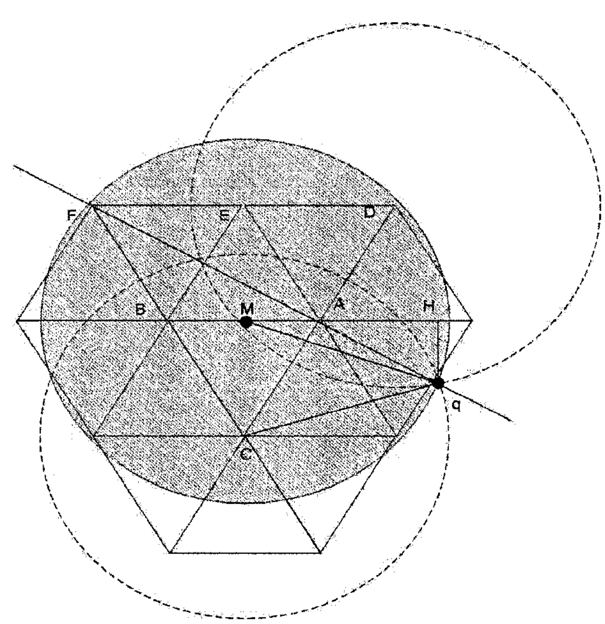

Figure 18: The case when $C \in U$ and $D \in U$. The point q is not covered by any of the discs centered at F, S, G, or X, yet is at distance < 1 from M.

$C \in U$ **and (** $T \in U$ **or** $J \in U$ **or** $Y \in U$ **):** We have already shown that any three-disc cover for Φ that includes the disc centered at C cannot include discs centered at P, Q, D, E or F. This means that the only possible lattice discs in U are all on or below the line $\overline{ST}$, which means that U cannot cover the point i.

Thus, no three disc-cover can include discs centered at C or E.

To reiterate, we first showed (Claim 2) that any cover for Φ must include a disc centered at $\{A, B, C, D\}$. We then showed that if U is a set of three lattice discs that contains A or B then U cannot cover Φ. Finally we showed that if U is set of three lattice discs that does not contain A or B, but does contain C or E, then U cannot cover Φ. We conclude that there is no set of three lattice discs that can cover Φ, which completes the proof of the necessary condition.

Sufficient Condition

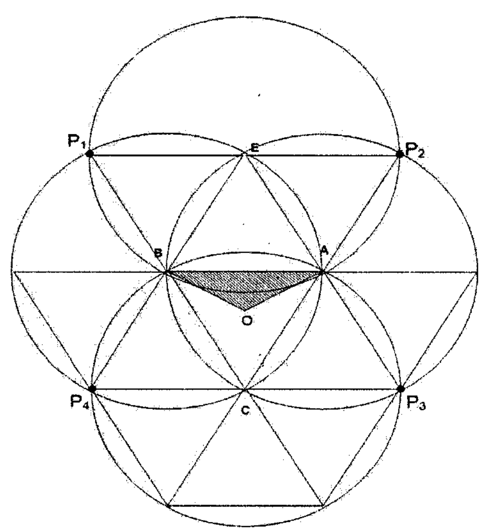

Figure 19: Four lattice discs A, B, C and E are sufficient to cover any disc Φ centered inside $\triangle ABO$

To prove the sufficient condition, we show that any disc Φ in the plane can be completely covered with at most 4 lattice discs when $\frac{2}{\sqrt{7}} < d \leq 1$. Consider a disc Φ with center c in the triangle ABC, as shown in Figure 19. We can consider the triangle ABC to be divided into three similar sub-triangles as shown in Figure 19. If the center of Φ lies in the sub-triangle OAB, where O is the circumcenter of the triangle ABC, then we claim that the four lattice discs that cover Φ are centered at A, B, C, and the third vertex of the other triangle that has side $\overline{AB}$ (here the fourth lattice point is E). The cases where the center of Φ lies in one of the other sub-triangles are identical and will not be discussed further here. To show that the four lattice discs given above completely cover the disc Φ, we argue that the distance from each of the three points O, A, B to the outside of the area covered by the four discs is at least 1. Let R be the closed curve containing the union of the four lattice discs. It is easy to see from Figure 19 that B, being the center of a lattice disc, is at distance 1 from P_1 and P_4. A is also at distance 1 from P_2 and P_3. Since $\angle OAP_3$ is a right angle as is $\angle OBP_4$, we have $||O - P_3|| > 1$ and $||O - P_4|| > 1$. This concludes the proof that the four given lattice discs completely cover the disc Φ.

4.4 Case 4: $d \leq 2/\sqrt{7}$

The necessary condition here follows from Lemma 11, therefore, in this section, we prove only the sufficient condition, that is, that any disc in the plane can be completely covered with at most 3 lattice discs when $d \leq \frac{2}{\sqrt{7}}$. Consider a disc Φ with center c in the triangle ABC, as shown in Figure 20. We can consider the triangle ABC to be divided into six similar sub-triangles as shown in the above figure. If the center of Φ lies in the sub-triangle AMO, where O is the circum-center of the triangle ABC, then we claim that one of the lattice disc sets U_1, U_2, and U_3 is a cover for Φ, where $U_1 = \{A, C, F\}$, $U_2 = \{E, G, J\}$, and $U_3 = \{B, D, J\}$. The cases where the center of Φ lies in one of the other sub-triangles are identical and will not be discussed further here.

To show that the three lattice discs given above completely cover the disc Φ, we argue that the distance from any possible center c inside $\triangle AMO$ to the outside of the area covered by one of the three-disc sets mentioned above, is at least 1. In other words we prove that for any point c (the center of Φ inside $\triangle AMO$), at least one the following conditions is true:

Condition 1: $(\|c - P_1\| \geq 1)$ and $(\|c - Q_1\| \geq 1)$ and $(\|c - R_1\| \geq 1)$

In this case lattice discs are centered at A, C, and F (see Figure 20).

Condition 2: $(\|c - P_2\| \geq 1)$ and $(\|c - Q_2\| \geq 1)$ and $(\|c - R_2\| \geq 1)$

In this case lattice discs are centered at E, G, and J (see Figure 21) .

Condition 3: $(\|c - P_3\| \geq 1)$ and $(\|c - Q_3\| \geq 1)$ and $(\|c - R_3\| \geq 1)$

In this case lattice discs are centered at B, D, and J (see Figure 22).

In Figure 20, the three discs centered at A, C, and F are shown as dashed circles. P_1 is the intersection of the discs centered at F and A such that disc centered at C does not contain P_1. Similarly Q_1 (R_1) is the intersection of the discs centered at F and C (respectively C and A) such that disc centered at A (respectively F) does not contain Q_1 (respectively R_1). The intersection points Q_2, R_2, P_3, Q_3, and R_3 are defined in a similar way (see Figures 21 and 22). In all these three figures a grey disc of radius one is pictured centered at these intersection points.

Suppose Condition 1 is false. Since $\|A - R_1\| = 1$ (see Figure 20), we have $\|AO - R_1\| \geq 1$. Therefore, it must be that $\|c - R_1\|$ is always at least 1, and since Condition 1 is false, either $\|c - P_1\| < 1$ or $\|c - Q_1| < 1$. The remainder of the proof consists of showing that if $\|c - P_1\| < 1$, then Condition 3 must be true, and if $\|c - Q_1\| < 1$, then Condition 2 must be true.

Suppose $\|c - P_1\| < 1$. Then we claim that Condition 3 is true. From $\triangle ADP_3$ (see

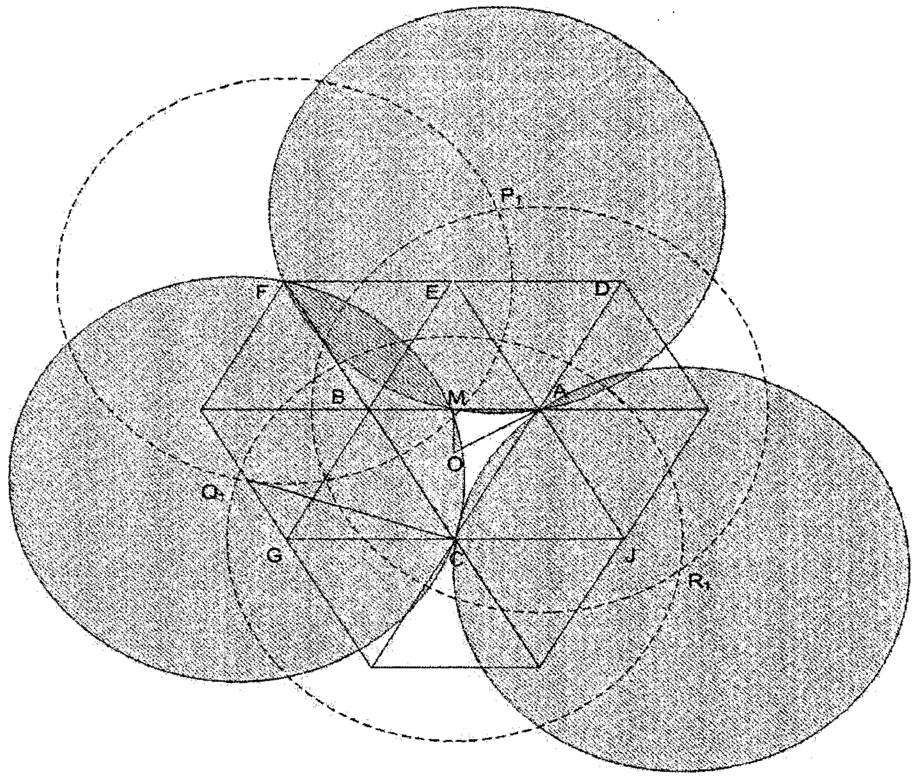

Figure 20: The set $U_1 = \{A, C, F\}$ and the intersection points P_1, Q_1, R_1.

Figure 22), since $\angle DAP_3$ is $\pi/3$ and $\angle ADP_3 > \pi/3$ for $d < 2/\sqrt{7}$, it follows that $\overline{AP_3}$ is the longest side, which means $\overline{AP_3} > \overline{DP_3} = 1$. Therefore $\|c - P_3\| \geq 1$ regardless of the place of c inside $\triangle AMO$. For $d \leq 2/\sqrt{7}$ it can be verified from Table 5 that $\|M - Q_3\| \geq 1$, so the disc centered at Q_3 cuts $\overline{AB}$ at the left of M; this means that $\left\|\overline{AM} - Q_3\right\| \geq 1$ and therefore $\|c - Q_3\| \geq 1$ regardless of the place of c inside $\triangle AMO$. It remains to show that $\|c - R_3\| \geq 1$. But it can be verified from Table 5 that $\|R_3 - P_1\| = 2\,\|R_3 - M\| \geq 2$ when $d < 2/\sqrt{7}$. Since $\|c - P_1\| < 1$ by assumption, it follows that $\|c - R_3\| \geq 1$. This shows that Condition 3 is true.

Suppose $\|c - Q_1\| < 1$. Then we claim that Condition 2 is true. From $\triangle GCR_2$ and $\triangle GCO$ we have $\|R_2 - O\|^2 = \frac{2d}{\sqrt{3}} + \sqrt{1 - d^2}$ (see Figure 21). It is straightforward to verify

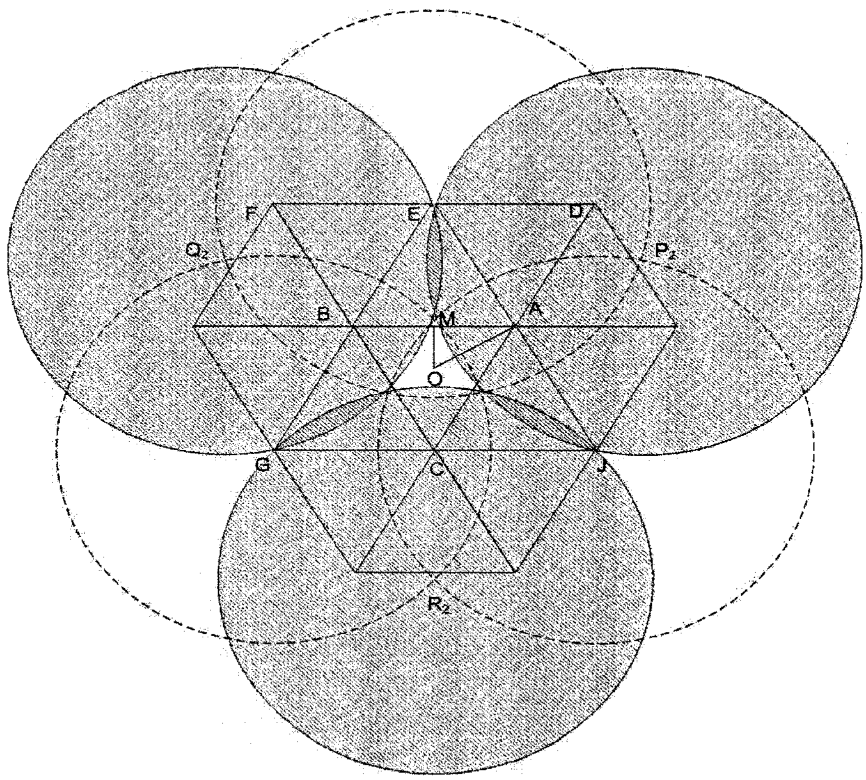

Figure 21: The set $U_2 = \{E, G, J\}$ and the intersection points P_2, Q_2, R_2.

that when $d \leq 2/\sqrt{7}$ we get $\|R_2 - O\| \geq 1$. In addition, for $d \leq 2/\sqrt{7}$ it can be verified from Table 5 that $\|M - Q_2\| = \|M - P_2\| \geq 1$, so the disc centered at Q_2 cuts $\overline{AB}$ at the left of M; this means that $\left\|\overline{AM} - Q_2\right\| \geq 1$ therefore ($\|c - Q_2\| \geq 1$) regardless of the place of c inside $\triangle AMO$. It remains to show that $\|c - P_2\| \geq 1$. But it can be verified easily from Table 5 that $\|P_2 - Q_1\| = 2\,\|P_2 - M\| \geq 2$ when $d < 2/\sqrt{7}$. Since $\|c - Q_1\| < 1$, it follows that $\|c - P_2\| \geq 1$. This shows that Condition 2 is true.

We have shown that three lattice discs are always sufficient to cover a disc centered at any point in $\triangle AMO$. The argument for the other sub-triangles is symmetric. This completes the proof of the sufficient condition for Case 4. $\square$

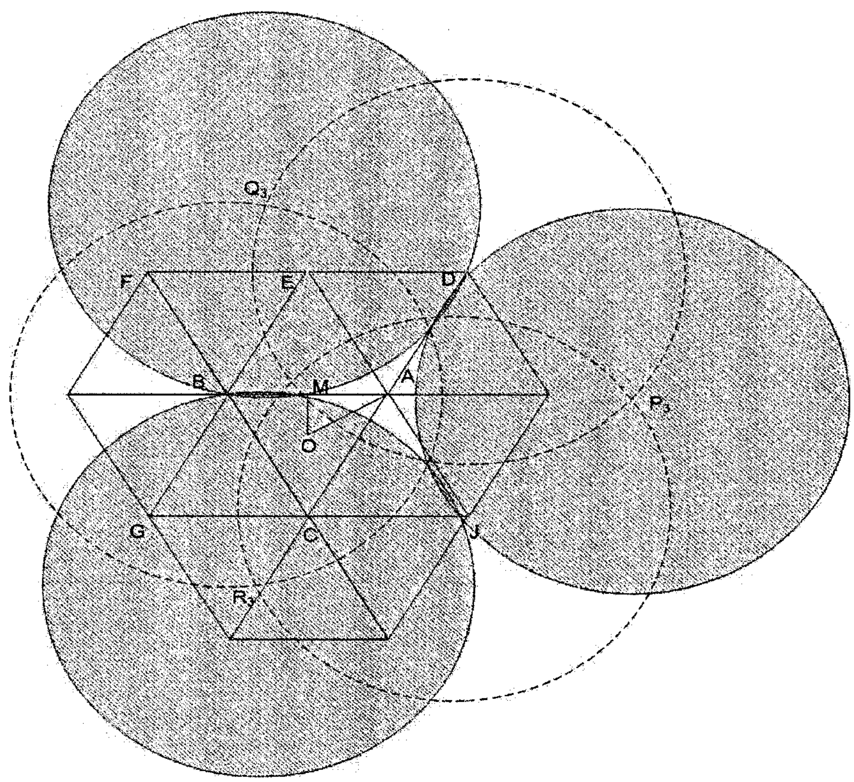

Figure 22: The set $U_3 = \{B, D, J\}$ and the intersection points P_3, Q_3, R_3.

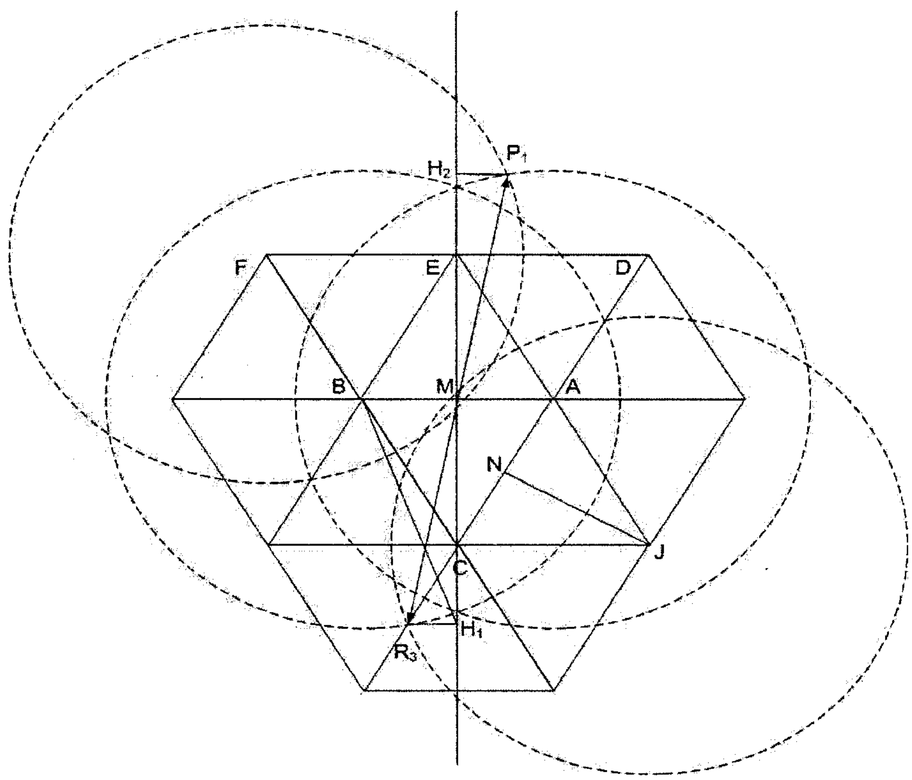

Figure 23: The distance $\|R_3 - P_1\|$.

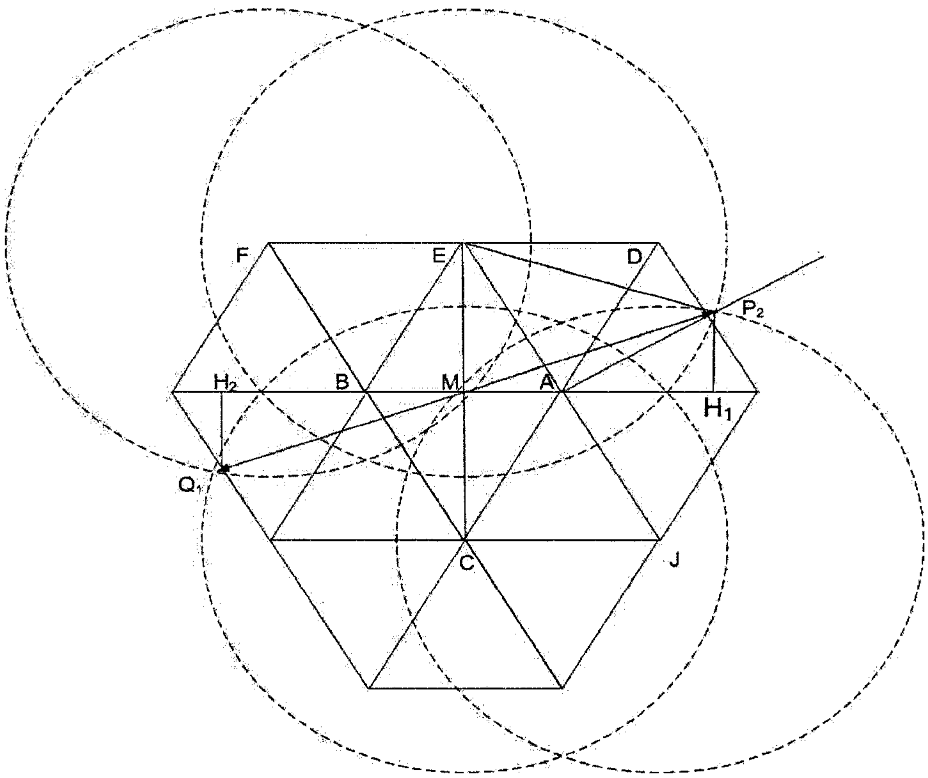

Figure 24: The distance $\|P_2 - Q_1\|$.

Chapter 5

Conclusions

We studied the problem of multiple sink positioning in wireless sensor networks. This problem can be understood as finding the stop points for a mobile sink in such a network. Using multiple or mobile sinks, it is possible to limit the number of hops that a packet may travel inside a network, therefore it is possible to reduce the error rate and latency and provide a longer lifetime for the network by wasting less energy on multi hop routing and providing a better load balance in the network. We modelled the problem of multiple sink positioning as the h-hop covering set problem for unit disk graphs. We defined the *h-hop covering set problem* as follows: given a unit disk graph $G = (V, E)$, where V represents the set of sensor node locations, find the minimum-sized set of points U (sink node locations) such that every node in V can reach a point in U using at most h hops.

In this work, we gave the first PTAS for the h-hop covering set problem. We proposed two different approximation algorithms with running times $O(\ell^2 max(\lceil \ell\sqrt{2} \rceil^2, n) n^{2\lceil \ell\sqrt{2} \rceil^2+1})$ and
$O(\ell^2 max(\lceil h\ell\sqrt{2} \rceil^2, n) n^{2\lceil h\ell\sqrt{2} \rceil^2+1})$, and performance ratios of $(1+\frac{h}{\ell})^2$ or $(1+\frac{1}{\ell})^2$ respectively. Both of these algorithms are based on the idea of shifting strategy originally described in

[13].

In addition, we gave a new PTAS for the lattice-based disc cover problem, based on a new approach deriving from recent results on dominating sets in unit disc graphs [24]. We generalized this result to give the first PTAS for the h-hop lattice-based covering set problem. We showed that this in turn yields an algorithm with a $3(1+\epsilon)$ performance ratio for the h-hop covering set problem with a running time of $O(cn^2)$ where c is a function of ϵ and h. We showed that this algorithm can be implemented in a distributed manner giving the first distributed algorithm for the h-hop covering set problem. The distributed solution, yields the same performance ratio of $3(1+\epsilon)$, and has a quadratic running time.

The approximation ratio of $3(1+\epsilon)$ for the h-hop covering set problem, which is based on a lattice-based approach for disc cover problem, is proved using a geometric theorem, in which we characterize the relationship between the side of a triangular lattice and the number of lattice discs that can cover an arbitrary disc on the plane.

Finally, we gave the first PTAS for the h-hop graph-based disc cover set problem, which is based on recent results on dominating sets in unit disc graphs [24]. This algorithm can also be implemented in a distributed and local manner. This algorithm delivers a $5(1+\epsilon)$ performance ratio for the h-hop covering set problem with running time of $O(n^{c^2})$, where c is a constant. Although the running time of this algorithm is not better than the lattice-based approach, it has the advantage of not requiring a geometric representation for the unit disc graph.

The theoretical result which is obtained in this work is to position and find the number of multiple sinks in a wireless sensor network, which enables every sensor nodes to reach a sink within h hops for some specified value of h. We provided distributed implementations for some of our algorithms, but the message complexity of these algorithms remains to be analyzed. Another important future direction of this work is designing and analyzing protocols which can use these algorithms. Specifically, in the case that we restrict the number of hops in a multi hop routing algorithm, we should design and verify the protocol which take cares of the communication and data delivery in such a network. For example, restricted flooding is a simple approach in which each of the multiple sinks can identify the sensor nodes which are within h hops around them. Details of such a protocol are beyond the scope of this thesis, but it is worth saying that the main issues in such protocols in different layers are message synchronization, finding the nearest sink, handling sleep mode, centralized and distributed administration of the network, and efficient multi hop routing.

It would be interesting to test our algorithms on networks with different parameters of real world applications. Since the algorithms provided in this work are all about using multiple or mobile sinks to increase the lifetime of the network, one of the future directions of this work is to simulate the networks with such elements. The parameters which are variable are the number of nodes, the maximum number of hops in a multi hop routing scenario, the size of the area which the network is scattered on it, and the speed of a mobile sink if it is applicable. The performance criterion to be evaluated is the lifetime of the network in different situations.

Finally, another important future aspect of this work is the mobile sink trajectory and motion control. For a mobile sink whose stop points are known, it is necessary to find the route that it should travel between the stop points. The time for each stop is another issue which affects the amount of data collected at each stop, and the latency of the network in general.

Bibliography

[1] L. Venkatraman A. Keshavarzian, H. Lee. Wakeup scheduling in wireless sensor networks. In *MobiHoc '06: Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 322–333, New York, NY, USA, 2006. ACM.

[2] Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994.

[3] H. Bronninamm and T. Goodrich. Almost optimal set covers in finite VC-dimension. In *SCG '94: Proceedings of the Tenth Annual Symposium on Computational Geometry*, pages 293–302, 1994.

[4] Jehoshua Bruck, Jie Gao, and Anxiao (Andrew) Jiang. Map: Medial axis based geometric routing in sensor networks. In *MobiCom '05: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, pages 88–102, 2005.

[5] Edgar H. Callaway. *Wireless Sensor Networks, Architectures and Protocols.* AUERBACH Press, 2004.

[6] R. Cole and U. Vishkin. Deterministic coin tossing with applications to optimal parallel list rankin. *Information and Control*, 70:32–53, 1986.

[7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition.* The MIT Press, September 2001.

[8] T. Feder and D. Greene. Optimal algorithms for approximate clustering. In *STOC '88: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 434–444, 1988.

[9] S.R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *IEEE GLOBECOM '04: Proceedings of Global Communications Conference Exhibition and Industry Forum*, pages 377–381, 2003.

[10] M. R. Garey and D.S. Johnson. *Computers and Interactability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1978.

[11] T. Gonzalez. Covering a set of points in multi-dimensional space. *Information Processing Letters*, 40(4):133–137, 1991.

[12] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, pages 8–20, 2000.

[13] Dorit S. Houchbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *JACM*, 32(1):130–136, 1985.

[14] M. A. Wong J. A. Hartigan. A k-means clustering algorithm. *J. Applied Statistics*, 28:100–108, 1997.

[15] David S. Johnson. The np-completeness column: An ongoing guide. *J. Algorithms*, 8(5):438–448, 1987.

[16] Aman Kansal, Arun A. Somasundara, David D. Jea, Mani B. Srivastava, and Deborah Estrin. Intelligent fluid infrastructure for embedded networks. In *MobiSys '04: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, pages 111–124, 2004.

[17] T. Kohonen. *Self-organizing Maps*. Springer-Verlag Press, New York, 1995.

[18] F. Kuhn, T. Moscibroda, T. Nieberg, and Wattenhofer. Fast deterministic distributed maximal independent set computation on growth-bounded graphs. In *DISC '05: 19th International Symposium on Distributed Computing*, pages 26–29. Springer-Verlag, 2005.

[19] Fabian Kuhn, Thomas Moscibroda, Tim Nieberg, and Roger Wattenhofer. Local approximation schemes for ad hoc and sensor networks. In *DIALM-POMC '05: Proceedings of the 2005 Joint Workshop on Foundations of Mobile Computing*, pages 97–103, 2005.

[20] J. Luo and P. Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. In *IEEE INFOCOM '05: Proceedings of 274th Conference on Computer Communications*, 2005.

[21] M. and J. Bruck. A geometric theorem for network design. *IEEE Transactions on Computers*, 53(4):483–489, April 2004.

[22] Xu Hui Jeong Byeong-Soo Lee Sungyoung Lee Young-Koo Meng Min, Wu Xiaoling. Energy efficient routing in multiple sink sensor networks. In *ICCSA '07: Proceedings of International Conference on Computational Science and Its Applications*, 2007.

[23] L. Narayanan and Y. Tang. Worst-case analysis of a dynamic channel assignment strategy. *Discrete Applied Mathematics*, 140(1–3):115–141, 2004.

[24] T. Nieberg and J. L. Hurink. A PTAS for the minimum dominating set problem in unit disk graphs. In T. Erlebach and G. Persiano, editors, *3rd International Workshop on Approximation and Online Algorithms, WAOA 2005, Palma de Mallorca, Spain*, volume 3879 of *Lecture Notes in Computer Science*, pages 296–306, 2006.

[25] T. Nieberg, J. L. Hurink, and W. Kern. A robust PTAS for maximum weight independent sets in unit disk graphs. In *Graph-Theoretic Concepts in Computer Science: 30th International Workshop, WG 2004*, volume 3879, pages 296–306, 2004.

[26] Eylem Ilker Oyman. Phd thesis, bogacizi university: Multiple sink location problem and energy efficiency in large scale wireless sensor networks. 2004.

[27] R. Shah, S. Roy, S. Jain, and W. Brunette. Date mules: Modeling a three-tier architecture for sparse sensor networks. In *IEEE Workshop on Sensor Network Protocols and Applications (SNPA)*, pages 30–41, 2003.

[28] A. Durresi R. Kannan V. Paruchuri, S. Basavaraju and S. S. Iyengar. Random asynchronous wakeup protocol for sensor networks. In *BROADNETS '04: Proceedings of the First International Conference on Broadband Networks*, pages 710–717, Washington, DC, USA, 2004. IEEE Computer Society.

[29] Wei Wang, Vikram Srinivasan, and Kee-Chaing Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. In *MobiCom '05: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, pages 270–283, 2005.

[30] Z.M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli. Exploiting sink mobility for maximizing sensor networks lifetime. In *HICSS '05: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 2005.

[31] N.H Viadya X. Yang. A wakeup scheme for sensor networks: achieving balance between energy saving and end-to-end delay. In *RTAS '04:In Proceeding of Real-Time and Embedded Tecnology and Applications Symposium*, pages 19 – 26, 2004.

[32] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 187–198, 2004.

Appendix A

Theorem 9 *The following table gives distances between points on the lattice.*

	Distance	Figure	
1	$\|E-Q\|$	$\frac{d\sqrt{3}}{2}+\frac{\sqrt{4-d^2}}{2}$	Figure 14
2	$\|D-Q\|$	$\sqrt{1+\frac{3d^2}{2}-\frac{d\sqrt{12-3d^2}}{2}}$	Figure 14
3	$\|O-P_2\|$	$\frac{\sqrt{4-d^2}}{2}+\frac{d\sqrt{3}}{6}$	Figure 15
4	$\|X-j\|$	$\sqrt{1+3d^2+d^2/4-d}$	Figure 16
5	$\|M-Q\|$	$\sqrt{\frac{4-2d^2+d\sqrt{4-3d^2}}{4}}$	Figure 17
6	$\|M-Q\|$	$\sqrt{1-\frac{3d^2}{4}+\frac{d\sqrt{3-3d^2}}{2}}$	Figure 18
7	$\|M-R_3\|$	$(1-\frac{d^2}{4}+\frac{d\sqrt{4-3d^2}}{4})^{\frac{1}{2}}$	Figure 23
8	$\|M-P_2\|$	$(1-\frac{3}{4}d^2+\frac{\sqrt{3}}{2}d\sqrt{1-d^2})^{\frac{1}{2}}$	Figure 24

Table 5: Useful distances in the lattice

Proof.

1. In Figure 14, since we have $\overline{EM}=\frac{d\sqrt{3}}{2}$ and $\overline{MQ}=\sqrt{\frac{\sqrt{4-d^2}}{2}}$, then we get $\|E-Q\| = \frac{d\sqrt{3}}{2}+\sqrt{\frac{\sqrt{4-d^2}}{2}}$.

2. In Figure 14 we have $\overline{CM} = d/2$ and $\overline{CQ} = 1$, then we get $\overline{MQ} = \frac{\sqrt{4-d^2}}{2}$. Since we have $\overline{MB} = \frac{d\sqrt{3}}{2}$, we obtain $\overline{BQ} = \frac{d\sqrt{3}-\sqrt{4-d^2}}{2}$. Using $\triangle BQD$ we have $\|D - Q\| = \sqrt{1 + \frac{3d^2}{2} - \frac{d\sqrt{12-3d^2}}{2}}$.

3. In Figure 15 we have $\|N - B\| = \frac{d}{2}$. Thus, $\|P_2 - N\| = \sqrt{1 - \frac{d^2}{4}} = \frac{\sqrt{4-d^2}}{2}$. Then, since $\angle OBN = \pi/6$, we have $\|O - N\| = \frac{d\sqrt{3}}{6}$. Finally, $\|O - P_2\| = \|O - N\| + \|N - P_2\| = \frac{\sqrt{4-d^2}}{2} + \frac{d\sqrt{3}}{6}$.

4. In Figure 16 since $\overline{XB} = \sqrt{3}d$ and $\overline{Bj} = 1 - \frac{d}{2}$, then using $\triangle XBj$ we get $\|X - j\| = \sqrt{1 + 13d^2/4 - d}$.

5. In Figure 17, $\overline{AF}$ is the bisector of $\overline{BE}$. In $\triangle AOQ$ we have $\overline{OQ}^2 = 1 - \overline{A0}^2$. Since $\overline{AO} = \sqrt{3}d/2$, we have $\overline{OQ} = \frac{\sqrt{4-3d^2}}{2}$ and $\overline{O'Q} = d/4 + \frac{\sqrt{4-3d^2}}{2}$. Using $\triangle BMO'$ where $\overline{BM} = d/2$ and $\angle MBO' = \frac{\pi}{3}$ and $\overline{BO'} = d/4$ we get $\overline{MO'} = d\sqrt{3}/4$. Next, using $\triangle MQO'$ we find MQ using $\overline{MQ'}^2 = \overline{MO'}^2 + \overline{QO'}^2 = \sqrt{\frac{4-2d^2+d\sqrt{4-3d^2}}{4}}$.

6. In Figure 18, from $\triangle AQC$. we can see that $\overline{AQ} = \sqrt{\overline{CQ}^2 - \overline{AC}^2} = \sqrt{1-d^2}$. Next we consider $\triangle AHQ$ to see that since $\angle HAQ = \pi/6$ and $AQ = \sqrt{1-d^2}$, we get $\overline{AH} = \frac{\sqrt{3}}{2} \times \sqrt{1-d^2}$ and $\overline{HQ} = \left(\frac{\sqrt{1-d^2}}{2}\right)^2$. Finally, since $\overline{MH} = d/2 + \overline{AH}$, using $\triangle MHQ$, we have $\|M - Q\| = \sqrt{\left(\frac{\sqrt{1-d^2}}{2}\right)^2 + \left(\frac{d}{2} + \frac{\sqrt{3}}{2} \times \sqrt{1-d^2}\right)^2}$.

7. In Figure 23 since $\overline{NJ} = \frac{\sqrt{3}d}{2}$, using $\triangle NJR_3$ we get $\overline{NR_3} = \frac{\sqrt{4-3d^2}}{2}$. Therefore $\overline{R_3C} = (\frac{\sqrt{4-3d^2}}{2} - \frac{d}{2})$. Using $\triangle CH_1R_3$, we have $\overline{R_3H_1} = (\frac{\sqrt{4-3d^2}}{4} - \frac{d}{4})$ and $CH_1 = (\frac{\sqrt{3}\times\sqrt{4-3d^2}}{4} - \frac{\sqrt{3}d}{4})$ since $\angle CR_3H_1 = \pi/3$. Having $\overline{CH_1}$ and therefore $\overline{MH_1}$ plus having $\overline{R_3H_1}$ we get $\|R_3 - M\| = (\frac{d^2}{4} + \frac{4-3d^2}{4} + \frac{d\sqrt{4-3d^2}}{4})^{\frac{1}{2}}$.

8. In Figure 24 using $\triangle EP_2A$, since $\overline{EP_2} = 1$ we get $\overline{AP_2} = \sqrt{1-d^2}$. Similary, using $\triangle AP_2H_1$, since $\angle P_2AH_1 = \pi/6$ we have $P_2H_1 = \frac{\sqrt{1-d^2}}{2}$ and $AH_1 = \frac{\sqrt{3}}{2} \times \sqrt{1-d^2}$.

Thus in $\triangle MH_1P_2$ since $AM = d/2$ we get $P_2M = (1 - \frac{3}{4}d^2 + \frac{\sqrt{3}}{2}d(1-d^2))^{\frac{1}{2}}$.

$\square$