Construction and Maintenance of a Mobile Backbone for Wireless Networks

Anand Srinivas, Gil Zussman, and Eytan Modiano
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139
{anand3, gilz, modiano}@mit.edu

ABSTRACT

We study a novel hierarchical wireless networking approach in which some of the nodes are more capable than others. In such networks, the more capable nodes can serve as Mobile Backbone Nodes and provide a backbone over which end-to-end communication can take place. Our approach consists of controlling the mobility of the Backbone Nodes in order to maintain connectivity. We formulate the problem of minimizing the number of backbone nodes and refer to it as the Connected Disk Cover problem. We show that it can be decomposed into the Geometric Disk Cover (GDC) problem and the Steiner Tree Problem with Minimum Number of Steiner Points (STP-MSP). We prove that if these subproblems are solved separately by γ- and δ-approximation algorithms, the approximation ratio of the joint solution is $\gamma+\delta$. Then, we focus on the two subproblems and present a number of distributed approximation algorithms that maintain a solution to the GDC problem under mobility. A new approach to the solution of the STP-MSP is also described. We show that this approach can be extended in order to obtain a joint approximate solution to the Connected Disk Cover problem. Finally, we evaluate the performance of the algorithms via simulation and show that the proposed GDC algorithms perform very well under mobility and that the new approach for the joint solution can significantly reduce the number of required Mobile Backbone Nodes.

Keywords: Wireless networks, Controlled mobility, Distributed algorithms, Approximation algorithms, Disk cover

1. INTRODUCTION

Wireless Sensor Networks (WSNs) and Mobile Ad Hoc Networks (MANETs) can operate without any physical infrastructure (e.g. base stations). Yet, it has been shown that it is sometimes desirable to construct a *virtual backbone* on which most of the multi-hop traffic will be routed [4]. If all nodes have similar communication capabilities and similar limited energy resources, the virtual backbone may pose several challenges. For example, bottleneck formation along the backbone may affect the available bandwidth and the lifetime of the backbone nodes. In addition, the virtual backbone cannot deal with network partitions resulting from the spatial distribution and mobility of the nodes.

Alternatively, if some of the nodes are more capable than others, these nodes can be dedicated to providing a backbone over which reliable end-to-end communication can take place. A novel hierarchical approach for a *Mobile Backbone Network* operating in such a way was recently proposed (e.g. [19],[22], and references therein). In this paper, we develop and analyze novel algorithms for the construction and maintenance (under node mobility) of a Mobile Backbone Network. Our approach is somewhat different from the previous works, since we focus on *controlling the mobility* of the more capable nodes in order to maintain network connectivity and to provide a backbone for reliable communication.

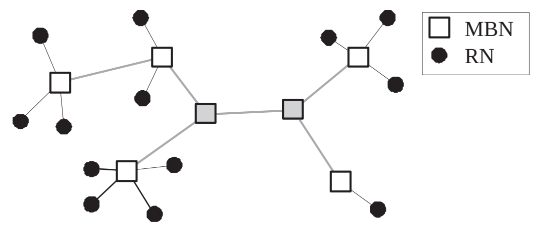

Figure 1: A Mobile Backbone Network in which every Regular Node (RN) can directly communicate with at least one Mobile Backbone Node (MBN). All communication is routed through a connected network formed by the MBNs.

A Mobile Backbone Network is composed of two types of nodes. The first type includes static or mobile nodes (e.g. sensors or MANET nodes) with limited capabilities. We refer to these nodes as *Regular Nodes* (RNs). The second type includes mobile nodes with superior communication, mobility, and computation capabilities as well as greater energy resources (e.g. Unmanned-Aerial-Vehicles). We refer to them as *Mobile Backbone Nodes* (MBNs). The main purpose of the MBNs is to provide a mobile infrastructure facilitating network-wide communication. We specifically focus on minimizing the number of MBNs needed for connectivity. However, the construction of a Mobile Backbone Network can improve other aspects of the network performance, including node lifetime and Quality of Service as well as network reliability and survivability. Note that a Mobile Backbone Network can be tailored to support the operation of both MANETs and WSNs. For example, in a MANET, MBNs should be repositioned in response to RNs mobility. On the other hand, in a static WSN, MBNs could move toward nodes with high requirements or limited energy resources.

Figure 1 illustrates an example of the architecture of a Mobile Backbone Network. The set of MBNs has to be placed such that (i) every RN can directly communicate with at least one MBN, and (ii) the network formed by the MBNs is connected. We assume a *disk* connectivity model, whereby two nodes can communicate if and only if they are within a certain communication range. We also assume that the communication range of the MBNs is significantly larger than the communication range of the RNs.

We term the problem of placing the *minimum* number of MBNs such that both of the above conditions are satisfied as the *Connected Disk Cover (CDC)* problem. While related problems have been studied in the past [2],[4],[11],[13],[21] (see Section 2 for more details), this paper is one of the first attempts to deal with the CDC problem.

Our first approach is based on a framework that *decomposes* the CDC problem into two subproblems. This framework enables us to develop efficient *distributed algorithms* that have good average performance as well as bounded worst case performance. We view the problem as a two-tiered problem. In the first phase, the minimum number of MBNs such that all RNs are *covered* (i.e. all RNs can communicate with at least one MBN) is placed. We refer to these MBNs as *Cover MBNs* and denote them in Figure 1 by white squares. In the second phase, the minimum number of MBNs such that the MBNs' network is connected is placed. We refer to them as *Relay MBNs* and denote them in the figure by gray squares.

In the first phase, the Geometric Disk Cover (GDC) problem [13] has to be solved, while in the second phase, a Steiner Tree Problem with Minimum Number of Steiner Points (STP-MSP) [17] has to be solved. We show that if these subproblems are solved separately by γ- and δ-approximation algorithms[1], the approximation ratio of the joint solution is $\gamma+\delta$.

We then focus on the Geometric Disk Cover (GDC) problem. In the context of static points (i.e. RNs), this problem has been extensively studied in the past (see Section 2). However, much of the previous work is either (i) centralized in nature, (ii) too impractical to implement (in terms of running time), or (iii) has poor average or worst-case performance. Recently, a few attempts to deal with related problems under node mobility have been made [6],[11],[14].

We attempt to develop algorithms that do not fall in any of the categories above. Thus, we develop a number of practically implementable *distributed algorithms* for covering mobile RNs by MBNs. We assume that all nodes can detect their position via GPS or a localization mechanism. This assumption allows us to take advantage of location information in designing distributed algorithms. We obtain the worst case approximation ratios of the developed algorithms and the average case approximation ratios for two of the algorithms. We note that using our analysis methodology, we show that the approximation ratios of algorithms presented in [8] and [11] are lower than the ratios obtained in the past. Finally, we evaluate the performance of the algorithms via simulation, and discuss the tradeoffs between the complexities and and approximation ratios. We show that on average some of algorithms obtain results that are close to optimal.

Regarding the STP-MSP, [17] and [2] propose 3 and 4-approximation algorithms which are based on finding a Minimum Spanning Tree (MST). However, when applied to the STP-MSP, such *MST-based* algorithms may overlook efficient solutions. We present a *Discretization Approach* that can potentially provide improved solutions. In certain practical instances the approach can yield a 2 approximate solution for the STP-MSP.

We extend the Discretization Approach and show that it can obtain a solution to the *joint CDC* problem in a centralized manner. Even for the CDC problem, using this approach enables obtaining a *2-approximation* for specific instances. Due to the continuous nature of the CDC problem, methods such as integer programming cannot yield an optimal solution. Thus, for specific instances this approach provides the lowest known approximation ratio. It is shown via simulation that this is also the case in practical scenarios.

To summarize, our first main contribution is the derivation of a decomposition result regarding the CDC problem. Additional major contributions are the development and analysis of distributed algorithms for the GDC problem in a mobile environment, as well as the design of a novel Discretization Approach for the solution of the STP-MSP and the CDC problem.

This paper is organized as follows. In Section 2 we review related work and in Section 3 we formulate the problem. Section 4 presents the decomposition framework. Distributed approximation algorithms for placing the Cover MBNs are presented in Section 5. A new approach to placing the Relay MBNs is described in Section 6. A joint solution to the CDC problem is discussed in Section 7. In Section 8 we evaluate the algorithms via simulation. We summarize the results and discuss future research directions in Section 9.

2. RELATED WORK

Several problems that are somewhat related to the CDC problem have been studied in the past. For simplicity, when describing these problems we will use our terminology (RNs and MBNs). One such problem is the Connected Dominating Set problem [4]. Unlike the CDC problem, in this problem there is no distinction between the communication ranges of RNs and MBNs. Additionally, MBNs' locations are restricted to RNs' locations. Similarly, the Connected Facility Location problem [21] also restricts potential MBN locations. Furthermore, this problem implies a cost structure that is not directly adaptable to that of the CDC problem. Finally, The Connected Sensor Cover problem [10] involves placing the minimum number of RNs such that they form a connected network, while still covering (i.e. sensing) a specified *area*. This is significantly different from the objective of the CDC problem.

We propose to solve the CDC problem by decomposing it into two NP-Complete subproblems: the Geometric Disk Cover (GDC) problem and the Steiner Tree Problem with Minimum number of Steiner Points (STP-MSP). Hochbaum and Maass [13] provided a Polynomial Time Approximation Scheme (PTAS) for the GDC problem. However, their algorithm is impractical for our purposes, since it is centralized and has a high running time for reasonable approximation ratios. Several other algorithms have been proposed for the GDC problem (see the review in [5]). For example, Gonzalez [8] presented an algorithm based on dividing the plane into strips. In [5] it is indicated that this is an 8-approximation algorithm. We will show that by a simple modification, the approximation ratio is reduced to 6.

Problems related to the GDC problem under node mobility are addressed in [6],[11], and [14]. In [14], a 4-approximate centralized algorithm and a 7-approximate distributed algorithm are presented. Hershberger [11] presents a *centralized*

[1] A γ-approximation algorithm always finds a solution with value at most γ times the value of the optimal solution.

9-approximation algorithm for a slightly different problem: the mobile geometric *square* cover problem. In this paper we build upon his approach in order to develop a *distributed* algorithm for the GDC problem.

Clustering nodes to form a hierarchical architecture has been extensively studied in the context of wireless networks (e.g. [1],[4]). However, the idea of deliberately controlling the motion of specific nodes in order to maintain some desirable network property (e.g. lifetime or connectivity) has been introduced only recently (e.g. [15]).

The algorithm for the STP-MSP proposed in [17] places Relay MBNs along edges of the Minimum Spanning Tree (MST) which connects the Cover MBNs. In [2] it was shown that the approximation ratio of this algorithm is 4, and in addition, a 3-approximation algorithm was proposed. Finally, a 2.5-approximation algorithm was presented in [3].

3. PROBLEM FORMULATION

We consider a set of *Regular Nodes* (RNs) distributed in the plane and assume that a set of *Mobile Backbone Nodes* (MBNs) has to be deployed in the plane. We denote by N the collection of *Regular Nodes* $\{1, 2, \ldots, n\}$, by $M = \{d_1, d_2, \ldots, d_m\}$ the collection of MBNs, and by d_{ij} the distance between nodes i and j. The locations of the RNs are denoted by the $x - y$ tuples (i_x, i_y) $\forall i$.

We assume that the RNs and MBNs have both a communication channel (e.g. for data) and a low-rate control channel. For the communication channel, we assume the disk connectivity model. Namely, an RN i can communicate bi-directionally with another node j (i.e. an MBN) if the distance between i and j, $d_{ij} \leq r$. We denote by $D = 2r$ the diameter of the disk covered by an MBN communicating with RNs. Regarding the MBNs, we assume that MBN i can communicate with MBN j if $d_{ij} \leq R$, where $R > r$. For the control channel, we assume that both RNs and MBNs can communicate over a much longer range than their respective data channels. Since given a fixed transmission power, the communication range is inversely related to data rate, this is a valid assumption.

At this stage, we assume that the number of available MBNs is not bounded (e.g. if required, additional MBNs can be dispatched). Yet, in our analysis, we will try to minimize the number of MBNs that are actually deployed. Finally, we assume that all nodes can detect their position, either via GPS or by a localization mechanism.We shall refer to the problem of Mobile Backbone Placement as the Connected Disk Cover (CDC) problem and define it as follows.

Problem CDC: Given a set of RNs (N) distributed in the plane, place the smallest set of MBNs (M) such that:

1. For every RN $i \in N$, there exists at least one MBN $j \in M$ such that $d_{ij} \leq r$.
2. The undirected graph $G = (M, E)$ imposed on M (i.e. $\forall k, l \in M$, define an edge $(k, l) \in E$ if $d_{kl} \leq R$) is connected.

We will study both the case in which the nodes are static and the case in which the RNs are mobile and some of the MBNs move around in order to maintain a solution the CDC problem. We assume that there exists some sort of MBN routing algorithm, which routes specific MBNs from their old locations to their new ones. The actual development of such an algorithm is beyond the scope of this paper.

Before proceeding, we introduce additional notation required for the presentation and analysis of the proposed solutions. A few of the proposed algorithms operate by dividing the plane into strips. When discussing such algorithms, we assume that the RNs in a strip are ordered from left to right by their x-coordinate and that ties are broken by the RNs' identities (e.g. MAC addresses). Namely, $i < j$, if $i_x < j_x$ or $i_x = j_x$ and the ID of i is lower than ID of j. We note that in property (1) of the CDC problem it is required that every RN is connected to at least one MBN. We assume that even if an RN can connect to multiple MBNs, it is actually assigned to exactly one MBN. Thus, we denote by P_{d_i} the set of RNs connected to MBN d_i. We denote by d_i^L and d_i^R the leftmost and rightmost RNs connected to MBN d_i (their x-coordinates will be denoted by $(d_i^L)_x$ and $(d_i^R)_x$). Similarly to the assumption regarding the RNs, we assume that the MBNs in a strip are ordered left to right by the x-coordinate of their leftmost RN ($(d_i^L)_x$).

In order to evaluate the performance of the *distributed algorithms*, we define the following standard performance measures. We define the *Time Complexity* as the number of communication rounds required in reaction to an RN movement. We assume that during each round a node can exchange errorless control messages with its neighbors. We define the *Local Computation Complexity* as the complexity of the computation that may be performed by a node in reaction to its (or another node's) movement. We assume that the nodes maintain an ordered list of their neighbors. Hence, the Local Computation Complexity refers to the computation required to maintain this list as well as to make algorithmic decisions.

4. DECOMPOSITION FRAMEWORK

In this section we obtain an upper bound on the performance of an approach that solves the CDC problem by decomposing it and solving each of the two subproblem separately. The first subproblem is the problem of placing the minimum number of *Cover MBNs* such that all the RNs are connected to at least one MBN. In other words, all the RNs have to satisfy only property (1) in the CDC problem definition. This problem is the Geometric Disk Cover (GDC) problem [13] which is formulated as follows:

Problem GDC: Given a set N of RNs (points) distributed in the plane, place the smallest set M of Cover MBNs (disks) such that for every RN $i \in N$, there exists at least one MBN $j \in M$ such that $d_{ij} \leq r$.

The second subproblem deals with a situation in which a set of *Cover MBNs* is given and there is a need to place the minimum number of *Relay MBNs* such that the formed network is connected (i.e. satisfying only property (2) in the CDC problem definition). This subproblem is equivalent to the Steiner Tree Problem with Minimum Number of Steiner Points (STP-MSP) [17] and can be formulated as follows:

Problem STP-MSP: Given a set of Cover MBNs (M_{cover}) distributed in the plane, place the smallest set of Relay MBNs (M_{relay}) such that the undirected graph $G = (M, E)$ imposed on $M = M_{cover} \cup M_{relay}$ (i.e. $\forall k, l \in M$, define an edge (k, l) if $d_{kl} \leq R$) is connected.

We now define a *Decomposition Based CDC Algorithm* and bound the worst case performance of such an algorithm.

DEFINITION 1. *A* Decomposition Based CDC Algorithm *solves the CDC problem by using a γ-approximation algorithm for solving the GDC problem, followed by using a δ-approximation algorithm for solving the STP-MSP.*

THEOREM 1. *For $R \geq 2r$, the Decomposition Based CDC Algorithm yields a $(\gamma + \delta)$-approximation for the CDC problem.*

PROOF. Define $ALGO$ as the solution found by solving the CDC problem by the Decomposition Based CDC Algorithm. Also, define $ALGO_{cov}$ and $ALGO_{rel}$ as the set of *Cover* and *Relay* MBNs placed by ALGO. Specifically, an MBN a_i is a *Cover* MBN if it covers at least 1 RN (i.e. $P_{a_i} \neq \emptyset$). Otherwise, a_i is a Relay MBN. Next, define OPT_{CDC} as the overall optimal solution similarly broken up into OPT^{cov}_{CDC} and OPT^{rel}_{CDC}. Thus we have that,

$$\begin{aligned} |ALGO| &= |ALGO_{cov}| + |ALGO_{rel}| \\ &\leq \gamma \cdot |OPT_{cov}| + \delta \cdot |OPT_{ALGO-cov-rel}| \quad (1) \end{aligned}$$

where OPT_{cov} represents the optimal GDC of the RNs, and $OPT_{ALGO-cov-rel}$ represents the optimal STP-MSP solution connecting the Cover MBNs placed by the γ approximate GDC algorithm, $ALGO_{cov}$.

Next, we make use of the fact that a candidate STP-MSP solution given $ALGO_{cov}$ as the input Cover MBNs can be constructed by placing MBNs in the positions defined by those in OPT_{CDC}. The reason this represents a valid STP-MSP solution is that since $ALGO_{cov}$ is a valid GDC for the RNs, it follows that every MBN in $ALGO_{cov}$ is at most a distance r away from *some* RN. Since OPT^{cov}_{CDC} is also a valid GDC, it follows that every MBN in $ALGO_{cov}$ is at most a distance $2r$ from *some* MBN in OPT^{cov}_{CDC}. Therefore, as long as $R \geq 2r$, the MBNs in $ALGO_{cov} \cup OPT_{CDC}$ form a connected network. Finally, since $OPT_{ALGO-cov-rel}$ represents an STP-MSP solution that must be of lower cost than this candidate solution, we have that,

$$\begin{aligned} |ALGO| &\leq \gamma \cdot |OPT_{cov}| + \delta \cdot (|OPT^{cov}_{CDC}| + |OPT^{rel}_{CDC}|) \\ &\leq (\gamma + \delta) \cdot |OPT^{cov}_{CDC}| + \delta \cdot |OPT^{rel}_{CDC}| \\ &\leq (\gamma + \delta) \cdot |OPT_{CDC}| \quad (2) \end{aligned}$$

where the second line follows from the fact that the optimal GDC for the RNs is of lower cost than OPT^{cov}_{CDC}. $\square$

According to Theorem 1, even if the two subproblems are solved optimally (i.e. with $\gamma = \delta = 1$), this yields a 2-approximation to the CDC problem. A tight example of this fact is illustrated in Figure 2. Figure 2-a shows an n node instance of the CDC problem, where $\varepsilon << r$ refers to a sufficiently small constant. Also shown is the optimal solution with cost n MBNs. Figure 2-b shows a solution using the decomposition framework (with $\gamma = \delta = 1$), composed of an optimal disk cover and an optimal STP-MSP solution. The cost is $n + n - 1 = 2n - 1$.

We note that if a centralized solution can be tolerated, the approximation ratio of the GDC problem can be very close to 1 (e.g. using a PTAS [13]). In addition, the lowest known approximation ratio of the STP-MSP solution is 2.5 [3]. Therefore, the *framework immediately yields a 3.5-approximation algorithm for the solution of the CDC problem.* We note that any future improvement to the approximation ratio of STP-MSP will directly reduce the CDC approximation ratio.

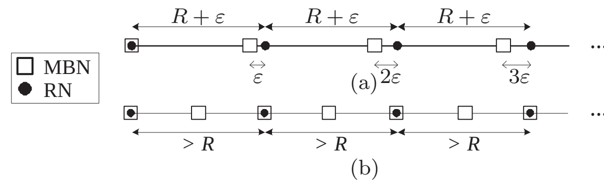

Figure 2: Tight example of the approximation ratio of the decomposition algorithm: (a) optimal solution and (b) decomposition algorithm solution.

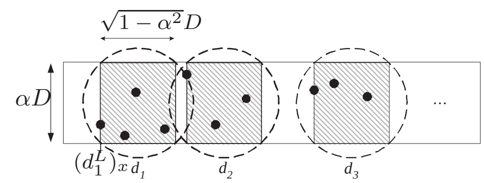

Figure 3: An example illustrating step 9 of the SCR algorithm.

5. PLACING THE COVER MBNS

5.1 Strip Cover Algorithms

Hochbaum and Maass [13] introduced a method for approaching the GDC problem by (i) dividing the plane into equal width strips, (ii) solving the problem locally on the points within each strip, and (iii) taking the overall solution as the union of all local solutions. Below we present algorithms that are based on this method. These algorithms are actually two different versions of a single generic algorithm. The first version locally covers the strip with rectangles encapsulated in disks while the second version locally covers the strip directly with disks. We then generalize (to arbitrary strip widths) the effects of solving the problem locally in strips. We use this extension to provide approximation guarantees for the two algorithms in the worst case and in the average case. Finally, we discuss the distributed implementation of the algorithms.

5.1.1 Centralized Algorithms

For simplicity of the presentation, we start by describing the centralized algorithms. The two versions of the Strip Cover algorithm (*Strip Cover with Rectangles* - SCR and *Strip Cover with Disks* - SCD) appear below. In line 6, the first version (SCR) calls the *Rectangles* procedure and the second one (SCD) calls the *Disks* procedure. The input is a set of points (RNs) $N = \{1, 2, \ldots, n\}$ and their (x, y) coordinates, $(i_x, i_y)\, \forall i$. The output includes a set of disks (MBNs) $M = \{d_1, d_2, \ldots, d_m\}$ and their locations such that all points are covered. The first step of the algorithm is to divide the plane into K strips of width $q_{SC} = \alpha D$ (recall that $D = 2r$). The values of q_{SC} that guarantee certain approximation ratios will be derived below. We denote the strips by S_j and the set of MBNs in strip S_j by M_{S_j}.

An example of the SCR algorithm and in particular of step 9 in which disks are placed such that they compactly cover all points in the rectangular area with x-coordinate range i_x to $i_x + \sqrt{1-\alpha^2}D$ is shown in Figure 3.

As mentioned in Section 2, Gonzalez [8] presented an al-

Algorithm 1 Strip Cover with Rectangles/Disks (SCR/SCD)

1: **divide** the plane into K strips of width $q_{SC} = \alpha D$
2: $M_{S_j} \leftarrow \emptyset, \forall j = 1, \ldots, K$
3: **for** all strips $S_j, j = 1, \ldots, K$ **do**
4:     **while** there exist uncovered RNs in S_j **do**
5:         **let** i be the leftmost uncovered RN in S_j
6:         **call Rectangles**(i) or **call Disks**(i)
7:         $M_{S_j} \leftarrow M_{S_j} \cup d_k$
8: **return** $\bigcup_j M_{S_j}$

Procedure Rectangles(i)

9: **place** an MBN d_k such that it covers all RNs in the rectangular area with x-coordinates $[i_x, i_x + \sqrt{1-\alpha^2}D]$
10: **return** d_k

Procedure Disks(i)

11: $P_{d_k} \leftarrow \emptyset$ {set of RNs covered by the current MBN d_k}
12: **while** $P_{d_k} \cup i$ coverable by a single MBN (disk) **do**
13:     $P_{d_k} \leftarrow P_{d_k} \cup i$
14:     **if** there are no more RNs in the strip **then**
15:         **break**
16:     **let** i be the next leftmost uncovered RN in S_j not currently in P_{d_k}
17: **place** MBN (disk) d_k such that it covers the RNs P_{d_k}
18: **return** d_k

gorithm for covering points with unit-squares. It is based on dividing the plane into equal width strips and covering the points in each of the strips separately. In [5] it was indicated that when the same algorithm is applied to covering points with unit disks, the approximation ratio is 8. The SCR algorithm is actually a slight modification to the algorithm of [8]. Unlike in [8], in our algorithm we allow the selection of the strip width. This will enable us to prove that the approximation ratio for covering points with unit disks is actually 6.

The SCD algorithm requires to answer the following question (in Step 12): can a set of points $P_{d_k} \cup i$ be covered by a single disk of radius r? This is actually the decision version of the *1-center problem*. Many algorithms for solving this problem exist, an example being an $O(n \log n)$ algorithm due to [12]. We will show that solving the 1-center problem instead of compactly covering rectangles (as done in the SCR algorithm) provides a lower approximation ratio.

The computation complexity of the SCR algorithm is $O(n \log n)$, resulting from sorting the points by ascending x-coordinate. In the SCD algorithm the 1-center subroutine may potentially need to be executed as many as $O(n)$ times for each of the $O(n)$ disks placed. Therefore, the computation complexity is $O(C(n)n^2)$, where $C(n)$ is the running time of the 1-center subroutine used in steps 12 and 17. By using a binary search technique to find the maximal P_{d_k}, we can lower the complexity to $O(C(n) n \log n)$.

5.1.2 Approximation Ratios

Let algorithm A denote the local algorithm within a strip, and let $|A_{S_j}|$ denote the cardinality of the GDC solution found by algorithm A covering only the points in strip S_j. Let algorithm B represent the overall algorithm, which works by running algorithm A locally within each strip and taking the union of the local solutions as the overall solution. In our case algorithm B is either the SCR or SCD algorithm and algorithm A is composed of steps 4-7 within the for loop.

Let $|OPT|$ represent the cardinality of an optimal solution of the GDC problem in the plane and $|OPT_{S_j}|$ the cardinality of an optimal solution for points exclusively within strip S_j. Note that $OPT \neq \bigcup_{S_j} OPT_{S_j}$, since OPT can utilize disks covering points across multiple strips. Finally, let Z_A denote the worst case approximation ratio of algorithm A. Namely, Z_A is the maximum of $|A_{S_j}|/|OPT_{S_j}|$ over all possible point-set configurations in a strip S_j. Similarly, let Z_B denote the worst case approximation ratio of algorithm B.

We characterize Z_B as a function of Z_A. Namely, if $q \leq D$, the cardinality of the solution found by algorithm B is at most $(\lceil \frac{D}{q} \rceil + 1) Z_A$ times that of the optimal solution, $|OPT|$.

Observation 1. *If the strip width is $q \leq D$, a single disk can cover points from at most $(\lceil \frac{D}{q} \rceil + 1)$ strips.*

Lemma 1. *If the strip width is $q \leq D$, $Z_B = (\lceil \frac{D}{q} \rceil + 1) Z_A$.*

Proof. Consider the set of disks in an optimal solution to the GDC problem in the plane, $OPT = d_1, \ldots, d_{|OPT|}$. From OPT, we can create an "algorithm B type" solution (i.e. made up of disks covering points only from single strips) in the following way. Assume OPT disk d_k covers points from c_k different strips (e.g. $S_j, S_{j+1}, \ldots, S_{j+c_k-1}$). For each such d_k, create c_k new disks $d'_1, d'_2, \ldots, d'_{c_k}$ and assign to each d'_j the points covered by d_k that lie exclusively within strip S_j.

Upon doing this for all $d_k \in OPT$, let OPT' denote the resulting set of disks. Clearly, OPT' can be expressed as $\bigcup_{S_j} OPT'_{S_j}$, where OPT'_{S_j} represents the subset of disks in OPT' that cover points exclusively within strip S_j. Therefore, we have that,

$$|OPT'| = \sum_{S_j} |OPT'_{S_j}| = \sum_{k=1}^{|OPT|} c_k \leq \left(\left\lceil \frac{D}{q} \right\rceil + 1 \right) \cdot |OPT| \quad (3)$$

where the second equality, i.e. converting a sum over strips into a sum over disks, follows from the construction of OPT', and the inequality follows from Observation 1.

Next, we note that by definition $|A_{S_j}| \leq Z_A \cdot |OPT_{S_j}|$. Combining this with the fact that $OPT_{S_j} \leq OPT'_{S_j}$ for all strips S_j, we have that,

$$\begin{aligned} |B| &= \sum_{S_j} |A_{S_j}| \leq Z_A \cdot \sum_{S_j} |OPT_{S_j}| \leq Z_A \cdot |OPT'| \\ &\leq Z_A \cdot \left(\left\lceil \frac{D}{q} \right\rceil + 1 \right) \cdot |OPT| \quad (4) \end{aligned}$$

where the last inequality followed from (3). □

We now show that in the SCR algorithm, $Z_A = 2$. This approximation ratio is tight, as illustrated in Figure 4-a.

Lemma 2. *If the strip width $q_{SC} \leq \frac{\sqrt{3}D}{2}$, steps 4-7 of the SCR algorithm provide a 2-approximation algorithm for the GDC problem within a strip.*

Proof. Consider some strip S. Let $OPT_S = \{d_1, d_2, \ldots, d_{|OPT_S|}\}$ and $ALGO_S = \{a_1, a_2, \ldots, a_{|ALGO_S|}\}$ denote an optimal in-strip solution and SCR in-strip subroutine (steps 4-7) solution, respectively. Recall that we assume that the MBNs of both OPT_S and $ALGO_S$ are ordered from left to right by x-coordinate of the leftmost covered point (i.e. $i < j$ if $(d_i^L)_x \leq (d_j^L)_x$). Finally, define a_{b_m} as the b_m^{th} algorithm

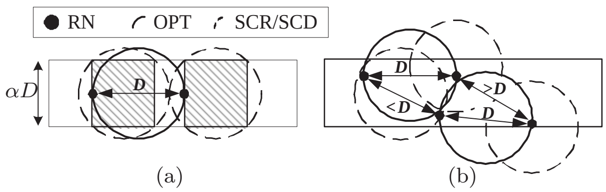

Figure 4: Tight examples of the 2 and 1.5 approximation ratios obtained by the in-strip subroutines of the (a) SCR and (b) SCD algorithms.

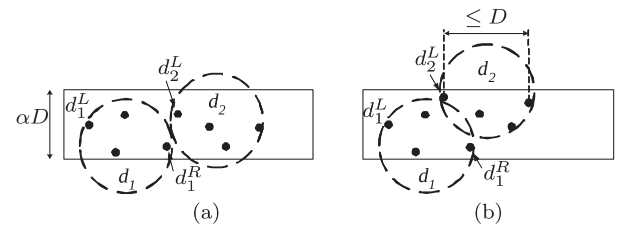

Figure 5: Illustration of the SCD induction proof. (a) "Sub" Base-Case 1: $(d_1^R)_x < (d_2^L)_x$. (b) "Sub" Base-Case 2: $(d_1^R)_x \geq (d_2^L)_x$.

disk (from the left) corresponding to the disk that covers the rightmost point covered by the m^{th} OPT_S disk d_m.

Let $q_{SC} = \alpha D, \alpha < 1$. We now prove by induction that if $\alpha \leq \sqrt{3}/2$, the in-strip subroutine has approximation ratio of 2, i.e. $|ALGO_S| = b_{|OPT_S|} \leq 2|OPT_S|$.

Base Case: The *area* covered by d_1 (the leftmost optimal disk) is bounded by a rectangle with x-coordinate range $(d_1^L)_x$ (the x-coordinate of the leftmost point) to $(d_1^L)_x + D$. The *minimum* area covered by two SCR algorithm disks whose leftmost point is $(d_1^L)_x$ is a rectangle with x-coordinate range $(d_1^L)_x$ to $(d_1^L)_x + 2\sqrt{1-\alpha^2}D$. Thus, if $2\sqrt{1-\alpha^2}D \geq D$, $b_1 \leq 2$. This condition is met if $q_{SC} \leq \sqrt{3}D/2$.

Inductive Step: Assume that the in-strip algorithm uses no more than $2m$ disks to cover all the points covered by $d_1, \ldots, d_m$ (i.e. $b_m \leq 2m$). Consider the number of additional disks it takes for the algorithm to cover the points covered by $d_1, \ldots, d_m, d_{m+1}$. Since all of the points up to the rightmost point of d_m are already covered, by the same argument as the base case, the algorithm will use at most 2 extra disks to cover the points covered by d_{m+1}. It thus follows that if $q \leq \sqrt{3}D/2$, $b_{m+1} \leq b_m + 2 \leq 2m + 2 = 2(m+1)$. □

By combining the results of lemmas 1 and 2, we obtain the approximation ratio of the SCR algorithm.

THEOREM 2. *If $\frac{D}{2} \leq q_{SC} \leq \frac{\sqrt{3}D}{2}$, the SCR algorithm is a 6-approximation algorithm for the GDC problem.*

PROOF. Define algorithm A as the in-strip subroutine of the SCR algorithm (steps 4-7) and algorithm B as the SCR algorithm. From Lemma 2, for $q \leq \sqrt{3}D/2$, $Z_A = 2$. From Lemma 1, $Z_B \leq Z_A(\lceil D/q \rceil + 1)$, the minimum value of which (for $q < D$) is $3Z_A$. This is attained when $q \geq D/2$. □

Below it is shown that in the SCD algorithm $Z_A = 1.5$. Combining this result with Lemma 1 (similarly to the derivation of Theorem 2), we obtain the approximation ratio of the SCD algorithm. Notice that the approximation ratio of 1.5 for the in-strip subroutine of the SCD algorithm is tight, as illustrated in Figure 4-b.

LEMMA 3. *If $q_{SC} \leq \frac{\sqrt{3}D}{2}$, steps 4-7 of the SCD algorithm provide a 1.5-approximation algorithm for the GDC problem within a strip.*

PROOF OF LEMMA 3. Similar to the proof of Lemma 2, we use induction to prove the result. We utilize the same definitions as from that proof.

Base Case: There are 2 "sub" base-cases to consider. First, assume $(d_1^R)_x < (d_2^L)_x$, as shown in Figure 5-a. If this this is the case, it is easy to see that $b_1 = 1$, as by definition all of the points from d_1^L to d_1^R are coverable by a single disk; this fact would have been exploited in step 12 of the SC algorithm. Second, assume $(d_1^R)_x \geq (d_2^L)_x$, as shown in Figure 5-b. In this scenario, we consider a base case of $m = 2$, and show that $b_2 \leq 3$. To see this, first note that all points immediately left of d_2^L could have and therefore would have been covered by a single SCD disk as per line 17 of the algorithm. Next, as shown in Figure 5-b, we note that the remaining uncovered points all lie within a rectangular area of at most D along the strip. Since a lower bound on the *area* each SCD disk must cover is $\sqrt{1-\alpha^2}D$ along the strip (i.e. this area is always compactly coverable), we have that these points will be covered by at most 2 disks as long as $2\sqrt{1-\alpha^2}D \geq D$. This condition is met if $\alpha \leq \frac{\sqrt{3}}{2}$.

Inductive Step: Assume the in-strip algorithm uses no more than $\frac{3}{2}m$ disks to cover all the points covered by $d_1, \ldots,$ d_m, i.e. $b_m \leq \frac{3}{2}m$. Assume also that m is even[2]. Now consider the number of additional disks it takes for the algorithm to cover the points covered by $d_1, \ldots, d_m, d_{m+1}$. Define d^{R*} as the rightmost point covered by $d_1, \ldots, d_m, d_{m+1}$, e.g. $d^{R*} = max_{1 \leq l \leq m+1}[d_l^R]$.

Again we have two cases: First, assume that $(d^{R*})_x < (d_{m+2}^L)_x$. This case is identical to the first base-case in that the algorithm uses exactly one extra disk to cover the points from d_{m+1}^L to d_{m+1}^R, i.e.,

$$b_{m+1} = b_m + 1 \leq \frac{3}{2}m + 1 \leq \frac{3}{2}(m+1). \quad (5)$$

The second case assumes $(d^{R*})_x \geq (d_{m+2}^L)_x$. This case is identical to the second base case, whereby we can conclude that the algorithm uses at most 3 extra disks in order to cover the points covered by $d_1, \ldots, d_{m+2}$, i.e.,

$$b_{m+2} \leq b_m + 3 = \frac{3}{2}m + 3 = \frac{3}{2}(m+2). \quad (6)$$

□

THEOREM 3. *If $\frac{D}{2} \leq q_{SC} \leq \frac{\sqrt{3}D}{2}$, the SCD algorithm is a 4.5-approximation algorithm for the GDC problem.*

PROOF. Exactly the same as the proof of Theorem 2, except that as per Lemma 3, we use $Z_A = 1.5$ instead of 2. □

Up to now we have discussed the *worst case* performance. We now wish to bound the approximation ratios of the SCR and the SCD algorithms in the *average case*. We assume that the RNs are randomly distributed according to a two

[2]The second base-case and second inductive-case ensure the lemma is true for all m.

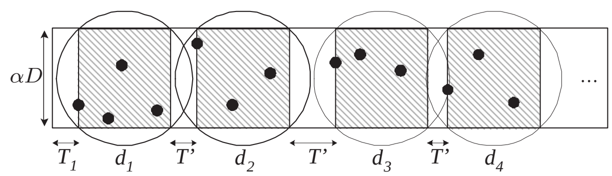

Figure 6: Probabilistic analysis of the performance of the SCR algorithm within a strip.

dimensional Poisson process[3]. Due to the random locations of the RNs, $|OPT|$ is a random variable. Similarly, we define $|SCR|$ and $|SCD|$ as random variables corresponding to the number of disks placed by the SCR and the SCD algorithms. We define the *average approximation ratios* β_{SCR} and β_{SCD} as,

$$\beta_{SCR} = \frac{E[|SCR|]}{E[|OPT|]},\ \beta_{SCD} = \frac{E[|SCD|]}{E[|OPT|]}. \tag{7}$$

It should be noted that β_{SCR} differs from the expected value of the approximation ratio ($E[|SCR|/|OPT|]$). Yet, it provides a good measure of the average performance.

The following Theorem and Corollary bound the average approximation ratio of the SCR algorithm, thereby bounding the ratio of the SCD algorithm (since SCD always outperforms SCR). It can be seen that although the worst case approximation ratios are 6 and 4.5 (respectively), selecting a specific strip width results in an average approximation ratio which is bounded by 3. In Section 8 we will show by simulation that in practice the approximation ratios are actually much lower.

THEOREM 4. *Given RNs distributed in the plane according to a two dimensional Poisson process with density* λ*,*

$$\beta_{SCD} \leq \beta_{SCR} \leq \frac{D^2\lambda + 2D\sqrt{\lambda} + 1}{\alpha\sqrt{1-\alpha^2}D^2\lambda + 1}. \tag{8}$$

PROOF. To prove the theorem, we start by upper bounding $E[|SCR|]$. To this end, consider a single strip S and recall that the SCR algorithm iteratively places disks by identifying the leftmost uncovered point i and fully covering the x-range between i_x and $i_x + \sqrt{1-\alpha^2}D$ along the strip. The points are distributed in the plane according to a two dimensional Poisson process with density λ. Therefore, the horizontal (x-coordinate) distance between points is exponentially distributed with average $\frac{1}{\lambda\alpha D}$. Thus, the expected distance to the location of the first disk is $E[T_1] = \frac{1}{\lambda\alpha D}$ (see Figure 6.). Furthermore, once a disk is placed, the expected distance between the end of its coverage and the start of the next disk is $E[T']$. Due to the memoryless property of the exponential random variable, we can conclude $E[T'] = \frac{1}{\lambda\alpha D}$.

It therefore follows that the expected number of disks used by the SCR algorithm within a strip is the total length of the strip (less the initial space) divided by the expected distance between the start of one disk and the start of another. Namely,

[3]When the number of RNs is given, their positions are independent and each is *uniformly distributed* in the plane.

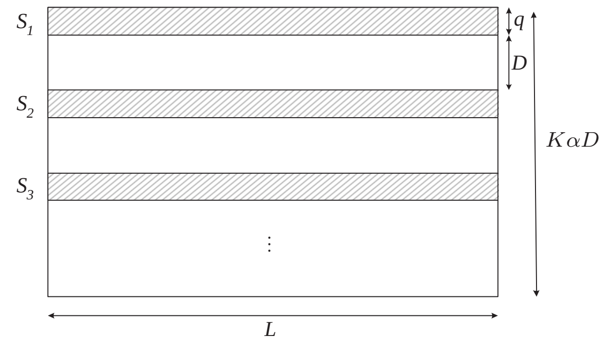

Figure 7: Dividing the plane into strips in order to lower bound $E[|OPT|]$

$$\begin{aligned} E[|SCR|_S] &= \frac{L - \frac{1}{\lambda\alpha D}}{\sqrt{1-\alpha^2}D + \frac{1}{\lambda\alpha D}} \\ &\approx \frac{L}{\sqrt{1-\alpha^2}D + \frac{1}{\lambda\alpha D}} \\ &= \frac{\lambda\alpha DL}{\lambda\alpha\sqrt{1-\alpha^2}D^2 + 1} \end{aligned} \tag{9}$$

where $E[|SCR|_S]$ is the expected number of disks used by the SCR algorithm within strip S. Note that in the second line of (9) we assume that $L >> \frac{1}{\lambda\alpha D}$; we technically don't need this assumption, but it makes the analysis cleaner. The expected total number of disks used by the algorithm over the entire plane is therefore this number multiplied by the total number of strips in the plane, i.e.,

$$E[|SCR|] = \frac{\lambda\alpha DLK}{\lambda\alpha\sqrt{1-\alpha^2}D^2 + 1}. \tag{10}$$

We next aim to lower bound $E[|OPT|]$. To this end, we divide the plane into D-spaced horizontal strips of width q as shown in Figure 7.

We can lower bound the expected number of disks used to cover points in a single strip S by an optimal algorithm by noting that the area coverable by each OPT disk is no more than a rectangle of size $q \times D$. Thus, using a similar argument to when we upper bounded the number of SCR disks required to cover a strip, we have that,

$$E[|OPT|_S] \geq \frac{L}{D + \frac{1}{\lambda q}} \tag{11}$$

where $E[|OPT|_S]$ is the expected number of disks used by the optimal solution within strip S. Next we note that an upper bound on the expected number of OPT disks used to cover points in the whole plane can be achieved by summing over the disks used to cover each of the individual strips. The reason we can do this is that since there is a distance D between strips, it is impossible for a single OPT disk to simultaneously cover points from two different strips. We therefore have that,

$$
\begin{aligned}
E[|OPT|] &\geq \left(\frac{L}{D+\frac{1}{\lambda q}}\right)\cdot\left(\frac{K\alpha D}{D+q}\right) \\
&= \frac{KL\alpha D}{D^2+\frac{1}{\lambda}+\left(Dq+\frac{D}{\lambda q}\right)}. \qquad (12)
\end{aligned}
$$

Next, since we have control over the strip size q, and want to find the tightest possible lower bound, we can select q so as to maximize $E[|OPT|]$, i.e. minimize the bracketed quantity in the denominator of (12). It turns out that setting $q=\sqrt{\frac{1}{\lambda}}$ achieves this. Substituting this into (12), we have that,

$$
E[|OPT|] \geq \frac{KL\alpha D}{D^2+\frac{1}{\lambda}+\frac{2D}{\sqrt{\lambda}}}. \qquad (13)
$$

Finally, combining (13), (10) and (7) gives us our desired upper bound on β_{SCR}, i.e.,

$$
\begin{aligned}
\beta_{SCR} &\leq \left(\frac{\lambda\alpha DLK}{\lambda\alpha\sqrt{1-\alpha^2}D^2+1}\right)\cdot\left(\frac{\lambda D^2+2\sqrt{\lambda}D+1}{\lambda\alpha DLK}\right) \\
&= \frac{D^2\lambda+2D\sqrt{\lambda}+1}{\alpha\sqrt{1-\alpha^2}D^2\lambda+1}. \qquad (14)
\end{aligned}
$$

□

Corollary 1. *If $q_{SC}=\frac{D}{\sqrt{2}}$, then $\beta_{SCD}\leq\beta_{SCR}\leq 3$.*

Proof. We derive the maximum value of (14) by differentiating with respect to λ. Upon doing so and plugging this value of λ into (14) gives us,

$$
\beta_{SCR}\mid_{\lambda=\lambda_{max}} \leq \frac{\alpha\sqrt{1-\alpha^2}+1}{\alpha\sqrt{1-\alpha^2}} \qquad (15)
$$

which is interestingly independent of D. Finally, we note that for $\frac{1}{2}\leq\alpha<1$, (15) is minimized when $\alpha=\frac{1}{\sqrt{2}}$, at which point it achieves a value of exactly 3. □

5.1.3 Distributed Implementation

By construction, the SCR and SCD algorithms can be easily implemented in a distributed manner. The algorithms are executed at the RNs and operate within the strips. Thus, we assume that the strips are fixed and that their boundaries are known to all nodes. The SCR algorithm executed at an RN i, consisting of rules regarding initial construction and maintenance under RN mobility is described below. RN mobility affects the design of the algorithms, since it can cause an RN to disconnect from its MBN or to move to a neighboring strip in which it is not covered by an MBN. Recall that we denote the RNs within a strip according to their order from the left (i.e. $i<j$ if $i_x\leq j_x$). Ties are broken by node ID.

It can be seen that every RN that has no left neighbors within distance D initiates the disk placement procedure that propagates along the strip. The propagation stops once there is a gap between nodes of at least D. If an RN arrives from a neighboring strip or leaves its MBN's coverage area, it initiates the disk placement procedure that may trigger an update of the MBN's locations within the strip. Notice that MBNs only move when a recalculation is required. Although the responsibility to place and move MBNs is with the RNs,

Algorithm 2 Distributed SCR (at RN i)

Initialization
1: **let** G_i be the set of RNs j such that $j<i$ and $i_x-j_x\leq D$
2: **if** $G_i=\emptyset$ **then**
3:   **call** Place MBN
Construction and Maintenance
4: **if** *MBN Placed* message received **then**
5:   **call** Place MBN
6: **if** i is disconnected from its MBN or enters from a neighboring strip **then**
7:   **if** there is at least one MBN within distance r **then**
8:     **join** one of these MBNs
9:   **else**
10:     **call** Place MBN
Procedure Place MBN
11: **let** i^R be the rightmost RN s.t. $(i^R)_x\leq i_x+\sqrt{1-\alpha^2}D$
12: **place** MBN d_k covering RNs j, where $j_x\in[i_x,(i^R)_x]$
13: **if** $(i^R+1)_x-(i^R)_x\leq D$ **then**
14:   **send** an *MBN Placed* message to i^R+1

simple enhancements would allow the MBNs to reposition themselves during the maintenance phase.

The time complexity (i.e. number of rounds) is $O(n)$, since *MBN Placed* messages may potentially have to propagate the entire length of the strip. As mentioned in Section 3, some of the nodes may need to perform a local computation to maintain the ordered list of their neighbors. The complexity of this computation is potentially $O(\log n)$. Control information has to be transmitted between RNs over a distance $D=2r$. Recall that in Section 3 we assumed that there is a long range control channel. Therefore, once RNs decide to place an MBN, we assume that there is a way to communicate this to one of the MBNs.

The distributed SCD algorithm is similar to the distributed SCR algorithm. The main difference is that in Step 11 of *Place MBN*, i^R is defined as the rightmost coverable point (by a single disk of radius r), given that i is the leftmost point. As mentioned earlier, finding this point requires solving 1-center problems. Then, in Step 12 a disk that covers all the points between i and i^R should be placed. The time complexity of the distributed SCD algorithm is again $O(n)$. The local computation complexity is $O(C(n)\log n)$ to calculate the value of i^R, where $C(n)$ is the running time of the 1-center subroutine used.

5.2 MObile Area Cover (MOAC) Algorithm

In the SCR and SCD algorithms, an RN movement may change the allocation of RNs to MBNs along the whole strip. Thus, although they may operate well in a relatively static environment, it is desirable to develop algorithms that are more tailored to frequent node movements. In this section we present such an algorithm which builds upon ideas presented in [11]. As mentioned in Section 2, Hershberger [11] studied the problem of covering moving points (e.g. RNs) with mobile unit-squares (e.g. MBNs). Since the d-dimensional *smooth maintenance scheme* proposed in [11] does not easily lend itself to distributed implementation, we focus on the *simple 1-D algorithm* proposed there.

Applied to our context, the Simple 1-D algorithm covers mobile RNs along the strip with length D rectangles (MBNs). The key feature is that point transfers between MBNs are *localized*. Namely, changes do not propagate along the strip. According to [11], the algorithm has a worst case performance ratio of 3.[4]

Extending the *Simple 1-D algorithm* of [11] to diameter D disks is not straightforward. We will first show that an attempt to simply use rectangles encapsulated in disks without any additional modifications results in a 4-approximation to the GDC problem within a strip. Then, we will present the MObile Area Cover (MOAC) algorithm which reduces the approximation ratio to 3.

We define the strip width as $q_{MOAC} = \alpha D$. We reduce disks to the rectangles encapsulated in them and use these rectangles to cover points within the strip, as was depicted in Figure 3. The rectangles cover the strip width (αD) and their length is *at most* $\sqrt{1-\alpha^2}D$. We set $D = 1$ and $\alpha = \sqrt{5}/3$ (resulting in $\sqrt{1-\alpha^2}D = 2/3$). These are arbitrary values selected for the ease of presentation. Yet, the algorithm and the analysis are applicable to any $1/2 \leq \alpha \leq \sqrt{5}/3$. We restate the set of rules from [11] using our terminology and assuming (unlike [11]) that the rectangles' lengths are at most 2/3.

Algorithm 3 Simple 1-D [11] with $\sqrt{1-\alpha^2}D = 2/3$

0 **initialize** the cover greedily {using the SCR algorithm}
1 **maintain** the leftmost RN and rightmost RN of each MBN rectangle
2 **if** two adjacent MBN rectangles come into contact **then**
  exchange their outermost RNs
3 **If** a set of RNs covered by an MBN becomes too long {the separation between its leftmost and rightmost RNs becomes greater than 2/3} **then**
  split off its rightmost RN into a singleton MBN
  check whether rule 4 applies
4 **if** two adjacent MBN rectangles fit in a 2/3 rectangle **then**
  merge the two MBNs

The following lemma provides the performance guarantee of this algorithm. Notice that since the changes are kept local, the approximation ratio holds at all time (i.e. there is no need to wait until the changes propagate).

Lemma 4. *The* Simple 1-D algorithm [11] *with* $\sqrt{1-\alpha^2} = 2/3$ *is* at all times *a 4-approximation algorithm for the GDC problem within a strip.*

Proof. To begin, we assume the same definitions of OPT_S, $ALGO_S$, and b_m from Lemma 2. We now proceed to prove the theorem by induction.

Base Case: The length (along the strip) covered by d_1 (the leftmost optimal disk) is at most 1 (recall that we preset $D = 1$ for this section). Next, we show by a packing argument that at most 4 $ALGO_S$ disks can simultaneously cover points from such a unit-length interval where by assumption, no uncovered points exist to the left of d_1^L. To see why, assume 5 such $ALGO_S$ disks existed. However, this would mean that the member points of four of the disks all lay within a unit interval. We define the "combined length" of adjacent $ALGO_S$ disks m_j and m_{j+1} as $Q_j = (d_{j+1}^R)_x - (d_j^L)_x$. We thus have that $Q_1 + Q_3 \leq 1$. However, from rule 4 of the Simple 1-D algorithm two adjacent disks m_j and m_{j+1} are merged if their combined length

[4]We note that using the same inductive proof methodology, used for Lemma 2, one can show that the simple 1-D algorithm actually maintains a 2-approximation at all times.

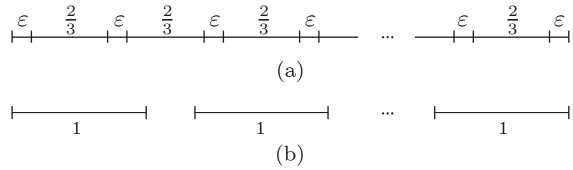

Figure 8: Worst case example for the performance of the Simple 1-D algorithm: (a) algorithmic solution and (b) optimal solution. The example is illustrated in 1-dimension, where the intervals represent the x-range of rectangles within the MBNs' disks.

$Q_j \leq \frac{2}{3}$. Therefore, assuming $Q_1 > \frac{2}{3}$ and $Q_3 > \frac{2}{3}$, we have that $Q_1 + Q_3 > \frac{4}{3} > 1$, which is a contradiction.

Inductive Step: Assume the Simple 1-D algorithm uses no more than $4m$ disks to cover all the points covered by $d_1, \ldots, d_m$, i.e. $b_m \leq 4m$. Now consider the number of additional disks it takes for the algorithm to cover the points covered by $d_1, \ldots, d_m, d_{m+1}$. Since all of the points up to the rightmost point of d_m are already covered, by the same argument as the base case the algorithm will use at most 4 extra disks to cover the points covered by d_{m+1}. Thus, we have that,

$$b_{m+1} \leq b_m + 4 \leq 4m + 4 = 4(m+1). \tag{16}$$

□

From lemmas 1 and 4 it follows that if implemented simultaneously in every strip, the algorithm provides a 12-approximation for the GDC problem in the plane, which is relatively high. We now focus on enhancements that reduce the approximation ratio while maintaining the desired locality property.

Figure 8 presents an example which shows that the approximation ratio described in Lemma 4 is tight. It can be seen that the performance ratio is $(4k-1)/k$, where k is the optimal number of disks. One of the sources of inefficiency is the potential presence of ε-length MBNs (e.g. covering a single RN) that cannot merge with their 2/3-length neighbor MBNs. Thus, up to 5 MBNs deployed by the Simple 1-D algorithm may cover points which are covered by a single optimal MBN. As long as such narrow MBNs can be avoided, a better approximation can be achieved. We now modify the Simple 1-D algorithm to yield the MOAC algorithm in which ε-length MBNs cannot exist.

Before describing the algorithm, we make the following definitions. For MBN d_i, in addition to its leftmost and rightmost RNs, defined earlier, as d_i^L and d_i^R, we also define L_i and R_i as the x-coordinates of its left and right *domain* boundaries. The interpretation of MBN d_i's domain is that any point in the x-range of $[L_i, R_i]$ will automatically become a member point of MBN d_i. Recall that by definition MBN d_i is to the left of MBN d_j if $(d_i^L)_x < (d_j^L)_x$.

The MOAC algorithm operates within strips and maintains the following *invariants* in each strip (in order of priority) at all times, for every MBN d_i:

1. Domain definition - $L_i \leq (d_i^L)_x \leq (d_i^R)_x \leq R_i$.
2. Domain length[5] - $\frac{1}{3} \leq |R_i - L_i| \leq \frac{2}{3}$.

[5]The upper bound is the coverage length of a MOAC MBN

3. Domain disjointness - $[L_i, R_i] \bigcap [L_j, R_j] = \emptyset, \forall d_j \in M$.

4. Domain influence - $\forall p \in N$, $L_i \leq p_x \leq R_i \leftrightarrow p_x \in P_{d_i}$.

We describe the MOAC algorithm below. It consists of rules regarding construction and maintenance of the MBN cover. This algorithm can be implemented in distributed manner by applying some of the rules at the MBNs and some of them at disconnected (i.e. uncovered) RNs (it is clear from the context where each rule should be applied). For brevity, we only state the maintenance rules for the case in which an RN moves outside its MBN's domain boundary to the *right* (analogous rules apply to a leftward movement).

Algorithm 4 MObile Area Cover (MOAC)

Initialization
1: **cover** the RNs with MBNs using the SCR algorithm
2: **for** all MBNs i **do**
3: $L_i \leftarrow d_i^L$; $R_i \leftarrow d_i^L + \frac{2}{3}$
4: $P_{d_i} \leftarrow$ all RNs within $[L_i, R_i]$

Maintenance
5: **if** an RN $p \in P_{d_i}$ moves *right* such that $p_x > R_i$ **then**
6: **if** $L_j \leq p_x \leq R_j$, $j \neq i$ {p in d_j's domain} **then**
7: **remove** p from P_{d_i}
8: **else if** $|p_x - L_i| \leq \frac{2}{3}$ **then**
9: **stretch** L_i and R_i to maintain invariant (1) by setting $R_i \leftarrow p_x$ and $L_i \leftarrow max(L_i, p_x - \frac{2}{3})$
10: **else** {p not in the immediate domain of any MBN}
11: **remove** p from P_{d_i}

Disconnection
12: **if** at any time there exists an uncovered RN p **then**
13: **if** for some MBN d_j, $L_j \leq p_x \leq R_j$ **then**
14: $P_{d_j} \leftarrow P_{d_j} \cup p$
15: **else if** for some MBN d_j, L_j and R_j can be *stretched* to include p while maintaining invariant (2) **then**
16: $P_{d_j} \leftarrow P_{d_j} \cup p$
17: **stretch** L_j and R_j to maintain invariants (1),(2)
18: **else** {p cannot be covered by an existing MBN}
19: **let** d_{j-1} and d_{j+1} represent the MBNs to the left and right of p
20: **if** $|L_{j+1} - R_{j-1}| \geq \frac{1}{3}$ {i.e. enough "open space" to maintain invariant (2)} **then**
21: **create** MBN d_j with $P_{d_j} = p$ and $|R_j - L_j| \geq \frac{1}{3}$ while maintaining invariant (3)
22: **else** {$< \frac{1}{3}$ space around p}
23: **shrink** MBN d_{j-1} such that $R_{j-1} = p_x - \frac{1}{3}$
24: **create** MBN d_j with $L_j = p_x - \frac{1}{3}$ and $R_j = p_x$
25: $P_{d_{j-1}} \leftarrow$ all points in $[L_{j-1}, R_{j-1}]$
26: $P_{d_j} \leftarrow$ all points in $[L_j, R_j]$

Merge
27: **if** there exists MBN d_j such that $|(d_j^R)_x - (d_i^L)_x| \leq \frac{2}{3}$ or $|(d_i^R)_x - (d_j^L)_x| \leq \frac{2}{3}$ **then**
28: **merge** d_j into d_i

It should be noted that the operations in lines 22-26 can always be accomplished without violating invariant (2). This is due to the fact that an MBN d_j is created for point p only if $|p_x - L_{j-1}| > 2/3$ (otherwise MBN d_{j-1} would have been stretched to cover p), which implies there is enough space for two MBNs of size greater or equal to 1/3 to coexist. Following the merge in line 28, the MBN should update its L_i and R_i such that the domain will include all RNs and will satisfy invariant (2). This is always possible, since the two merged MBNs satisfy the invariants prior to their merger.

(here arbitrarily chosen as $\sqrt{1-\alpha^2}D = 2/3$). To maintain the algorithm's properties, the lower bound should be half of the upper bound and their sum should be at least one. In addition, due to Lemma 1, $\alpha \geq 0.5$.

The following lemma provides the performance guarantee of the MOAC algorithm within the strip. From Lemma 1 it follows that if MOAC is simultaneously executed in all strips, it is a 9-approximation algorithm.

Lemma 5. *The MOAC algorithm is a 3-approximation algorithm* at all times *for the GDC problem within a strip.*

Proof. The proof is almost identical to that of Lemma 4, except now we define the "domain length" of each $ALGO_S$ MBN m_j separately, as $Q_j = |R_j - L_j|$.
Base Case: Again, recall that the length (along the strip) covered by d_1 (the leftmost optimal disk) is at most 1. This time we show that at most 3 $ALGO_S$ disks can simultaneously cover points from this interval where by assumption, no uncovered points exist to the left of d_1^L. Too see why, assume 4 such $ALGO_S$ disks existed. As before, this means that the member points of 3 of these $ALGO_S$ disks must all lay within this interval, requiring that $\sum_{j=1}^{3} Q_j \leq 1$. However, the merging rule of the algorithm implies that the sum of domain lengths of two adjacent disks must be $> \frac{2}{3}$. Furthermore, invariant (ii) of the algorithm states that the domain length of any disk must be $\geq \frac{1}{3}$. We therefore have that $Q_1 + Q_2 > \frac{2}{3}$ and $Q_3 \geq \frac{1}{3}$, which together imply $\sum_{j=1}^{3} Q_j > 1$, which is a contradiction.
Inductive Step: Assume the MOAC algorithm uses no more than $3m$ disks to cover all the points covered by $d_1, \ldots, d_m$, i.e. $b_m \leq 3m$. Now consider the number of additional disks it takes for the algorithm to cover the points covered by $d_1, \ldots, d_m, d_{m+1}$. Since all of the points up to the rightmost point of d_m are already covered, by the same argument as the base case the algorithm will use at most 3 extra disks to cover the points covered by d_{m+1}. Thus, we have that,

$$b_{m+1} \leq b_m + 3 \leq 3m + 3 \leq 3(m+1). \qquad (17)$$

□

The time complexity of the MOAC algorithm is $O(1)$, since all node exchanges are local. The local computation complexity is potentially $O(\log n)$, due to the operation in line 23. The only assumption required is that MBNs and disconnected RNs have access to information regarding L_j, d_j^L, d_j^R and R_j of their immediate neighbors to the right and left (as long as they are less than $2D$ away). Thus, in terms of complexity, the MOAC algorithm is the best of the distributed algorithms.

5.3 Merge-and-Separate (MAS) Algorithm

The relatively high approximation ratio of the MOAC algorithm results from the fact that it reduces disks into rectangles, thereby losing about 35% of disk coverage area. The difficulty in dealing with disks is that there are no clear *borders* and that even confined to a single strip, many disks can overlap even though they cover disjoint nodes.

On average any algorithm with a merge rule should perform well. However, just having a merge rule is not sufficient in the rare but possible case where many mutually pairwise *non-mergeable* MBNs move into the same area. Based on this premise, we present the Merge-And-Separate (MAS) algorithm, as an algorithm which merges pairwise disks where possible (similar to the MOAC algorithm) and separates disks, if too many mutually non-mergeable disks concentrate in a small area. As will be shown, the MAS algorithm retains some of the localized features of the MOAC and obtains a better performance ratio. However, this comes at a cost of increased local computation complexity.

We define the strip-widths as $q_{MAS} = \alpha D$ and set $D = 1$, $\alpha = \sqrt{5}/3$, $\sqrt{1-\alpha^2} = 2/3$. These are arbitrary values selected for the ease of presentation, the algorithm and the analysis are applicable to any $0.5 \le \alpha < \sqrt{3}/2$. Let $x_{R_{\{i,j,k\}}}$ and $x_{L_{\{i,j,k\}}}$ be the x-coordinates of the rightmost and leftmost points of $\{P_{d_i} \cup P_{d_j} \cup P_{d_k}\}$. The algorithm is initialized by covering the nodes within a strip with MBNs by using the distributed SCR algorithm. The algorithm that then operates at an MBN d_i is described below. We note that as in the previous algorithms, most of the operations are performed in reaction to an RN movement. However, in order to maintain the locality of the algorithm, the Separation operation is performed periodically at each MBN. Figure 9 demonstrates the Separation done at lines 8-11. For correctness of the algorithm, we assume that both the merge and separate operations can be executed atomically (i.e. without any interrupting operation).

Algorithm 5 Merge-and-Seperate (MAS)

Initialization
1: **cover** the RNs with MBNs using the SCR algorithm
2: $P_{d_i} \leftarrow$ all RNs within $[L_i, R_i]$
Merge
3: **for** all MBNs d_k within $2D$ of d_i **do**
4: **if** $\{P_{d_i} \bigcup P_{d_k}\}$ can be covered by a single MBN **then**
5: **merge** d_i and d_k
Separation
6: **for** all MBN pairs d_j, d_k within $2D$ of d_i **do**
7: **if** $|x_{R_{\{i,j,k\}}} - x_{L_{\{i,j,k\}}}| \le 2D$ **then**
8: **separate and reassign** MBNs and RNs such that
9: $P_{d_i} \leftarrow$ all RNs in $[x_{L_{\{i,j,k\}}}, x_{L_{\{i,j,k\}}} + \frac{2}{3}]$
10: $P_{d_j} \leftarrow$ all RNs in $[x_{L_{\{i,j,k\}}} + \frac{2}{3}, x_{L_{\{i,j,k\}}} + \frac{4}{3}]$
11: $P_{d_k} \leftarrow$ all RNs in $[x_{L_{\{i,j,k\}}} + \frac{4}{3}, x_{R_{\{i,j,k\}}}]$
Creation
12: **if** an RN p enters from a neighboring strip or an RN $p \in P_{d_i}$, moves s.t. MBN d_i cannot cover P_{d_i} **then**
13: **create** a virtual MBN for p
14: **if** the virtual MBN cannot be *merged* with any of its neighbors **then**
15: **create** a new MBN to cover p

Define *steady state* as any point in time in which there are no merge or separate actions currently possible. Below we describe the performance of the MAS algorithm.

LEMMA 6. *In steady state, the MAS algorithm is a 2-approximation algorithm for the GDC problem within a strip.*

PROOF. We prove the lemma by induction in a similar way to the proof of Lemma 2. We assume the same definitions of OPT_S, $ALGO_S$, and b_m as in that proof.

Base Case: Consider the leftmost OPT_S disk d_1 and its member point-set P_{d_1}. Assume there exist 3 $ALGO_S$ disks that cover at least one point from P_{d_1}. However, if this was the case then all of the points covered by these 3 $ALGO_S$ disks would lie within an x-range of $2D$ (i.e. $[[d_1^L, d_1^L + 2D]$), and would be re-organized as per the separate rule of the in-strip MAS algorithm. Once re-organized, we note that since

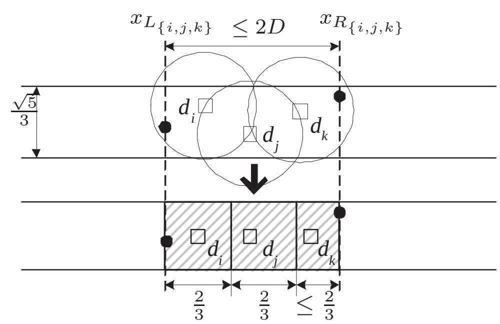

Figure 9: The Separation rule of the MAS algorithm

there exist no uncovered points left of d_1^L, that as per the separate rule of the MAS algorithm, the first 2 re-organized disks would cover all points within the x-range $[d_1^L, d_1^L + \frac{4}{3}D]$, and thus the third re-organized disk could not cover any points from P_{d_1}, which is a contradiction.

Inductive Step: Assume the in-strip MAS uses no more than $2m$ disks to cover all the points covered by $d_1, \ldots, d_m$, i.e. $b_m \le 2m$. Now consider the number of additional disks it takes for the algorithm to cover the points covered by $d_1, \ldots, d_m, d_{m+1}$. Since all of the points up to the rightmost point of d_m are already covered, by the same argument as the base case the algorithm will use at most 2 extra disks to cover the points covered by d_{m+1}. Thus, we have that,

$$b_{m+1} \le b_m + 2 \le 2m + 2 \le 2(m+1). \quad (18)$$

□

Since point transfers are local (i.e. only take place between adjacent MBNs), the time complexity is $O(1)$. The computation complexity is $O(C(n))$ to evaluate the merge and the create rules, where $C(n)$ is the running time of the 1-center subroutine used. In order to make the required decisions, we assume that an MBN has access to all nearby (i.e. within a distance of $3D$) MBNs' point-sets and locations.

6. PLACING THE RELAY MBNS

Recall that in Section 4 we showed that the CDC problem can be decomposed into two subproblems. In this section, we focus on the second subproblem that deals with a situation in which a set of nodes (*Cover MBNs*) is given and there is a need to place the minimum number of nodes (*Relay MBNs*) such that the resulting network is connected. Recall that the distance between connected MBNs cannot exceed R. This problem is equivalent to the Steiner Tree Problem with Minimum number of Steiner Points (STP-MSP) [17].

In [17] a 4-approximation algorithm that places nodes along edges of the Minimum Spanning Tree (MST) which connects the Cover MBNs has been proposed. In [2] an improved MST-based algorithm that provides an approximation ratio of 3 has been proposed. These algorithms are simple and perform reasonably well in practice. However, their main limitation is that they only find *MST-based* solutions. Namely, since the Relay MBNs are in general placed along the edges of the MST, these algorithms cannot find solutions in which a Relay MBN is used as a central junction that connects multiple other Relay MBNs. An example demonstrating this inefficiency appears in Figure 10.

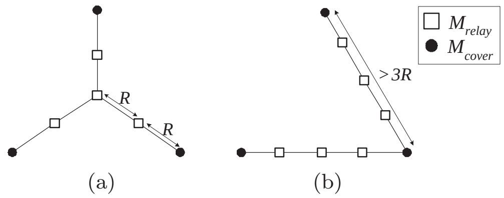

Figure 10: STP-MSP solutions: (a) Optimal (4 Relay MBNs) and (b) MST-based (6 Relay MBNs).

Below we present and analyze a *Discretization Approach* which provides a theoretical footing towards the application of the vast family of discrete and combinatorial approaches (e.g. integer programming and local search) that can potentially rectify the above inefficiency. In particular, the approach transforms the STP-MSP from a Euclidean problem to a discrete problem on a graph. Although the transformed problem does not admit a constant factor approximation algorithm, in many practical cases it can be solved optimally. We will show that if such a solution is obtained, it is 2-approximation for the STP-MSP.

Our approach is based on an idea used by Provan [18] for dealing with the continuous analog of the STP-MSP problem, the Euclidean Steiner Minimal Tree (ESMT) problem [7]. In [18] it was proposed to discretize the plane and to solve a Network Steiner Tree problem [7] on the induced graph, yielding an efficient approximate solution for the ESMT. We present a somewhat similar approach for solving the STP-MSP problem. Our approach is quite different from the approach of [18], since the STP-MSP problem is more sensitive to discretizing the plane than the ESMT problem.

Define V_0 as the lattice of points in the plane generated by gridding the plane with horizontal/vertical spacing Δ, the exact value of which will be derived later. Next, define V_1 as the set of pairwise intersection points of radius R circles drawn around each of the Cover MBNs. For the intersection region of any two circles, add three equally spaced points along the line between the two intersection points. Let V_2 denote the set of these points. Finally, define $conv(M_{cover})$ as the convex hull of the of Cover MBNs. We can now define

$$V = \left\{ (V_0 \cup V_1 \cup V_2 \cup M_{cover}) \cap^* conv(M_{cover}) \right\}. \quad (19)$$

where we define a special intersection operator $\cap^*$ to ensure that we pick enough points to be in V such that $conv(V) \supseteq conv(M_{cover})$.

For all $u, v \in V$, if $d_{uv} \leq R$, we define an edge (u, v). We denote the set of edges by E and the *induced graph* by $G = (V, E)$. Let the node weights be denoted by w_v. We now state the Node-Weighted Steiner Tree (NWST) problem [9],[16],[20], which has to be solved as part of our Discretization algorithm, presented below.

Problem NWST: Given a *node-weighted* undirected graph $G = (V, E)$ with zero-cost edges and a terminal set $M_{cover} \subseteq V$, find a minimum weight tree $T \subseteq G$ spanning M_{cover}.

The set of nodes selected in step 5 correspond to the Relay MBNs in the STP-MSP solution. We assume that step 5 is performed by a β_{NWST}-approximation algorithm. The following theorem provides the performance guarantee of the Discretization algorithm.

Algorithm 6 Discretization

1: **create** the sets V_0, V_1, V_2, and V {Δ derived below}
2: $w_v \leftarrow 1 \, \forall v \in V - M_{cover}$
3: $w_v \leftarrow 0 \, \forall v \in M_{cover}$
4: **create** the set E
5: **find** a minimum weight NWST on $G = (V, E)$

Theorem 5. *If $\Delta \leq \frac{R}{7}$, the Discretization algorithm is a $2\beta_{NWST}$-approximation algorithm for the STP-MSP.*

Our methodology in proving the theorem is as follows. We start by assuming the optimal STP-MSP tree is known, and we define an algorithm to construct a candidate Steiner tree T in G from this optimal tree. Notice that the optimal solution is of course not known, and therefore, T will not be constructed in practice. However, we will use the definition of T in order to bound the ratio between an *approximate solution* to the Node-Weighted Steiner Tree (NWST) problem in G to the *optimal solution* of the STP-MSP in the plane.

Recall that the set of terminals/Cover MBNs M_{cover} is given as input to the problem. Define $T_{OPT} = (M^*, E^*)$ as the optimal solution to the STP-MSP. The node set M^* is composed of the Cover MBNs M_{cover} and the optimal set of Relay MBNs denoted by M^*_{relay}. We now present an algorithm for the construction of a candidate tree $T = (M^T, E^T)$ in the graph $G = (V, E)$. An example of steps 4-5, 7, and 12-14 of the algorithm is illustrated in Figure 11.

Algorithm 7 Construction of a Feasible STP-MSP (CFS)

1: $M^T \leftarrow M_{cover}$
2: $E^T \leftarrow$ edges from E^* connecting M_{cover} nodes to each other
3: **for** all $u \in M^*_{relay}$ that have edges (in E^*) to a set of Cover MBNs (in M_{cover}) **do**
4:     **add** to M^T a Relay MBN $u' \in V$ located at the nearest point to u that can be directly connected to the same set of Cover MBNs
5:     **add** to E^T edges connecting u' and the set of Cover MBNs
6: **for** all $u \in M^*_{relay}$ that do not have edges (in E^*) to any Cover MBNs in M_{cover} **do**
7:     **add** to M^T a Relay MBN $u' \in V$ located at the nearest point to u
8: **for** all Relay MBNs $u, v \in M^*_{relay}$ such that $(u, v) \in E^*$ **do**
9:     **if** $d_{u'v'} \leq R$ **then**
10:         **add** to E^T an edge (u', v')
11:     **else**
12:         $w \leftarrow$ midpoint of the line segment $\overrightarrow{(u, v)}$
13:         **add** to M^T a Relay MBN $w' \in V$ located at the nearest point to w
14:         **add** to E^T edges $(u', w'), (w', v')$

In the following lemma we show that T is a feasible solution to the NWST problem in G.

Lemma 7. *If $\Delta \leq \frac{R}{7}$, then T, constructed by the CFS algorithm, is a Steiner tree in G.*

Proof. We denote the Euclidean distance between nodes u and v by $|uv|$. We have to show that T connects all the

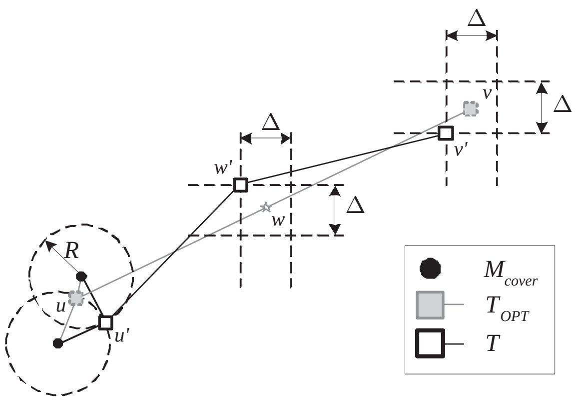

Figure 11: Example of the construction of the candidate tree T from the optimal STP-MSP tree T_{OPT}.

nodes from M_{cover} by a tree whose nodes are in V and that the edges added to E^T are valid edges in E.

The nodes of T (i.e. M^T) are by definition in V, since they are selected from V. We can see that following Step 1 all the nodes from M_{cover} are included in T. Then, in Step 2 all the M_{cover} nodes that were directly connected to each other in T_{OPT} are similarly connected in T. In steps 3-5 all the Cover MBNs that were directly connected to a relay MBN in T_{OPT} are also similarly connected to new Relay MBNs (that connect the same groups of Cover MBNs) in T. The new Relay MBNs always exist and are always less than R away from their Cover MBNs, since V includes the intersections of radius R circles drawn around each of the Cover MBNs.

Up to this point all of the edges added to E^T are clearly of length at most R. We now show that this is the case for edges connecting new Relay MBNs as well. In Step 7 each Relay MBN is replaced by a new Relay MBN which is a node in V. If two Relay MBNs are less than R from each other, they are connected in Step 10. It remains to show that if this is not the case, the new edges are shorter than R. Consider an edge (u, v), $u, v \in M^*_{relay}$ and the corresponding new edges in T - (u', w') and (w', v') (generated in Steps 13-14). We show that $|u'w'| \leq R$, thereby it is an edge in E (the proof for $|w'v'|$ is symmetric).

By the definition of the STP-MSP solution $|uw| \leq R/2$. In addition, by applying the triangle inequality to the distance between an arbitrary point w to the nearest grid point w' in V_0, we get that $|ww'| \leq \Delta$. Using these facts and the triangle inequality we have that,

$$|u'w'| \leq |u'u| + |uw| + |ww'| \leq |u'u| + \frac{R}{2} + \Delta. \qquad (20)$$

Obtaining an upper bound on $|u'u|$ requires to take into account the case in which u is directly connected to multiple Cover MBNs. In such a case in step 4, u' may potentially have to be located at a specific point in $V_1 \cup V_2$ which is not necessarily its nearest point in V. Two scenarios have to be considered. In the first scenario, u' *can* be located at a grid point in V_0. Namely, it can be placed at a grid point located in the intersection region of radius R circles centered around the Cover MBNs that u is directly connected to. In this case, $|u'u| \leq 2\Delta$, since u' can potentially be located at this (not necessarily the nearest) grid point.

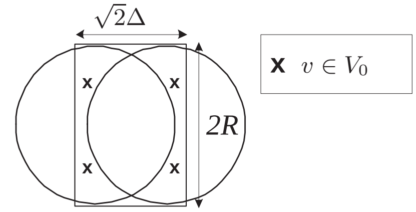

Figure 12: A loose upper bound on the area of any intersection region of 2 circles that *does not contain a grid point.*

In the second case, no point in V_0 is located in the relevant intersection region. In that case we can (loosely) bound the size of the intersection region by a $2R \times \sqrt{2}\Delta$ rectangle, as shown in Figure 12. Note that the $\sqrt{2}\Delta$ width of the rectangle corresponds the *diagonal* distance between grid points, and therefore, it is quite conservative. In the construction of V_2 we included 3 points along each line between two intersection points. Therefore, by using the triangle inequality, we get that $|u'u| \leq \Delta/\sqrt{2} + R/4$. Combining this with (20), we have that,

$$|u'w'| \quad \leq \quad \max\left(2\Delta, \frac{1}{\sqrt{2}}\Delta + \frac{R}{4}\right) + \frac{R}{2} + \Delta \qquad (21)$$

which is less than R if $\Delta \leq \frac{R}{7}$. □

In the following lemma we show that the number of Relay MBNs in T, denoted by $|M^T_{relay}| = |M^T| - |M_{cover}|$, is less than twice the number of Relay MBNs in the optimal solution of the STP-MSP (T_{OPT}).

LEMMA 8. *In T, constructed by the CFS algorithm, $|M^T_{relay}| < 2|M^*_{relay}|$.*

PROOF. In the CFS algorithm, each Relay MBN u in T_{OPT} is replaced by a Relay MBN u' in T (steps 4 and 7). For each edge connecting a pair of Relay MBNs in T_{OPT}, at most one additional MBN is added in T (w' in step 13). Since T_{OPT} is a tree, there can be at most $|M^*_{relay}| - 1$ such edges. Therefore, the total number of Relay MBNs in T is,

$$|M^T_{relay}| \leq |M^*_{relay}| + |M^*_{relay}| - 1 < 2|M^*_{relay}|. \qquad (22)$$

□

PROOF OF THEOREM 5. Let the number of Relay MBNs in T_{OPT} and T be $|T_{OPT}| = |M^*_{relay}|$ and $|T| = |M^T_{relay}|$, respectively. Recall that in the Discretization algorithm, the Cover MBNs in G were assigned a weight of 0 and the other nodes were assigned a weight of 1. Let T^{NWST}_{OPT} be the optimal (minimum weight) Node-Weighted Steiner Tree (NWST) in G and denote its weight by $|T^{NWST}_{OPT}|$. Due to Lemma 7 when $\Delta \leq R/7$, T is a feasible solution to the NWST problem in G. Therefore, and due to Lemma 8,

$$|T^{NWST}_{OPT}| \leq |T| \leq 2|T_{OPT}|. \qquad (23)$$

In Step 5 of the Discretization algorithm, the NWST problem in G is solved by a β_{NWST} approximation algorithm. We denote the obtained solution by T_{ALGO} and denote the number of Relay MBNs in this solution by $|T_{ALGO}|$. From (23) we get that

$$|T_{ALGO}| \le \beta_{NWST}|T_{OPT}^{NWST}| \le 2\beta_{NWST}|T_{OPT}|. \quad (24)$$

□

It was shown in [16] that the NWST problem does not admit a constant factor approximation algorithm and that the best theoretically achievable approximation ratio is $\ln k$, where k is the number of terminals (in our formulation $k = |M_{cover}|$). Indeed, for the case in which all node weights are equal, [9] presented a $(\ln k)$-approximation algorithm. Thus, in general, the Discretization algorithm yields a worst case approximation ratio of $2 \ln |M_{cover}|$. However, in some cases the NWST problem can be solved optimally by discrete methods such as integer programming [20]. Since in such cases $\beta_{NWST} = 1$, the approximation ratio will be 2. Notice that it is likely that the Discretization algorithm will have better average performance than the MST-type algorithms, due to the use of Relay MBNs as central junctions.

Finally, it should be noted that the Discretization algorithm is centralized. Since this algorithm takes care of placing only the *Relay MBNs*, it might be feasible to implement it in a central location. However, if there is a need to solve the problem in a distributed manner, one of the MST-based algorithms [2],[17] should be used. Although these algorithms can be implemented in a distributed manner, they do not deal very well with the mobility of Cover MBNs (i.e. a small change in the location of a Cover MBN may require repositioning several Relay MBNs). Thus, the development of distributed algorithms for the STP-MSP that take into account mobility remains an open problem.

7. JOINT SOLUTION

Using the decomposition framework presented in Section 4, the overall approximation ratio of the CDC problem is the sum of the approximation ratios of the algorithms used to solve the subproblems. Hence, this framework yields a centralized 3.5-approximation algorithm. In this section, we note that the Discretized algorithm developed in the previous section can be applied towards solving the CDC problem. Accordingly, in *specific instances* when the Node-Weighted Steiner Tree (NWST) problem can be solved optimally (e.g. using integer programming), a centralized 2-approximate solution for the CDC problem can be obtained.

The key insight is that the CDC problem can be viewed as an extended variant of the STP-MSP problem. Namely, given a set of RNs (terminals) distributed in the plane, place the smallest set of MBNs (Steiner points) such that the RNs and MBNs form a connected network. Additionally, RNs must be *leaves in the tree*, and edges connecting them to the tree must be of length at most r. The remaining edges in the tree must be of at most R.

For the Discretization algorithm to apply, we need to make the following modifications. First, in the definition of the vertex set V, M_{cover} should be replaced with the set of RNs, N. Second, V_1 and V_2 should now be defined with respect to the pairwise intersections of radius r circles drawn around each of the RNs. Finally, in the definition of the edge set E, RNs should only have edges to vertices in V within distance r, and no two RNs should have an edge between them. With these modifications, it can be shown that if $R \ge 2r$ and $\Delta \le R/6$, the Discretization algorithm is a $2\beta_{NWST}$-approximation algorithm for the overall CDC problem.

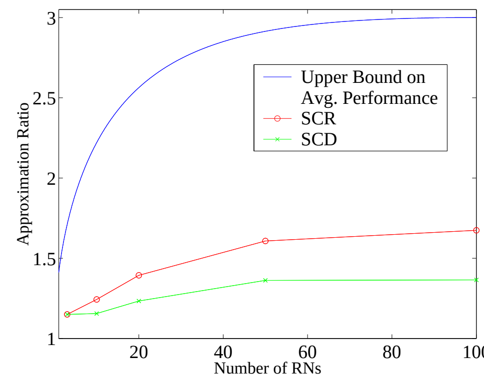

Figure 13: An upper bound on the average approximation ratios and ratios between the SCD and SCR solutions and the optimal solution.

8. PERFORMANCE EVALUATION

We now evaluate the performance of the algorithms. We start by considering the GDC algorithms. We briefly discuss the tradeoffs between the complexities and approximation ratios. We also evaluate via simulation the GDC algorithms in both static and mobile environments. Finally, we focus on the CDC problem and compare results obtained by the Discretization algorithm to results obtained by decomposing the problem. We simulated the algorithms using Java[6].

For a network with static RNs, Figure 13 presents the average ratio between the solutions obtained by the *centralized* SCD and SCR algorithms, and the optimal solution. We used a plane of dimensions $1000m \times 1000m$ and set the RNs communication range as $r = 100m$. For each data point, the average was obtained over 10 different random instances in which the RNs are uniformly distributed in the plane. The optimal solutions were obtained by formulating *each instance* of the GDC problem as an Integer Program and solving it using CPLEX. It can be seen that although the worst case performance ratios of the SCR and SCD algorithms are 6 and 4.5, the performance ratios attained in simulation are closer to 1.7 and 1.4, respectively. The figure also presents the upper bound on the average approximation ratios (β_{SCR} and β_{SCD}) derived in Theorem 4.[7] It can be seen that there is a large gap between the bound on the average approximation ratios and the actual ratios, indicating that the bound is somewhat loose.

Table 1 describes the complexities and approximation ratios of the *distributed* GDC algorithms. It can be seen that there are clear tradeoffs between decentralization and approximation. These tradeoffs are further demonstrated by simulation. Figures 14 and 15 illustrate simulation results for a network with mobile RNs. The mobility model used is the Random Waypoint Model in which RNs continually repeat the process of picking a random destination in the plane and moving there at a random speed in the range $[V_{min}, V_{max}]$, where $V_{min} = 10m/s$ and $V_{max} = 30m/s$. We

[6]An applet demonstrating some of the algorithms can be found at http://web.mit.edu/anand3/www/gdc/gdc.html.

[7]Recall that in Theorem 4, we assume that the RNs are randomly distributed according to a two dimensional Poisson process. Therefore when the number of RNs is given, their positions are uniformly distributed in the plane.

Table 1: Time complexity (number of rounds), local computation complexity, and approximation ratio of the distributed GDC algorithms ($C(n)$ is the complexity of a decision 1-center algorithm).

Algorithm	*Time Complexity*	*Local Computation Complexity*	*In-Strip Approximation Ratio*
MOAC	$O(1)$	$O(\log n)$	3
SCR	$O(n)$	$O(\log n)$	2
MAS[8]	$O(1)$	$O(C(n))$	2
SCD	$O(n)$	$O(C(n) \log n)$	1.5

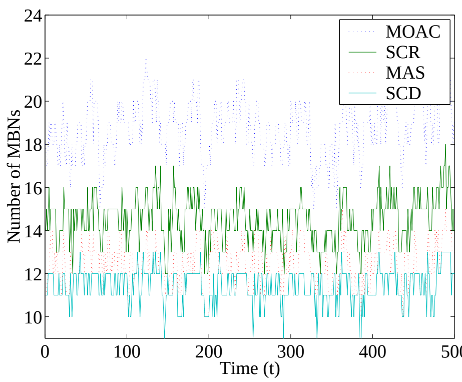

Figure 14: The number of Cover MBNs used by the GDC algorithms during a time period of $500s$ in a network of 80 RNs.

used a plane of dimensions $600m \times 600m$ and set $r = 100m$.

Figure 14 depicts an example of the evolution (over a $500s$ time period) of the required number of MBNs used by the different GDC algorithms in a network with 80 RNs. Note that we simulated a $1000s$ time period and discarded the first 500 seconds. As expected, the most distributed and least computationally complex algorithm (MOAC) performs the poorest, and the least distributed and most computationally complex algorithm (SCD) performs the best. Moreover, both algorithms that utilize 1-center subroutines (MAS and SCD) perform better than the MOAC and SCR algorithms, which reduce disks to rectangles. Figure 15 presents the average number of MBNs used over a $500s$ time period as a function of the number of RNs. Each data point is an average of 10 instances (each instance was simulated over $1000s$ from which the first $500s$ were discarded).

Next we compare solutions of the CDC problem obtained by the decomposition framework to joint solutions obtained by the Discretization algorithm. Figure 16 depicts a random example of 10 RNs distributed in a $1000m \times 1000m$ area.[9] The communication ranges of the RNs and the MBNs are $r = 100m$ and $R = 200m$, respectively. In the decomposition framework, we used an optimal disk cover (obtained by integer programming) and the 3-approximation STP-MSP

[8]The approximation ratio of the MAS algorithm holds when the algorithm is in steady state.

[9]We deliberately selected a small number of RNs in order to generate a partitioned network that requires Relay MBNs.

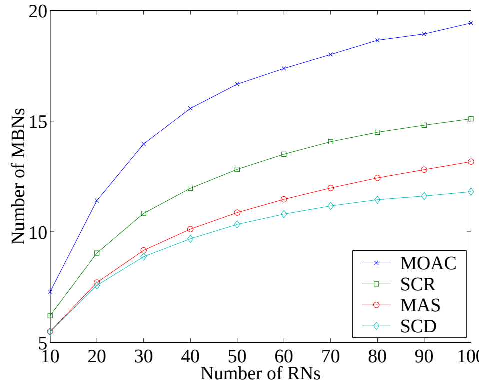

Figure 15: The average number of Cover MBNs used by GDC algorithms over a time period of $500s$.

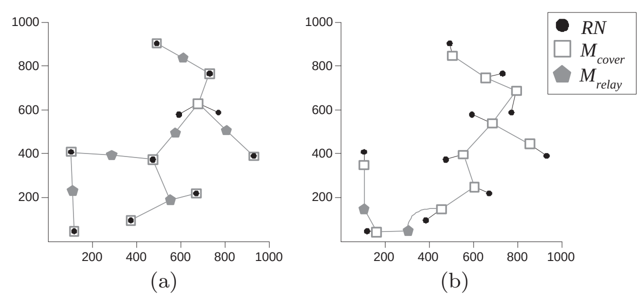

Figure 16: An example comparing solutions obtained by (a) an optimal Disk Cover and the STP-MSP algorithm from [2], and (b) the Discretization algorithm using an NWST algorithm [16].

algorithm from [2]. The Discretization algorithm uses the NWST approximation algorithm from [16]. In this example, the joint solution requires 12 MBNs while the decomposition based solution requires 15 MBNs .

Figure 17 presents similar results for a more general case with the same parameters (area, r, and R). The Decomposition framework used the SCD algorithm along with the MST algorithm [17] and along with the Modified MST-based algorithm [2]. Each data point is averaged over 10 random instances. It can be seen that the joint solution provides a significant performance improvement (about 25% for large number of RNs). Yet, while the decomposition framework uses distributed algorithms, the joint solution must be obtained in a centralized manner. Thus, a reasonable compromise could be to place the Cover MBNs in a distributed manner and to place the Relay MBNs (e.g. Unmanned-Aerial-Vehicles) by a centralized Discretization algorithm.

9. CONCLUSIONS

The architecture of a hierarchical Mobile Backbone Network has been presented only recently. Such an architecture can significantly improve the performance, lifetime, and reliability of MANETs and WSNs. In this paper, we concentrate on placing and mobilizing backbone nodes, dedicated to maintaining connectivity of the regular nodes. We have formulated the Mobile Backbone Nodes placement problem

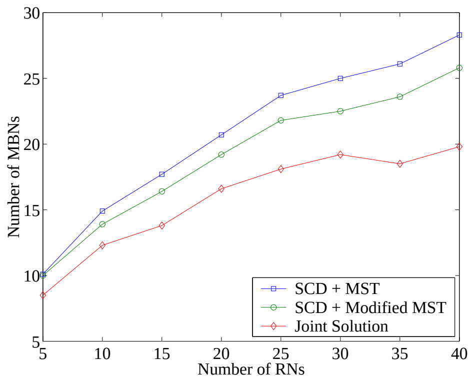

Figure 17: Number of MBNs as a function of the number of RNs computed by: (i) the decomposition approach using the SCD with the MST-based [17] algorithms, (ii) the decomposition approach using the SCD with the modified MST-based [2] algorithm, and (iii) the Discretization algorithm.

as a Connected Disk Cover problem and shown that it can be decomposed into two subproblems. We have proposed a number of *distributed* algorithms for the first subproblem (Geometric Disk Cover), bounded their worst case performance, and studied their performance under mobility via simulation. As a byproduct, it has been shown that the approximation ratios of algorithms presented in [8] and [11] are 6 and 2 (instead of 8 and 3 as was shown in the past). A new approach for the solution of the second subproblem (STP-MSP) and of the joint problem (CDC) has also been proposed. We have demonstrated via simulation that when it is used to solve the CDC problem in a centralized manner, the number of the required MBNs is significantly reduced.

This work is the first approach towards the design of distributed algorithms for construction and maintenance of a Mobile Backbone Network. Hence, there are still many open problems to deal with. For example, moving away from the strip approach may be beneficial. Moreover, there is a need for distributed algorithms for the STP-MSP, capable of dealing with Cover MBNs mobility. A major future research direction is to generalize the model to other connectivity constraints and other objective functions. For instance, we intend to extend the results to connectivity models that are more realistic than the disk connectivity model.

Acknowledgments

This research was partially supported by Draper Laboratories, Korea University, and Samsung Corporation. The research of Gil Zussman was supported by a Marie Curie International Fellowship within the 6th European Community Framework Programme.

10. REFERENCES

[1] D.J. Baker and A. Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Trans. Commun.*, 29(11):1694-1701, 1981.

[2] D. Chen, D.-Z. Du, X.-D. Hu, G.-H. Lin, L. Wang, and G. Xue. Approximations for steiner trees with minimum number of steiner points. *J. Global Optimization*, 18(1):17-33, 2000.

[3] X. Cheng, D.-Z. Du, L. Wang, and B. Xu. Relay sensor placement in wireless sensor networks. To appear in *ACM/Springer WINET*, 2005.

[4] B. Das, R. Sivakumar, and V. Bharghavan. Routing in ad-hoc networks using a virtual backbone. In *Proc. IEEE ICCCN'97*, September 1997.

[5] M. Franceschetti, M. Cook, and J. Bruck. A geometric theorem for approximate disk covering algorithms. *Technical Report ETR035*, Caltech, 2001.

[6] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Discrete mobile centers. *Disc. Comput. Geom.*, 30(1):45-63, 2003.

[7] M.R. Garey and D.S. Johnson. *Computers and Intractibility*, Freeman, 1979.

[8] T. Gonzalez. Covering a set of points in multidimensional space. *Inf. Proc. Letters*, 40(4):181-188, 1991.

[9] S. Guha and S. Khuller. Improved methods for approximating node weighted steiner trees and connected dominating sets. *Inf. Comput.*, 150(1):57-74, 1999.

[10] H. Gupta, S. Das, and Q. Gu. Connected sensor cover: self organization of sensor networks for efficient query execution. In *Proc. ACM MOBIHOC'03*, June 2003.

[11] J. Hershberger. Smooth kinetic maintenance of clusters. In *Proc. ACM SoCG'03*, June 2003.

[12] J. Hershberger and S. Suri. Finding tailored partitions. *J. Algorithms*, 12(3):431-463, 1991.

[13] D.S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130-136, 1985.

[14] H. Huang, A. Richa, and M. Segal. Approximation algorithms for the mobile piercing set problem with applications to clustering in ad-hoc networks. *ACM/Kluwer MONET*, 9(2):151-161, 2004.

[15] A. Kansal, M. Rahimi, D. Estrin, W.J. Kaiser, G. J. Pottie, and M. B. Srivastava. Controlled mobility for sustainable wireless sensor networks. In *Proc. IEEE SECON'04*, October 2004.

[16] P. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted steiner trees. *J. Algorithms*, 19(1):104-115, 1995.

[17] G. Lin and G. Xue. Steiner tree problem with minimum number of steiner points and bounded edge-length. *Inf. Proc. Letters*, 69(2):53-57, 1999.

[18] J.S. Provan. Convexity and the steiner tree problem. *Networks*, 18(1):55-72, 1988.

[19] I. Rubin, A. Behzad, R. Zhang, H. Luo, and E. Caballero. TBONE: a mobile-backbone protocol for ad hoc wireless networks. In *Proc. IEEE Aerospace Conf.*, March 2002.

[20] A. Segev. The node-weighted steiner tree problem. *Networks*, 17(1):1-17, 1987.

[21] C. Swamy and A. Kumar. Primal-dual algorithms for connected facility location problems. *Algorithmica*, 40(4):245-269, 2004.

[22] K. Xu, X. Hong, and M. Gerla. Landmark routing in ad hoc networks with mobile backbones. *J. Parallel Distrib. Comput.*, 63(2):110-122, 2003.