# POLYNOMIAL TIME EXACT SOLUTIONS FOR FORWARDING SET PROBLEMS IN WIRELESS AD HOC NETWORKS

by

Mehmet Baysan

APPROVED BY SUPERVISORY COMMITTEE:

R. Chandrasekaran, Chair

Kamil Sarac, Co-Chair

Ravi Prakash

Sergey Bereg

Dedicated to my family.

Especially to my Father Arif, my mother Habibe and my wife Ayşe Nur.

# POLYNOMIAL TIME EXACT SOLUTIONS FOR FORWARDING SET PROBLEMS IN WIRELESS AD HOC NETWORKS

by

MEHMET BAYSAN, B.S., M.S.

DISSERTATION
Presented to the Faculty of
The University of Texas at Dallas
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS
December, 2008

# ACKNOWLEDGEMENTS

First of all I would like to thank God Almighty for the bounties he provided for my whole life. Then I want to express my gratitude to his messenger for bringing us the good news of this life and hereafter.

I am forever indebted to my family without whose constant support, understanding and encouragement none of my academic goals would have been realized.

I would like to thank to my advisor Dr. Chandrasekaran for all his support and guidance. He set an excellent example for me as an academician. I would also like to thank my co-advisor Dr. Sarac for his invaluable comments to help shaping my academic career. I would like to thank Dr. Prakash and Dr. Bereg for helping me with my research.

I would like to thank my colleague Mehmet Gunes whose presence as a labmate made my life much easier. I am grateful to all my friends in Dallas for having a happy life together.

August 2008

# POLYNOMIAL TIME EXACT SOLUTIONS FOR FORWARDING SET PROBLEMS IN WIRELESS AD HOC NETWORKS

Publication No. __________________

Mehmet Baysan, Ph.D.
The University of Texas at Dallas, 2008

Supervising Professors: R. Chandrasekaran, Chair
Kamil Sarac, Co-Chair

Network-wide broadcast (simply broadcast) is a frequent operation in wireless ad hoc networks. A simple flooding based implementation of broadcast may result in excessive use of system energy and wireless bandwidth that are two scarce resources in wireless ad hoc networks. One promising practical approach to improve the efficiency of broadcast is to use localized algorithms that minimize the number of nodes involved in the propagation of broadcast messages. In this context, the minimum forwarding set problem (MFSP) (also known as multi-point relay (MPR) selection problem) has received a considerable attention in the research community. Even though the general form of the problem is shown to be NP-complete, the complexity of the problem has not been known under the practical application context of ad hoc networks.

In this study, I present two polynomial time algorithms to solve the MFSP for wireless networks under unit disk coverage model and disk coverage model. Leveraging the practical characteristics of the application environment, I prove the existence of a certain class of optimal solutions that hold some nice properties. I then propose polynomial time algorithms to build an optimal solution within this class of solutions to a given instance of the MFSP

problem. My algorithms are the first polynomial time solutions to the MFSP problem under these models.

Furthermore I present a new version of MFSP which provides collision awareness. I believe that the work presented in this thesis will have an impact on the design and development of new algorithms for several wireless network applications including energy efficient multicast and broadcast protocols; energy efficient topology control protocols; and energy efficient virtual backbone construction protocols for wireless ad hoc networks and sensor networks.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1
# INTRODUCTION

Wireless ad hoc networks (WANETs) consist of nodes that can communicate with each other via wireless channel. Since there is no static backbone, nodes communicate with distant nodes by using other nodes in the network as relays. Wireless Sensor Networks and Mobile Ad Hoc Networks (MANETs) are two well-studied examples for such networks. Due to their ease-of-deployment and support for mobility, WANETs are useful for many environments including active battlefield, search and rescue, and emergency relief. Energy and wireless bandwidth are two scarce WANET resources that need to be used efficiently. Energy is a limited source as the nodes typically operate on battery power. Wireless bandwidth is also a limited resource as the nodes share the same transmission medium which is open to collision and contention. In addition, the mobility of the nodes makes efficient communication challenging as the nodes spend power and bandwidth resources in exchanging control messages to adjust for the changes in the topology.

Network-wide broadcast (simply broadcast) is defined as the delivery of a certain packet to all nodes in the network. In addition to data dissemination, many proactive and reactive protocols utilize broadcast to communicate control messages [34, 24, 16, 36]. Proactive protocols like Optimized Link State Routing (OLSR) [16] maintain routes between nodes of the WANET. This way whenever a node wants to send a message to another node it uses the pre-established route to send the message. Reactive protocols like Dynamic Source Routing (DSR) [24] and Ad Hoc On Demand Distance Vector Routing (AODV) [33] do not maintain routes and routes are established whenever they are needed.

The simplest way of doing broadcasting is pure-flooding, where each node rebroadcasts the received broadcast message exactly once. Pure-flooding has major drawbacks.

Since each node broadcasts the packet, there is a high risk of many redundant rebroadcasts, collisions and contention. Neighboring nodes will get broadcast messages simultaneously and they will try to retransmit in parallel. Such parallel retransmissions will cause many collisions and contention in the network. As a result of these collisions there is a potential risk of reduction in the reachability (number of nodes getting the message) of the broadcast operation. The negative effects of pure-flooding is examined under the name of the broadcast-storm problem [31] in the literature.

There are two main approaches to design and implement efficient algorithms to support broadcast operation in WANETs. One approach is aiming to solve the problem for the whole network and developing centralized or distributed algorithms for the optimization. The other approach is defining local problems which requires 1-hop or 2-hop neighborhood information. In this approach local problems should be designed in a way that their optimal solutions should lead to the global optimal solution for the whole network.

The localized approach provides certain advantages over the centralized approach with respect to the nature of WANETs. First of all, it is a difficult task to collect accurate information about the whole network in a small amount of time. After the collection of information, the algorithms should be executed and nodes should be informed about the decisions have been taken. During this time, it is likely that the topology will change. In this case, the solution developed might not even be feasible when nodes actually execute the algorithm. On the other hand, the flexibility provided by the local algorithms make them suitable for dynamic environments of WANETs. The problem with local algorithms is the difficulty in determining the relation between developed local solutions and their effects on the global objective function. Despite application-wise being efficient, it is difficult to give worst-case bounds for localized algorithms. As a result of this, localized algorithms are best effort solutions in general.

In this thesis, I present the first polynomial time algorithm to solve the MFSP problem for the unit disk and disk coverage model. Furthermore, I present a new version of MFSP which provides a scheme to handle the effect of collisions.

It is necessary to make certain assumptions to design efficient algorithms, as it is difficult to model the random nature of wireless communication. Under unit disk coverage model, the wireless coverage range of each node is represented by a unit disk, I first introduce two geometrical properties named as *Two-Set Property* and *Non-Interleaving Property.* I then present an algorithm that uses a dynamic programming approach to build an optimal solution and prove its correctness. The algorithm has $O(n^3 + n^2m)$ time complexity where $m = |N(v)|$ and $n = |N_2(v)|$ for a broadcasting node $v$. I have also used this algorithm as an instrument in evaluating the performance of more practical heuristics with simulation studies.

Under disk coverage model, the wireless coverage range of each node is represented by a disk. I presented a solution for this model based on non-interleaving property (see Section 4.1.4). After proving this property for disc coverage model, I use a dynamic programming algorithm to construct an optimal solution to the problem. The disc coverage model is a more general model compared unit disk coverage model. This causes an increase in the time complexity to $O(n^5m)$ for the disc coverage model. Even though disc model is more general, it may not be applicable for all real world wireless network environments. Finally, I design a new problem to handle the effect of collisions. I show the NP-completeness of this problem and present heuristics with evaluations as practical solutions.

In addition to efficient selection of relay nodes, a successful broadcast algorithm should address a number of other issues including packet scheduling, broadcast reliability and network coverage. These issues require a holistic approach where the protocol takes into account the characteristics of the underlying physical layer and MAC layer protocol. In this study, I consider the efficient selection of relay nodes as an important building block for a successful broadcast algorithm and leave the other issues as possible future work. Note that, the efficient selection of relay nodes can be helpful in other communication application contexts such as building global structures (e.g., connected dominating set based structures) for energy efficient communication [1, 37] or implementing efficient route update dissemination as in the case of OLSR ad hoc routing protocol [16].

# CHAPTER 2
# BACKGROUND

## 2.1 Problem Definition

Energy efficient broadcast problem received significant attention from the research community and a large number of studies have been published in the area [23]. One promising approach that was proposed for energy efficient broadcast is the *neighbor designation* approach [35] where the goal is to prevent unnecessary transmission of broadcast packets for energy efficiency. Each node collects 2-hop neighborhood information and then identifies a subset of its 1-hop neighbors as forwarding nodes for relaying a broadcast message toward its 2-hop neighbors. This way, the neighbor designation approach saves both energy and bandwidth resources in the network. In addition, due to its dependence on local 2-hop neighbor information, the approach can quickly adapt to topology changes. The efficiency of neighbor designation approach depends on finding a minimum size forwarding node set among the 1-hop neighbors. This problem is referred to as *Minimum Forwarding Set Problem (MFSP)* [35, 11] and is formally defined as follows:

**Definition 2.1 (Minimum Forwarding Set Problem (MFSP))** *Consider a graph* $G = (V, E)$ *where* $V$ *is the set of nodes and* $E$ *is the set of links in the network. Given a node* $v \in V$*, let* $N(v)$ *and* $N_2(v)$ *represent the set of 1-hop and 2-hop neighbors of* $v$*, respectively.* $N(v)$ *and* $N_2(v)$ *are strict sets such that* $v \notin N(v)$ *and* $N(v) \cap N_2(v) = \varnothing$*. MFSP asks for a minimum-size subset* $S$ *of* $N(v)$ *such that every node in* $N_2(v)$ *is within the coverage of at least one node in* $S$*. More formally, MFSP asks for a minimum cardinality set* $S$ *such that* $S \subseteq N(v)$ *and* $(\forall x \in N_2(v),\ \exists y \in S \mid x \in N(y))$*.*

A solution to the MFSP problem at a node $v$ is $S \subseteq N(v)$ where $S$ is a minimum cardinality set called *forwarding set.* Note that, in certain cases, multiple different optimal solutions may exist.

## 2.2 Existing Solutions

The MFSP problem is shown to be NP-complete [35] with a reduction from the Set Cover problem. The heuristic proposed in [35] is an application of the well-known Chvatal's greedy algorithm for the Set Cover problem [13] and gives an approximation ratio of $(1+ln(|S_i|_{max}))$ where $|S_i|_{max}$ is the size of the largest subset of $N_2(v)$ that is covered by a node $i \in N(v)$.

The general case of the MFSP problem does not consider the practical characteristics of the WANET environment. More specifically, it does not take the geometric properties of the wireless transmissions into account.

More recently, Calinescu et al. [11] studied the problem under the assumption that nodes are distributed in 2-dimensional plane and they have a fixed coverage range for their transmissions, i.e., a unit disk coverage [14]. They proposed a 6-approximation algorithm that runs in $O(n\log n)$ time and a 3-approximation algorithm that runs in $O(n\log^2 n)$ time. In addition, they presented an exact $O(n\log^2 n)$ time algorithm for a special case of the MFSP problem when all 2-hop neighbors are in the same quadrant of a 2-dimensional coordinate space with respect to the broadcasting node.

## 2.3 Related Work

The general case of the MFSP problem is an instance of the well-known NP-complete Set Cover problem [35]. Set Cover problem has been extensively studied in the literature and early approximation algorithms have been proposed for both unweighted version by Johnson [25] and by Lovasz [28], and for weighted version by Chvatal [13]. These algorithms give an approximation ratio of $1 + ln(\Delta)$ where $\Delta$ is the cardinality of the maximum cardinality subset ($\max_{i \leq n} |S_i|$). In [19], Hochbaum presents an algorithm for the weighted version with an approximation ratio of $\alpha$ where $\alpha$ represents the maximum number of subsets covering an element. The running time of this algorithm is $O(n^3)$. In [5], Bar-Yehuda and Even present an algorithm with a similar approximation ratio but an improved running time of $O(n^2)$. We refer readers to [32] for other approximation algorithms on the Set Cover problem.

The initial work on the MFSP problem was by Qayyum et al. [35]. They extended Chvatal's heuristic [13] by introducing a pre-processing step in which they select those one-hop neighbors of $v$ as relay nodes that are the only neighbors of some node in $N_2(v)$, and add these one-hop nodes to the solution set $MFS(v)$. Similar to Chvatal's heuristic, this heuristic considers the general case of the MFSP problem and is therefore applicable for the Set Cover problem.

The general case of the MFSP problem does not consider the practical characteristics of the ad hoc network environment. More specifically, it does not take the geometric properties of the wireless transmissions into account. From the practical context of the problem, we know that the coverage characteristics of wireless transmission is not completely random. Even though wireless signal propagation is effected by various external/physical factors, it has been a common practice to model wireless coverage by an arbitrary or a unit disk [26],[27].

Minimum Forwarding Set Problem becomes a geometrical problem when we use disks to model the coverage areas of wireless nodes. Unit Disk Graphs (UDGs) and Disk Graphs (DGs) have been studied extensively in the literature.

In general we define Disk Graphs(DG) as the intersection of a set of disks in the Euclidean plane. If the diameter ratio of disks is bounded by some constant $\sigma$ we define this as a $\sigma$-disk graph ($\sigma$-DG). Unit Disk Graph (UDG) is a special case where $\sigma = 1$. Recognition problem of UDG and $\sigma$-DG is shown to be NP-hard in [7] and [18]. Since it is not trivial to find an embedding to the Euclidean plane, there is a distinction between the two cases of a problem if the location of the disks are given in the input or not.

Most of the NP-complete problems remain so in UDGs and $\sigma$-DGs such as Independent Set and Coloring, even the disks are given in the input [15].An exception is the clique problem, there is a polynomial time solution for UDGs [15]. Unfortunately this result does not extend to $\sigma$-DGs, the complexity of the problem is still open.

The amount of results known for UDGs is much more compared to DGs. Because of that we present the main results which are related to our problem for UDGs. Note that UDGs are special DGs, as a result negative results for UDGs are also valid for DGs.

UDGs are neither perfect nor planar graphs [15]. They are not perfect because odd cycles of length 5 or greater are unit disk graphs. They are not planar because cliques of size 5 or greater are unit disk graphs. Thus efficient algorithms proposed for planar and perfect graphs cannot be applied to unit disk graphs.

MFSP problem resembles to the well-known Minimum Dominating Set (MDS) problem. It has been shown that MDS problem is an NP-complete problem for UDGs [14] but accepts polynomial time approximations [29]. In [29], Marathe et al. present a linear time approximation algorithm with a performance guarantee of 5 for the MDS problem for UDGs. In [22], a $((k+1)/k)^2$ guarantee is given for a constant $k$ in $n^{O(k^3)}$ by using a *shifting strategy* similar to [4]. In [4], the author uses this strategy to develop approximations for planar graphs.

Minimum Connected Dominating Set (MCDS) problem is a different version of the problem in which the dominating set should be connected. In [12], Cheng et al. presented a PTAS for MCDS problem. In [2], Ambuhl et al. presented constant-factor approximation algorithms for the weighted versions of MDS and MCDS problems.

These approximations do not apply to MFSP problem as the dominating nodes in MFSP should be chosen from only 1-hop neighbors.

Another related problem to MFSP problem is *covering with disks* which aims at finding a minimally sized set of unit disks to cover given points on the plane (disks can be placed arbitrarily). This problem is examined in [20] and a $O(l^2(l*\sqrt{2})^2.(2n)^{2(l\sqrt{2})^2+1})$ time approximation is given with a performance guarantee of $(1+1/l)^2$. This approximation also uses *shifting strategy*. The difference between this problem and our problem is in the selection of the disks. This problem selects arbitrary disks to cover given points, but in our problem we are bound to select disks from the set of 1-hop neighbors.

MFSP is a special case of *disc cover* problem [8]. Disc cover problem is defined as; for a given set $P$ of $n$ points and a given set $D$ of discs given on the plane, find a subset $M$ of $D$ with minimum cardinality which covers all points in $P$. In [8] authors refer to [21] for the definition of this problem and compare their results with results given there. Actually these two problems are different because in one of them available discs are given and in the other one arbitrary discs can be used. The running time of approximation given in [8] is $O(c^2 n \log n \log(n/c))$ where $c$ represents the size of the optimal solution and the guaranteed approximation ratio is $O(1)$. Actually despite being constant the guaranteed approximation factor is quite large. Their algorithm is based on the fact that the system given by the *disc cover* problem admits a $O(r)$ sized $(1/r)$-net. This fact is shown in [30], and the actual guarantee on that paper is a $60\lceil r\rceil - 10$ sized $(1/r)$-net for the given system. As a result guaranteed approximation factor in [8] is 60.

# CHAPTER 3

# MFSP FOR UNIT DISK GRAPHS

## 3.1 Preliminaries

In this section, we first present some preliminary information on the practical context of the MFSP problem. We then present observations on some geometric relations about intersecting unit disks. We also introduce some definitions that we use in the rest of the chapter.

### 3.1.1 Problem Formulation

Most studies use a unit disk or a sphere to represent the shape of the effective coverage area of wireless transmissions [27]. This assumption, though may not always hold in practice, helps in gaining more insight to the problem within the practical context of wireless transmissions. The previous work by Calinescu et al. [11] indicates that the complexity of the MFSP problem under unit disk coverage model may not necessarily be the same as the general form of the problem. However, to the best of our knowledge, no results have been published on the exact solution of the MFSP problem under the unit disk coverage model. In this chapter we consider a similar setup for the problem and assume a unit disk coverage model for the coverage areas of wireless transmissions. In addition, as most local knowledge based broadcast approaches [23], our approach requires the availability of 2-hop neighborhood information. The required information includes (1) the identities of the 1-hop and 2-hop neighbors and (2) a radial ordering (which we define in Section 3.1.2) of the 2-hop neighbors with respect to the broadcasting node. The availability of the position information for the nodes is sufficient to compute the radial ordering of the 2-hop neighbors. One simple way of acquiring the position information is to use a GPS unit at each node. Another possibility is to use the distance and angle information between the neighboring nodes. The distance information can be calculated by using the transmission and reception power level within an energy consumption model [17] that is representative for the environment. The angle information between neighboring nodes can be measured by using multiple ultrasound receivers or directional antennas. It was shown that neither the distance [3] nor the angle [9] information is sufficient alone to find a valid embedding for the nodes on a plane. However, it is trivial to find a valid embedding by combining both. Recently, Calinescu [10] proposed methods to calculate 2-hop neighborhood information (identities and positions) for the cases where GPS or distance and angle information is available with a message complexity of $O(n)$ where $n$ is the total number of the nodes in the network.

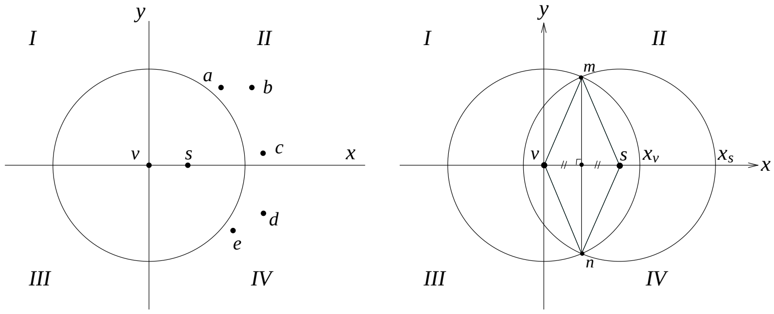
### 3.1.2 Definitions

Consider an instance of MFSP problem at a node $v$ with $N(v)$ and $N_2(v)$ representing 1-hop and 2-hop neighbors of $v$.

**Definition 3.1 (Radial Order)** *Radial order is the ordering of the nodes in $N_2(v)$ by using the angle that they make with $v$. Radial order is a cyclic order. If two or more nodes make the same angle with $v$, then their distance to $v$ can be used to put them into a total order.*

Consider the example scenario in Figure 3.1-(a) where $N_2(v) = \{a, b, c, d, e\}$. Starting from the exact south position, the nodes in $N_2(v)$ form a radial order as $(e < d < c < b < a)$. The geometrical properties introduced below and the algorithm presented in Section 3.3.1 use the radial ordering of the nodes in $N_2(v)$ in finding an optimal solution. As we discussed in Section 3.1.1, a node $v$ can compute the radial ordering of the nodes in $N_2(v)$ from the collected geographical location information from its neighbors. Therefore, from now on we assume that the radial ordering of the nodes in $N_2(v)$ is known by $v$.

**Definition 3.2 (Radially Continuous Neighbor (RCN) Interval)** *One or more elements of the set $N_2(v)$ that form a continuous interval in the radial order with respect to (w.r.t.) $v$ are said to form a radially continuous neighbor (RCN) interval.*



(a) (b)

Figure 3.1. Some geometric relations of two intersecting unit disks.

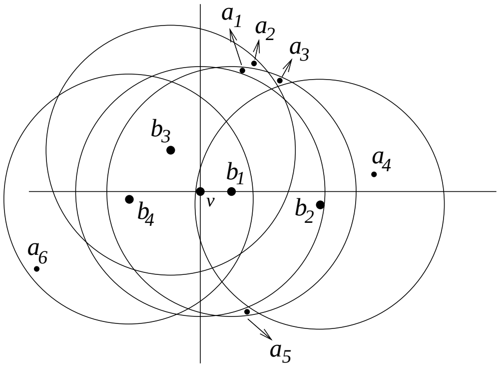
As an example, in Figure 3.1-(a), $(a > b > c)$ form a RCN interval w.r.t. $v$ but $(a > b > d)$ does not as $c \in N_2(v)$ separates this interval into two non-consecutive intervals.

**Definition 3.3 (Radially Continuous Coverage Area (RCCA))** *RCCA is a coverage area that a node $s \in N(v)$ may have outside the coverage area of $v$. RCCA of a node $s \in N(v)$ defines the area that an RCN interval of $s$ resides. A node $s$ may have multiple RCCAs.*

**Definition 3.4 (Connectivity Matrix)** *Consider an instance of MFPS problem at a node $v$. Let $N(v) = \{b_1, b_2, \ldots, b_m\}$ and $N_2(v) = \{a_1, a_2, \ldots, a_n\}$ be the 1-hop and 2-hop neighbors of $v$ respectively. A connectivity matrix $R$ is an $m \times n$ matrix that shows the connectivity relation between the nodes in $N(v)$ and $N_2(v)$. For a given $b_i \in N(v)$ and $a_j \in N_2(v)$, $R_{i,j} = 1$ if $a_j \in N(b_i)$ and $R_{i,j} = 0$ otherwise.*

**Definition 3.5 (Coverage Matrix)** *Let $m = |N(v)|$ and $n = |N_2(v)|$ for a node $v$. Using two hop neighborhood information, $v$ generates a coverage matrix as a $m \times n$ matrix $C$. Each row in $C$ corresponds to a 1-hop neighbor of $v$ and each column corresponds to a 2-hop neighbor of $v$. An entry $C_{ef} = (a_p, \bar{p})$ represents the longest RCN interval in $N_2(v)$ that is covered by $e \in N(v)$ and that includes $f$ in it. If $f \notin N(e)$, then $C_{ef} = \varnothing$.*

Please see Figure 3.2 for an example of connectivity and coverage matrices for a node $v$.



*Connectivity Matrix*

|  | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ |
|---|---|---|---|---|---|---|
| $b_1$ | 1 | 0 | 1 | 0 | 1 | 0 |
| $b_2$ | 0 | 0 | 0 | 1 | 0 | 0 |
| $b_3$ | 1 | 1 | 0 | 0 | 0 | 0 |
| $b_4$ | 0 | 0 | 0 | 0 | 0 | 1 |

*Coverage Matrix*

|  | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ |
|---|---|---|---|---|---|---|
| $b_1$ | $(a_1,1)$ | $\varnothing$ | $(a_3,1)$ | $\varnothing$ | $(a_5,1)$ | $\varnothing$ |
| $b_2$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $(a_4,1)$ | $\varnothing$ | $\varnothing$ |
| $b_3$ | $(a_1,2)$ | $(a_1,2)$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ |
| $b_4$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $\varnothing$ | $(a_6,1)$ |

Figure 3.2. An MFPS instance at $v$ and its connectivity and coverage matrices.

**Definition 3.6 (Maximum Coverage Interval (MCI))** *An MCI of a node $s \in N(v)$ is an RCN interval $\{a_i, \ldots, a_j\} \in N_2(v)$ that is completely covered by $s$ such that $s$ cannot cover neither of $a_{i-1}$ and $a_{j+1}$. Note that $s$ can have multiple MCIs in $N_2(v)$.*

**Definition 3.7 (Essential Coverage)** *In an instance of an MFSP problem at a node $v$, a node $s \in N(v)$ that covers a node $a_i \in N_2(v)$ is essential if no other node $t \in N(v)$ covers $a_i$. If $s$ is essential for a node $a_i$ in an MCI that it covers, than $s$ is essential for this MCI.*

A node $s \in S$ is said to be essential for a node $a_i \in N_2(v)$ if for all $t \in S$ $a_i \notin N(t)$. Note that there may be another node $t' \in N(v)$ such that $a_i \in N(t')$ but $t' \notin S$.

### 3.1.3 Intersection Characteristics of Two Unit Disks

Consider a unit disk centered at a point $v$. Let $D_v$ represent the unit disk and the area covered by it. Let $C_v$ represent the circle enclosing $D_v$. Assume that $v$ is at the origin of a two-dimensional space which is divided into four sub-spaces, named as $I$, $II$, $III$, and $IV$, by the $x$ and $y$ coordinate axes as shown in Figure 3.1-(a). Consider a second point $s$ that is on the $x$-axis to the east of $v$. Similar to the case for $v$, let $D_s$ and $C_s$ represent the unit disk and the enclosing circle for $s$.

When $|vs| = 2$, $C_v$ and $C_s$ are tangent to each other and $D_v \cap D_s = \varnothing$. When $|vs| < 2$, $C_v$ and $C_s$ intersect twice and $D_v \cap D_s \neq \varnothing$ as shown in Figure 3.1-(b). Let $m$ and

$n$ represent the intersection points of $C_v$ and $C_s$. Note that, since $s$ is on $x$-axis to the east of $v$, $m$ and $n$ have to be to the east of $y$-axis. This intersection forms two equal angles as $\widehat{mvn}$ and $\widehat{msn}$. Let $\alpha$ represent these two equal angles. When $|vs| \leq 1$, $\alpha$ is in $[\frac{2\pi}{3}, \pi]$ and when $1 < |vs| \leq 2$, $\alpha$ is in $[0, \frac{2\pi}{3})$. Also note that when $|vs| \leq 1$, the length of the arc $arc(mx_v n)$ (the segment of $C_v$ corresponding to $\alpha$ and $x_v$ is the intersection of $C_v$ with $x$ axis) is in $[\frac{2\pi}{3}, \pi]$. Similarly, the length of the arc $arc(mx_s n)$ (the segment of $C_s$ corresponding to $2\pi - \alpha$ and $x_s$ is the intersection of $C_s$ with $x$ axis) is in $[\pi, \frac{4\pi}{3}]$. Finally, note that the line segment connecting $m$ and $n$ vertically divides the line segment between $v$ and $s$ into two equal parts. In other words, when $C_v$ and $C_s$ intersect twice, the intersection points $m$ and $n$ can be found by drawing a vertical line from the midpoint of the line segment connecting $v$ and $s$ (see Figure 3.1-b). We refer to this vertical line as $Chord_{vs}$.

### 3.1.4 Intersection Characteristics of Three Unit Disks

In this section, we study the intersection characteristics of three unit disks in a special setup that is relevant to the MFSP problem. Consider three points $v$, $s$, and $t$ where $v \neq s \neq t$. Similar to the above discussion, we assume that $v$ defines a two-dimensional coordinate space and $s$ lies to the exact east of $v$ in this coordinate space. We assume that both $s$ and $t$ are 1-hop neighbors of $v$, i.e., $\{s, t\} \in N(v)$, and study the intersection characteristics of $C_s$ and $C_t$ outside of the coverage area of $D_v$. Note that since $s$ and $t$ are neighbors of $v$, $|vs| \leq 1$ and $|vt| \leq 1$. This also indicates that $|st| \leq 2$. Hence, $C_s$ and $C_t$ intersect twice and $D_s \cap D_t \neq \varnothing$. Note again that when $|st| = 2$, $C_s$ and $C_t$ are tangent to each other, $D_s \cap D_t = \varnothing$. Observe that since $s$ lies to the east of $v$, the coverage area of $D_s$ beyond $D_v$ (i.e., $D_{s/v} = D_s \setminus D_v$) lies in region $(II \cup IV)$. Similarly, let $D_{t/v}$ represent the coverage area of $D_t$ beyond $D_v$, i.e., $D_{t/v} = D_t \setminus D_v$. In addition, let $C_{s/v}$ and $C_{t/v}$ represent the segments of $C_s$ and $C_t$ outside the coverage area of $D_v$, respectively.

From MFSP problem's point of view, we study the nature of the coverage area in $D_{s/v}$ that $s$ is essential for in the presence of various intersections that $C_{s/v}$ may have with a $C_{t/v}$ for any $t \in N(v)$. We now consider the intersection relations between $C_{s/v}$ and $C_{t/v}$. Note that $C_{s/v}$ and $C_{t/v}$ can have zero, one, or two intersections with each other.

**Lemma 3.1** *When $C_{s/v}$ has no intersection with a $C_{t/v}$ for any $t \in N(v)$, then the coverage area that $s$ is essential for, $D_{s/v}$, is one single RCCA.*

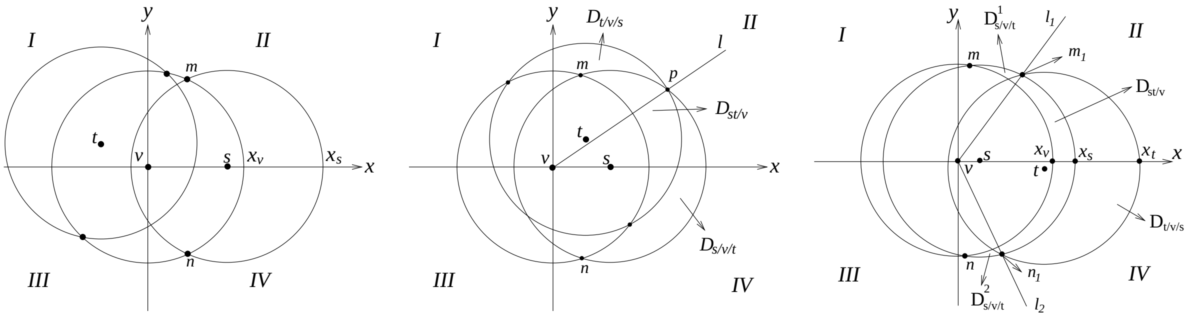
`Proof of Lemma 3.1`: Consider a $t \in N(v)$. When $C_{s/v}$ and $C_{t/v}$ have zero intersection, there are three possibilities for the relation between $D_{s/v}$ and $D_{s/t}$: (1) $D_{s/v} \subset D_{t/v}$; (2) $D_{t/v} \subset D_{s/v}$; and (3) $(D_{s/v} \cap D_{t/v}) = \varnothing$. Note that the first two cases are not possible. It is well-known that given two unit disks $D_s$ and $D_t$, the longest segment of the peripheral circle $C_s$ that $D_t$ can enclose is less than $\pi$. On the other hand, the segment $C_{s/v}$ is at least $\pi$. Therefore, it is not possible for $D_{t/v}$ to enclose the segment $C_{s/v}$ without $C_{t/v}$ intersecting $C_{s/v}$. This intersection indicates that $D_{s/v} \not\subset D_{t/v}$. A symmetric argument applies for the second case and $D_{t/v} \not\subset D_{s/v}$. Hence, the only relation between $D_{s/v}$ and $D_{t/v}$ is as $(D_{s/v} \cap D_{t/v}) = \varnothing$. If this is the case for any $t \in N(v)$, then it requires that the coverage are that $s$ is essential for is $D_{s/v}$ which is a single RCCA (see Figure 3.3-(a) as an example). $\square$

**Lemma 3.2** *When $C_{s/v}$ has one intersection with a $C_{t/v}$ for a $t \in N(v)$, then the coverage area that $s$ is essential for $D_{s/v} \setminus D_{t/v}$ remains to be one single RCCA.*

`Proof of Lemma 3.2`: From the previous section, the intersection of two disks $D_s$ and $D_t$ results in three coverage areas as (1) $D_s \setminus D_t$, (2) $D_t \setminus D_s$ and (3) $D_s \cap D_t$. In our case, we are interested in $D_s \setminus D_t$ outside of $D_v$ which we denote as $D_{s/v/t}$ as shown in Figure 3.3-(b). Note that $s$ is not essential any more for the area $D_s \cap D_t$ outside of $D_v$ (which we denote as $D_{st/v}$) as this region is also covered by $t$. Therefore, we only need to show that the area $D_{s/v/t}$ that $s$ is essential for after the intersection is an RCCA. Now, let $p$ be the intersection point of $C_{s/v}$ and $C_{t/v}$. Consider a line $l$ that originates at $v$ and crosses $p$ as in Figure 3.3-(b). The line $l$ divides $D_{t/v/s}$ and $D_{s/v/t}$ areas such that the radial order of all the points in

$D_{t/v/s}$ and $D_{s/v/t}$ are disjoint from each other. As a result, the coverage area $D_{s/v/t}$ that $s$ is essential for remains to be an RCCA. $\square$

Observe that the intersection of $C_{s/v}$ with $C_{t/v}$ reduces the essential coverage area of $s$ from $D_{s/v}$ to $D_{s/v/t}$. If $C_{s/v/t}$ has another single intersection with some other $t' \in N(v)$, from Lemma 3.2, this intersection will again reduce the essential coverage area of $s$ into $D_{s/v/t/t'}$ which will remain to be a single RCCA.



(a) (b) (c)

Figure 3.3. Some geometric relations of three intersecting unit disks.

**Lemma 3.3** *When $C_{s/v}$ has two intersections with a $C_{t/v}$ for a $t \in N(v)$, then the coverage area that $s$ is essential for, $D_{s/v} \setminus D_{t/v}$, is at most two disjoint RCCAs.*

`Proof of Lemma 3.3`: When $C_{s/v}$ and $C_{t/v}$ have two intersections, we have two symmetric cases depending on the distance between $s$ to $y$-axis and $t$ to $y$-axis. We present the case where $s$ is closer to $y$-axis and the other case (i.e., $t$ is closer to $y$-axis) follows from the symmetry. In this case, the intersection of $C_{s/v}$ and $C_{t/v}$ divides the coverage area of $D_{s/v}$ into three parts as (1) $D^1_{s/v/t} = (D_{s/v} \setminus D_{t/v}) \cap II$; (2) $D_{st/v} = D_{s/v} \cap D_{t/v}$; and (3) $D^2_{s/v/t} = (D_{s/v} \setminus D_{t/v}) \cap IV$ (see Figure 3.3-(c)). The intersection also divides the coverage area of $D_{t/v}$ into two parts as (1) $D_{ts/v} = D_{st/v} = (D_{t/v} \cap D_{s/v})$ and (2) $D_{t/v/s} = D_{t/v} \setminus D_{s/v}$.

Note that $s$ is not essential any more for the area $D_{ts/v} = D_{st/v} = (D_{t/v} \cap D_{s/v})$ as this is also covered by $t$. Therefore, we only need to show that the areas $D^1_{s/v/t}$ and $D^2_{s/v/t}$ are both RCCAs. Let $m_1$ and $n_1$ be the intersection points of $C_{s/v}$ and $C_{t/v}$ as shown in Figure 3.3-(c).

Consider the lines $l_1$ and $l_2$ that originate at $v$ and cross $m_1$ and $n_1$ respectively. Observe that the lines $l_1$ and $l_2$ divide $D^1_{s/v/t}$, $D_{t/v/s}$, and $D^2_{s/v/t}$ such that the radial order of all the points in these three areas are disjoint from each other. As a result, the coverage areas $D^1_{s/v/t}$ and $D^2_{s/v/t}$ that $s$ is essential for are both RCCAs. Similarly, the coverage area $D_{t/v/s}$ that $t$ is essential is also an RCCA. The symmetric case where $t$ is closer to $y$-axis results in the corresponding case where $t$ is essential for two RCCAs namely $D^1_{t/v/s}$ and $D^2_{t/v/s}$ and $s$ is essential for one RCCA namely $D_{s/v/t}$. □

**Lemma 3.4** *When $C_{s/v}$ has two intersections with a $C_{t/v}$ for a $t \in N(v)$, then $t \in D_s$.*

`Proof of Lemma 3.4`: When $s$ is closer to $y$-axis, $C_s$ and $C_v$ intersects at two points, $m$ and $n$, as shown in Figure 3.3-(c). Note that $Chord_{vs}$ is parallel to $y$-axis as $s$ sits on the $x$-axis. Note also that all the points of $D_v$ to the east of $Chord_{vs}$ is also in $D_s$. For $C_{t/v}$ to intersect $C_{s/v}$ twice, the two intersection points, $m_1$ and $n_1$, should be to the east of $m$ and $n$ respectively as otherwise $C_{s/v}$ and $C_{t/v}$ cannot intersect twice. Since $t \in N(v)$ and all points of $D_v$ to the east of $Chord_{vs}$ is also in $D_s$, $t \in D_s$ (more precisely, $t \in \{D_v \cap D_s\}$). □

**Lemma 3.5** *The arc $C^1_{s/v/t}$ (or $C^2_{s/v/t}$) that encloses the area $D^1_{s/v/t}$ (or $D^2_{s/v/t}$) that $s$ is essential for can have at most one intersection with any other arc $C_{t'/v}$ for a $t' \in N(v)$.*

`Proof of Lemma 3.5`: Without loss of generality, this lemma considers the case where $C_{s/v}$ intersects with $C_{t/v}$ twice in a way that $s$ is closer to $y$-axis. As discussed in the previous lemma, this intersection gives us two RCCAs, $D^1_{s/v/t}$ and $D^2_{s/v/t}$, that $s$ is essential for. Consider the characteristics of the arcs $arc(mm_1)$, $arc(m_1n_1)$, and $arc(n_1n)$ that form from the two intersections of $C_{s/v}$ and $C_{t/v}$ as shown in Figure 3.3-(c). We observe the following:

1. The length of the arc $arc(m_1x_sn_1)$ is in $[\frac{2\pi}{3}, \pi]$. From the two intersections of $C_{s/v}$ and $C_{t/v}$, we have the arc $arc(m_1x_tn_1)$ in $[\pi, \frac{4\pi}{3}]$. This requires that the angle $\widehat{m_1sn_1}$ is in $[\frac{2\pi}{3}, \pi]$. Consequently, the arc $arc(m_1x_sn_1)$ is in $[\frac{2\pi}{3}, \pi]$.

2. The sum $arc(mm_1)+arc(n_1n)$ is in $[0, \frac{2\pi}{3})$. This follows from the fact that $arc(mx_sn) = (arc(mm_1) + arc(m_1x_sn_1) + arc(n_1n))$ and $arc(m_1x_sn_1)$ is in $[\frac{2\pi}{3}, \pi]$.

From these observations, the length of $C^1_{s/v/t}$ and $C^2_{s/v/t}$ are both less than $\frac{2\pi}{3}$. Now, for $C_{t'/v}$ to intersect $C^1_{s/v/t}$ (or $C^2_{s/v/t}$) twice, Lemma 3.4 requires that $t' \in D_s$, i.e., $|t's| < 1$. From the previous section, if $|t's| < 1$, we have $arc(\tilde{m}\tilde{n}) \geq \frac{2\pi}{3}$ where $\tilde{m}$ and $\tilde{n}$ are the two intersection points of $C_{t'/v}$ and $C^1_{s/v/t}$ (or $C^2_{s/v/t}$) and $arc(\tilde{m}\tilde{n})$ is a segment of $C^1_{s/v/t} = arc(mm_1)$ enclosed between the two intersection points $\tilde{m}$ and $\tilde{n}$. From the above observations, $arc(mm_1) < \frac{2\pi}{3}$ but the two intersections of $C_{t'/v}$ and $C^1_{s/v/t}$ require that $arc(\tilde{m}\tilde{n}) \leq arc(mm_1) < \frac{2\pi}{3}$ and $arc(\tilde{m}\tilde{n}) \geq \frac{2\pi}{3}$ resulting in a contradiction. As a result, $C^1_{s/v/t}$ (or $C^2_{s/v/t}$) cannot intersect twice with a $C_{t'/v}$ for any $t' \in N(v)$. □

## 3.2 Geometrical Properties

### 3.2.1 Two-Set Property

**Theorem 3.1 (Two-Set Property)** *Given an instance of the MFSP problem (i.e., a node $v$ and its 1-hop and 2-hop neighbor sets $N(v)$ and $N_2(v)$), each node $s \in N(v)$ is* essential *to cover at most two MCIs. As a result, the connectivity scenario of $b_1$ presented in the connectivity matrix in Figure 3.4-(a) is not possible.*

`Proof of Theorem 3.1`: The proof is based on the coverage areas, RCCAs, of the nodes in $N(v)$. Consider $s \in N(v)$. If $N(s) \cap N_2(v) = \varnothing$, then $s$ does not cover any nodes in $N_2(v)$, i.e., $s$ is essential for zero MCIs. Assume $N(s) \cap N_2(v) \neq \varnothing$. From Lemma 3.1, if $C_{s/v}$ does not intersect with a $C_{t/v}$ for any $t \in N(v)$, then $s$ is essential to cover one single RCCA, $D_{s/v}$. Based on this, the nodes in $N_2(v) \cap D_{s/v}$ gives the corresponding single MCI that $s$ is essential to cover in $N_2(v)$.

From Lemma 3.2, if $C_{s/v}$ intersects once with a $C_{t/v}$ for any $t \in N(v)$, then $s$ is essential to cover one RCCA, $D_{s/v/t}$, and the area $D_{st/v}$ can be covered by both $s$ and $t$. If $N_2(v) \cap D_{t/v/s} = \varnothing$, then $s$ is essential to cover one single MCI that includes the nodes in

$N_2(v) \cap D_{s/v}$. If $N_2(v) \cap D_{t/v/s} \neq \varnothing$, then we consider the area $D_{st/v}$. If $N_2(v) \cap D_{st/v} = \varnothing$, then $s$ is essential to cover one single MCI that includes the nodes in $N_2(v) \cap D_{s/v/t}$ as $D_{s/v/t}$ and $D_{t/v/s}$ are radially disjoint. When $N_2(v) \cap D_{st/v} \neq \varnothing$, $s$ is still essential for one single MCI that includes all the nodes in $N_2(v) \cap D_{s/v/t}$ and possibly some other nodes in $N_2(v) \cap D_{st/v}$. Note that $s$ is not essential for the nodes in $N_2(v) \cap D_{st/v}$ but based on our definition of an MCI, some of these nodes may be part of the MCI that $s$ is essential for.

From Lemma 3.3, if $C_{s/v}$ intersects twice with a $C_{t/v}$ for any $t \in N(v)$, then $s$ is essential to cover at most two RCCAs, $D^1_{s/v/t}$ and $D^2_{s/v/t}$. If $N_2(v) \cap D^1_{s/v/t} \neq \varnothing$ and $N_2(v) \cap D^2_{s/v/t} \neq \varnothing$, then $s$ is essential to cover two MCIs that includes all the nodes in $N_2(v) \cap D^1_{s/v/t}$ and $N_2(v) \cap D^2_{s/v/t}$, respectively. Observe that the only way to divide an RCCA that $s$ is essential for into two disjoint RCCAs is to have $C_{s/v}$ intersect twice with another $C_{t/v}$ for any $t \in N(v)$. But once such an intersection occurs, by Lemma 3.5, neither of the the resulting RCCAs that $s$ is essential for, namely $D^1_{s/v/t}$ and $D^2_{s/v/t}$, can be divided into two other RCCAs by any other $C_{t'/v}$ for any $t' \in N(v)$. And, from Lemma 3.2, a single intersection between $D_{t'/v}$ and $D^1_{s/v/t}$ (or $D^2_{s/v/t}$) results in one single (and smaller) RCCA $D^1_{s/v/t/t'}$. As a result, $s$ can be essential for at most two MCIs. $\square$

*Connectivity Matrix*

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ |
|---|---|---|---|---|---|
| $b_1$ | 1 | 0 | 1 | 0 | 1 |
| $b_2$ | 0 | 0 | 0 | 1 | 0 |
| $b_3$ | 0 | 1 | 0 | 0 | 0 |

(a)

*Connectivity Matrix*

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
|---|---|---|---|---|
| $b_1$ | 1 | 0 | 1 | 0 |
| $b_2$ | 0 | 1 | 0 | 1 |

(b)

Figure 3.4. Unfeasible connectivity scenarios.

### 3.2.2 Non-Interleaving Property

**Definition 3.8 (Interleaving Coverage)** *Consider two nodes* $\{s, t\} \in N(v)$ *in an instance of MFSP problem at* $v$. *Assume* $s$ *covers* $\{a, c\} \in N_2(v)$ *but does not cover* $\{b, d\} \in N_2(v)$. *Similarly, assume* $t$ *covers* $\{b, d\} \in N_2(v)$ *but does not cover* $\{a, c\} \in N_2(v)$. *Finally,* *assume that the radial order between the nodes in $N_2(v)$ is as ($a > b > c > d$). The coverage of this form between the nodes $s$ and $t$ is called an interleaving coverage.*

**Theorem 3.2 (Non-Interleaving Property)** *In an instance of the MFSP problem (i.e., a node $v$ and its 1-hop and 2-hop neighbor sets $N(v)$ and $N_2(v)$), no two nodes $\{s, t\} \in N(v)$ can have interleaving coverage, i.e., the connectivity matrix in Figure 3.4-(b) is not feasible.*

`Proof of Theorem 3.2`: Note that interleaving is considered between two nodes $\{s, t\} \in N(v)$. If $C_{s/v}$ and $C_{t/v}$ intersect zero times, $s$ and $t$ both have two disjoint coverage areas. If $C_{s/v}$ and $C_{t/v}$ intersect once, based on Lemma 3.2, the coverage area that $s$ (and $t$) is essential for is a single RCCA. When this RCCA includes some node $a_i \in N_2(v)$, then $s$ (and $t$) has one single MCI (including such node $a_i$) that it is essential for. Finally, when they intersect twice, based on Lemma 3.3, one of the nodes, say $s$, can be essential to cover two MCIs and the other node $t$ can be essential to cover one MCI. Note that, as shown in Figure 3.3-c, the two intersections radially separate the coverage areas $D^1_{s/v/t}$, $D_{t/v/s}$, and $D^2_{s/v/t}$ and no points $q \in D_{t/v}$ can lie beyond the two areas $D^1_{s/v/t}$ and $D^2_{s/v/t}$. $\square$

## 3.3 Solution

In this section, we present a polynomial time algorithm to solve the MFSP problem. The main idea in our approach is to break a given problem into subproblems and use the solutions of the subproblems in building a solution to the given instance of the problem. Consider an instance of an MFSP problem at a node $v$. Let $n$ be the number of 2-hop neighbors of $v$, i.e., $n = |N_2(v)|$. Assume that nodes $a_i \in N_2(v)$ are sorted based on their radial order w.r.t. $v$. The algorithm presented below executes in $n$ rounds. At a round $j$, we divide $N_2(v)$ into $n$ different consecutive regions each starting from a different $a_i$ ($1 \leq i \leq n$) and covering $j$ consecutive nodes in $N_2(v)$. We use the tuple $(a_i, j)$ to represent the consecutive interval of nodes $(a_i, a_{i+1}, \ldots, a_{i+j-1}) \in N_2(v)$ considered by each of these problems[1]. The problems are then solved by using the solutions of the smaller size problems solved in previous rounds.

Let $L_{min}(a_i, j)$ be a list of minimum number of first hop neighbors of $v$ that are essential to cover the interval $(a_i, j)$. At round $n$, the algorithm returns $n$ solutions as $L_{min}(a_i, n)$ where $i = [1, n]$. The optimal solution is given by a minimum size $L_{min}(a_i, n)$ where $i = [1, n]$.

### 3.3.1 Construction

In our algorithm, we find the optimal solution in three steps. The overall algorithm is provided in Figure 3.5. In the figure, we use $N_{min}(a_i, n) = |L_{min}(a_i, n)|$ for ease-of-presentation. The algorithm works in $n$ rounds. At each round, we consider $n$ subproblems for $n$ different intervals of length $j$, i.e., $\{(a_1, j), (a_2, j), \ldots, (a_n, j)\}$. Below are the steps of the algorithm in finding a solution for an interval $(a_i, j)$ where $\{i, j\} \in [1, n]$:

1. **Step 1:** The best possible solution for $(a_i, j)$ is that the entire interval is covered by a single node $e \in N(v)$. This can be checked by searching the column $f = a_i$ of the coverage matrix. If there exists an MCI $C_{ef} = (a_p, \bar{p})$ that completely includes the interval $(a_i, j)$, then the corresponding one hop neighbor $e$ can be assigned to cover the interval $(a_i, j)$ in the solution. Since this is an assignment with minimum size, i.e., $N_{min}(a_i, j) = 1$, there is no need to check for the other cases below.
2. **Step 2:** In this step, we split the interval $(a_i, j)$ to two consecutive sub-intervals as $(a_i, k)$ and $(a_{i+k},\, j - k)$. We can combine the optimal solutions of these intervals and this will be a solution to $(a_i, j)$. We consider each possible case for splitting the interval $(a_i, j)$ into two intervals. There are $j - 1$ possible cases. Since we are interested in minimum cardinality solution, we take the minimum one in cardinality.
3. **Step 3:** In this step, we pick a special 1-hop node $s$ which covers $a_i$ and $a_{i+j-1}$ (end nodes of $(a_i, j)$) and find the MCIs $(a_p, \bar{p})$ and $(a_q, \bar{q})$ that $s$ covers such that $a_i \in (a_p, \bar{p})$ and $a_{i+j-1} \in (a_q, \bar{q})$. A solution in this case can be given by $\{s\} \cup L_{min}(a_r, \bar{r})$ where

---

[1] Note that the arithmetic in computing the subscripts of the nodes preserves radial circularity.

$(a_r, \bar{r}) = \{a_{p+\bar{p}}, \ldots, a_{q-1}\}$ . We find such solutions for all possible $s$ and save it as the optimal solution if it is better than the current solution.

Starting from $j = 1$, the algorithm applies the above procedure for all intervals of lengths up to $j = n$. For an interval $(a_i, j)$, it considers possible solution scenarios by applying the above procedure. Among those solutions, it chooses the one that gives $L_{min}(a_i, j)$. At the end, it chooses a solution with a minimum size, $L_{min}(a_i, n)$, as an optimal solution to the given instance of the problem. The total running time of the algorithm is $O(n^3 + n^2m)$ where $O(n^3)$ comes from the execution of Step 2 and $O(n^2m)$ comes from the execution of Steps 1 and 3. Similarly, the space complexity of the algorithm becomes $\Theta(n^2k)$ where $k$ represents an upper bound for the number of forwarding nodes in an optimal solution ($k < min(n, m)$).

### 3.3.2 Proof of Correctness

**Theorem 3.3** *The algorithm presented in Figure 3.5 produces an optimal solution to the MSFP problem at a node $v$.*

`Proof of Theorem 3.3`: The proof is by induction. The base case is for any $(a_i, j)$ where $j = 1$. The optimal solution in this case is found during the initialization phase of the algorithm (at lines 6-8 in Figure 3.5). For the hypothesis case, we assume that we have the optimal solution for any $(a_i, j-1)$. Then, in the following, we prove that the algorithm finds an optimal solution for $(a_i, j)$.

Let $S = L_{min}(a_i, j)$ be an optimal solution for $(a_i, j)$. Let $b_k \in S$ be a node that covers the largest MCI starting from $a_i$, i.e., $b_k$ covers $(a_i, \bar{x})$ and for any $b_l \neq b_k \in S$ covering an MCI $(a_i, \tilde{x})$, $\tilde{x} \leq \bar{x}$. Note that we know only the existence of such a $b_k$ and do not need to know its identity. We analyze the relations between the coverage characteristics of $b_k$ and all other nodes in $S$ in all possible coverage cases. Note that $b_k$ is essential for at least one and at most two MCIs in $(a_i, j)$. It is essential for at least one MCI because it is in $S$. It is essential for at most two MCIs by the 2-set property. Figure 3.6 presents possible

```
INPUT: An instance of MFSP problem at a node $v$:
  1. $N(v) := \{b_1, \ldots, b_m\}$
  2. $N_2(v) := \{a_1, \ldots, a_n\}$
  3. An $mtimen$ coverage matrix $C$ as defined in Definition 3.5
OUTPUT: $MFS(v) = N_{min}(a_i, n)$ for any $i$ where $1 \leq i \leq n$

/* initialization */
FOR $i := 1$ to $n$
    $N_{min}(a_i, 1) := 1$, $L_{min}(a_i, 1) := (b)$ where $b \in N(v)$ and $a_i \in N(b)$

/* main body of the algorithm */
FOR $j := 2$ to $n$ /* for each round */
    FOR $i := 1$ to $n$ /* for each interval $(a_i, j)$*/
        $N_{min}(a_i, j) := \infty$, $L_{min}(a_i, j) := \emptyset$

        /* Step 1: Find $b_e \in N(v)$ that covers the largest interval including $(a_i, j)$ */
        $blen := 0$
          FOR $e := 1$ to $m$
           IF $C_{ei}$ includes $(a_i, j)$ as a subinterval THEN
               $N_{min}(a_i, j) := 1$
               IF $(|C_{ei}| > blen)$ THEN
                   $blen := |C_{ei}|$, $L_{min}(a_i, j) := (b_e)$

        IF $(N_{min}(a_i, j) \neq 1)$ THEN
           /* Step 2: Check for alternative solutions */
           FOR $k := 1$ to $j - 1$ /* for all possible splits of the interval $(a_i, j)$ */
               IF $(N_{min}(a_i, k) + N_{min}(a_{i+k}, j - k) < N_{min}(a_i, j))$ THEN
                   $N_{min}(a_i, j) := N_{min}(a_i, k) + N_{min}(a_{i+k}, j - k)$
                   $L_{min}(a_i, j) := (L_{min}(a_i, k)) + (L_{min}(a_{i+k}, j - k))$ /* + is list append */

           /* Step 3: Check if a $b_e \in N(v)$ covers both ends of interval $(a_i, j)$ */
           FOR $e := 1$ to $m$
               Consider coverage matrix entries $C_{ei} = (a_p, \bar{p})$ and $C_{e(i+j-1)} = (a_q, \bar{q})$
               IF $(C_{ei} \neq \varnothing)$ AND $(C_{e(i+j-1)} \neq \varnothing)$ THEN
                   Let $(a_r, \bar{r}) = (a_{p+\bar{p}}, a_{p+\bar{p}+1}, \ldots, a_{q-1})$ be the subinterval that $b_e$ does not cover
                   IF $(N_{min}(a_r, \bar{r}) + 1 < N_{min}(a_i, j))$ THEN
                       $N_{min}(a_i, j) := N_{min}(a_r, \bar{r}) + 1$
                       $L_{min}(a_i, j) := (L_{min}(a_r, \bar{r})) + (b_e)$ /* + is list append */

/* finalization */
$MFS(v) := m$, $L_{MFS(v)} := N(v)$
FOR $i := 1$ to $n$
    IF $N_{min}(a_i, n) < MFS(v)$ THEN
        $MFS(v) := N_{min}(a_i, n)$, $L_{MFS(v)} := L_{min}(a_i, n)$

RETURN $MFS(v)$ and $L_{MFS(v)}$
```

Figure 3.5. Outline of the algorithm.

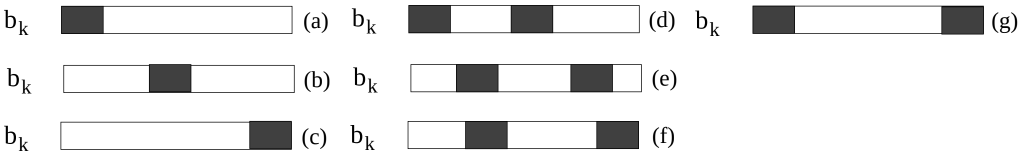


Figure 3.6. Possible MCIs $b_k$ is essential for in $S$.

configurations of MCIs where $b_k$ is essential for in $S$. In the following, we first show that the cases in Figure 3.6-(e),(f) are not possible and then present how our algorithm handles the other cases.

**Claim 1** *If $b_k$ is essential for two MCIs, one of them should be the MCI $(a_i, \bar{x})$.*

`Proof of Claim 1:` Assume that this is not the case and $b_k$ is essential for two MCIs $(a_z, \bar{z})$ and $(a_d, \bar{d})$ as in Figure 3.7-(a) or -(b). Note that these cases correspond to the cases in Figure 3.6-(e),(f). Let $b_l \in S$ be a node covering $a_y$ and $b_m \in S$ be a node covering $a_c$ in the figure. Note that since $b_k$ does not cover either $a_y$ or $a_c$, $b_l \neq b_m$ due to non-interleaving property. Note also that since $b_k$ covers the largest MCI $(a_i, \bar{x})$, $b_l$ cannot cover this MCI completely. Finally, since $b_k$ is assumed to be essential for $(a_z, \bar{z})$ and $(a_d, \bar{d})$, neither $b_l$ nor $b_m$ can cover these MCIs completely. This then causes a violation of 2-set property by $b_k$. □

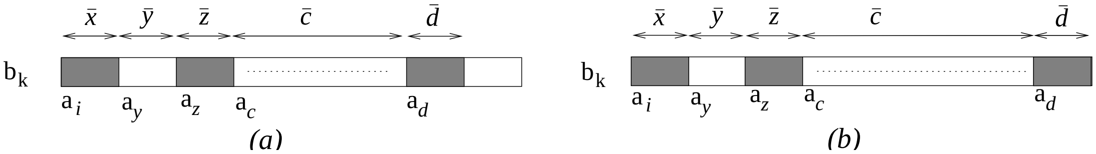


Figure 3.7. Coverage characteristics of $b_k$ in $S$.

**Case 1:** Assume $b_k$ is essential for only one MCI on one of the ends of the interval, e.g., $(a_i, \bar{x})$ as in Figure 3.7-(a) (or $(a_d, \bar{d})$ as in Figure 3.7-(b)). We explain the behavior of the algorithm using the first case and the same argument applies for the second case. Since $b_k$ is not essential for any other nodes in $(a_y, j - \bar{x})$, $S \setminus \{b_k\}$ covers $(a_y, j - \bar{x})$. Consider an optimal solution $L_{min}(a_y, j - \bar{x})$ for the interval $(a_y, j - \bar{x})$. Note that $|L_{min}(a_y, j - \bar{x})|$ cannot be larger than $|S \setminus \{b_k\}|$ as the former is an optimal solution and the latter is a solution covering $(a_y, j-\bar{x})$, i.e., $|L_{min}(a_y, j-\bar{x})| \leq |S \setminus \{b_k\}| = |S|-1$. Similarly, an optimal solution $L_{min}(a_i, x)$ for the interval $(a_i, \bar{x})$ can not be larger than $|\{b_k\}|$ as the former is an optimal solution and the letter is a solution covering $(a_i, \bar{x})$. This makes $L_{min}(a_i, \bar{x}) \cup L_{min}(a_y, j-\bar{x})$ an optimal solution for $(a_i, j)$ as we assumed that $S$ is an optimal solution for $(a_i, j)$.

This case covers the essential coverage scenarios of $b_k$ in $S$ as presented in Figure 3.6-(a),-(c). If the optimal solution $S$ is of this nature, since we know the optimal solutions for all intervals of length up to $j-1$, the presented algorithm finds this solution in Step 2 by looking at each split points for the interval $(a_i, j)$.

**Case 2:** Assume that $b_k$ is essential for two MCIs $(a_i, \bar{x})$ and $(a_d, \bar{d})$ at both ends of $(a_i, j)$ as shown in Figure 3.7-(b). In this case, $S \setminus \{b_k\}$ covers $(a_y, j-(\bar{x}+\bar{d}))$ since $b_k$ is not essential for that part. Consider an optimal solution $L_{min}(a_y, j-(\bar{x}+\bar{d}))$ for the interval $(a_y, j-(\bar{x}+\bar{d}))$. Note that $|L_{min}(a_y, j-(\bar{x}+\bar{d}))|$ cannot be larger than $|S \setminus \{b_k\}|$ as the former is an optimal solution and the latter is a solution $(a_y, j-(\bar{x}+\bar{d}))$, i.e., $|L_{min}(a_y, j-(\bar{x}+\bar{d}))| \leq |S \setminus \{b_k\}| = |S|-1$. This then requires that $\{b_k\} \cup L_{min}(a_y, j-(\bar{x}+\bar{d}))$ is an optimal solution for $(a_i, j)$ as we started with an assumption that $S$ is an optimal solution for $(a_i, j)$, i.e., $|L_{min}(a_y, j-(\bar{x}+\bar{d}))| + |\{b_k\}| \leq |S|$.

This case covers the essential coverage scenarios of $b_k$ in $S$ as presented in Figure 3.6-(g). If the optimal solution $S$ is of this nature, since we know the optimal solutions for all intervals of length up to $j-1$, the presented algorithm finds this solution in Step 3.

**Case 3:** Assume that $b_k$ is essential for two MCIs $(a_i, \bar{x})$ and $(a_z, \bar{z})$ as shown in Figure 3.7-(a) or -(b). In this scenario, we divide the nodes in $S$ into two subgroups, $S_1$ and $S_2$ in a way that $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \varnothing$. Our goal is to create $S_1$ and $S_2$ such that $S_1$ covers $(a_i, \bar{x}+\bar{y}+\bar{z})$ and $S_2$ covers $(a_c, j-(\bar{x}+\bar{y}+\bar{z}))$. When we assign a node $b_x$ to $S_1$ (or $S_2$), if we can guarantee that there exist another node $b_y$ in $S_2$ (or $S_1$) which covers all the nodes that $b_x$ covers in $(a_c, j-(\bar{x}+\bar{y}+\bar{z}))$ (or $(a_i, \bar{x}+\bar{y}+\bar{z})$), then this partition will give two sets $S_1$ and $S_2$ with the desired properties.

Assume that $b_k$ is the only node covering $(a_i, \bar{x})$ as in Figure 3.7-(a) or -(b). Then, there is a node, say $b_m$, in $S$ which covers $a_c$.

**Claim 2** *$b_k$ should cover all nodes that $b_m$ covers in $(a_i, \bar{x} + \bar{y} + \bar{z})$.*

`Proof of Claim 2:` $b_k$ covers a node in $(a_i, \bar{x})$ and a node in $(a_z, \bar{z})$ which are not covered by $b_m$. $b_m$ covers $a_c$ which is not covered by $b_k$. If $b_k$ does not cover a node that $b_m$ covers in $(a_y, \bar{y})$, this leads to an interleaving coverage between $b_k$ and $b_m$ contradicting Theorem 3.2. $\square$

**Claim 3** *$b_m$ should cover all the nodes that $b_k$ covers in $(a_c, j - (\bar{x} + \bar{y} + \bar{z}))$.*

`Proof of Claim 3:` Recall that $b_k$ does not cover $a_y$. Let $b_l$ be a node covering $a_y$. Note that $b_l \neq b_m$ as otherwise $b_k$ and $b_l$ have an interleaving coverage. Now, $b_k$ covers nodes in $(a_i, \bar{x})$ and in $(a_z, \bar{z})$ which $b_l$ does not cover. Note also that $b_m$ covers $a_c$ which is not covered by $b_k$. If $b_m$ does not cover a node $a_d \in (a_c, j - (\bar{x} + \bar{y} + \bar{z}))$ $(a_d \neq a_c)$ that $b_k$ covers, $b_k$ violates 2-set property. As a result $b_m$ has to cover all such $a_d \in (a_c, j - (\bar{x} + \bar{y} + \bar{z}))$. $\square$

Based on Claims 2 and 3, we can put $b_k$ into $S_1$ and $b_m$ into $S_2$. For any other node $b_o$:

1. If $b_o$ covers a node in $(a_c, j - (\bar{x} + \bar{y} + \bar{z}))$ which is not covered by $b_k$, then $b_o \in S_2$ and $b_k$ covers all nodes that $b_o$ covers in $(a_i, \bar{x} + \bar{y} + \bar{z})$ due to Theorem 3.2.
2. If $b_o$ covers a node in $(a_y, \bar{y})$ which is not covered by $b_k$, then $b_o \in S_1$ and $b_k$ covers all nodes that $b_o$ covers in $(a_c, j - (\bar{x} + \bar{y} + \bar{z}))$ due to Theorem 3.2. From Claim 3, $b_m$ covers all nodes that $b_k$ covers in $(a_c, j - (\bar{x} + \bar{y} + \bar{z}))$. Therefore, $b_m$ covers all nodes that $b_o$ covers in $(a_c, j - (\bar{x} + \bar{y} + \bar{z}))$.

Note that $b_o$ should be in one of the above two cases as otherwise $b_l \notin S$. Based on the above construction, $S_1$ covers $(a_i, \bar{x} + \bar{y} + \bar{z})$ and $S_2$ covers $(a_c, j - (\bar{x} + \bar{y} + \bar{z}))$. Consider optimal solutions $L_{min}(a_i, \bar{x} + \bar{y} + \bar{z})$ and $L_{min}(a_c, j - (\bar{x} + \bar{y} + \bar{z}))$. Due to optimality, we have $|L_{min}(a_i, \bar{x}+\bar{y}+\bar{z})| \leq |S_1|$ and $|L_{min}(a_c, j-(\bar{x}+\bar{y}+\bar{z}))| \leq |S_2|$. Since $S = S_1 \cup S_2$ is an optimal solution, we have $|L_{min}(a_i, \bar{x}+\bar{y}+\bar{z})| + |L_{min}(a_c, j-(\bar{x}+\bar{y}+\bar{z}))| \leq |S_1| + |S_2| = |S|$.

This case covers the essential coverage scenarios of $b_k$ in $S$ as presented in Figure 3.6-(d). If the optimal solution $S$ is of this nature, since we know the optimal solutions for all intervals of length up to $j-1$, the presented algorithm finds this solution in Step 2.

**Case 4:** We now examine the case where $b_k$ is essential for one interval $(a_y, \bar{y})$ which corresponds to the case in Figure 3.6-(b). Similar to the above discussion, we again divide $S$ into $S_1$ and $S_2$ and try to put nodes $b_x$ into one of these two sets such that $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \varnothing$. Let $b_m \in S$ a node covering $a_c$ and $b_l \in S$ a node covering $a_y$. As presented above $b_l \neq b_m$ to avoid an interleaving coverage between $b_k$ and $b_l$.

**Claim 4** *We claim that $b_k$ or $b_l$ should cover all nodes that $b_m$ covers in $(a_i, \bar{x}+\bar{y}+\bar{z})$.* □

`Proof of Claim 4:` Note that if $b_m$ does not cover $(a_i, \bar{x})$, then the proof is the same as the proof of Claim 2 above. If $b_m$ covers $(a_i, \bar{x})$, it can cover some node in $(a_y, \bar{y})$ and some node in $(a_c, \bar{c})$ which are not covered by $b_k$. If this is the case, $b_l$ should cover all the nodes $b_m$ covers in $(a_y, \bar{y})$ as otherwise $b_m$ violates 2-set property. That is, $b_m$ covers $(a_i, \bar{x})$ and some nodes in $a_{\tilde{y}} \neq a_y \in (a_y, \bar{y})$ which $b_l$ does not cover; and $b_l$ covers $a_y$ which $b_m$ does not cover. Based on this, $b_l$ separates the coverage of $b_m$ into two sets. In addition, $b_m$ covers $a_{\tilde{y}} \neq a_y \in (a_y, \bar{y})$ and $a_c$ which are not covered by $b_k$ and $b_k$ covers $(a_z, \bar{z})$ which $b_m$ does not cover. Based on this, $b_k$ increases the coverage of $b_m$ from two sets to three sets which contradicts Theorem 3.1. □

**Claim 5** *$b_m$ should cover all nodes $b_k$ covers in $(a_c, j-(\bar{x}+\bar{y}+\bar{z}))$.*

`Proof of Claim 5:` $b_k$ covers a node in $(a_i, \bar{x})$ and a node in $(a_z, \bar{z})$ which are not covered by $b_l$. $b_l$ covers $a_y$ which is not covered by $b_k$. $b_m$ covers $a_c$ which is not covered by $b_k$. If $b_m$ does not cover a node that $b_k$ covers in $(a_c, j-(\bar{x}+\bar{y}+\bar{z}))$, this will be a violation of 2-set property by $b_k$. □

**Claim 6** *$b_k$ should cover all nodes that $b_l$ covers in $(a_c, j-(\bar{x}+\bar{y}+\bar{z}))$.*

`Proof of Claim 6:` $b_k$ covers nodes in $(a_i, \bar{x})$ and in $(a_z, \bar{z})$ which are not covered by $b_l$. $b_l$ covers $a_y$ that $b_k$ does not cover. If $b_k$ does not cover a node $b_l$ covers in $(a_c, j-(\bar{x}+\bar{y}+\bar{z}))$, this will lead to an interleaving coverage between $b_k$ and $b_l$ contradicting Theorem 3.2. $\square$

Combining Claims 5 and 6, $b_m$ covers all nodes that $b_l$ covers in $(a_c, j-(\bar{x}+\bar{y}+\bar{z}))$. Based on these results, we can put $b_k$ and $b_l$ into $S_1$ and $b_m$ into $S_2$. For any other node $b_o$:

1. If $b_o$ does not cover a node in $(a_y, \bar{y})$ which is not covered by $b_k$, the $b_o \in S_2$.
2. If $b_o$ does not cover a node in $(a_c, j-(\bar{x}+\bar{y}+\bar{z}))$ which is not covered by $b_k$, then $b_o \in S_1$. $b_m$ covers all nodes $b_o$ covers in $(a_c, j-(\bar{x}+\bar{y}+\bar{z}))$.
3. If $b_o$ covers nodes from both $(a_y, \bar{y})$ and $(a_c, j-(\bar{x}+\bar{y}+\bar{z}))$ which are not covered by $b_k$, then $b_o \in S_2$. In this case, $b_l$ covers all nodes $b_o$ covers in $(a_y, \bar{y})$. Now, to have a non-interleaving coverage between $b_o$ and $b_k$, $b_o$ should cover $(a_i, \bar{x})$ as $b_k$ covers a node in $(a_z, \bar{z})$ that $b_o$ does not cover. Next, if $b_l$ does not cover all nodes $b_o$ covers in $(a_y, \bar{y})$, $b_o$ will violate 2-set property. This case is similar to the case of $b_m$ in the proof of Claim 4 and is therefore omitted.

Note that $b_o$ should be in one of the above two cases as otherwise $b_o \notin S$. Similar to the discussion above, for optimal solutions $L_{min}(a_i, \bar{x}+\bar{y}+\bar{z})$ and $L_{min}(a_c, j-(\bar{x}+\bar{y}+\bar{z}))$, we have $|L_{min}(a_i, \bar{x}+\bar{y}+\bar{z})| \leq |S_1|$ and $|L_{min}(a_c, j-(\bar{x}+\bar{y}+\bar{z}))| \leq |S_2|$. Since $S = S_1 \cup S_2$ is an optimal solution, we have $|L_{min}(a_i, \bar{x}+\bar{y}+\bar{z})| + |L_{min}(a_c, j-(\bar{x}+\bar{y}+\bar{z}))| \leq |S_1| + |S_2| = |S|$.

This case covers the essential coverage scenarios of $b_k$ in $S$ as presented in Figure 3.6-(b). If the optimal solution $S$ is of this nature, since we know the optimal solutions for all intervals of length up to $j-1$, the presented algorithm finds this solution in Step 2. $\square$

As a result, depending on the nature of the optimal solution for $(a_i, j)$, one of the above mentioned four cases correspond to the optimal solution $L_{min}(a_i, j)$. This concludes the proof of correctness of the algorithm.

### 3.3.3 Discussion

In this section, we justify the need for the finalization step of the algorithm presented in Figure 3.5. During each round of the algorithm (starting from line 10 in Figure 3.5), we consider an interval of length $j$ including nodes $a_{x+1}, \ldots, a_{x+j} \in N_2(v)$. However, during the iteration of the algorithm, we ignore the circularity of this interval, i.e., we ignore that $a_{x+j}$ and $a_{x+1}$ are radially consecutive. Since we ignore circularity, depending on the beginning of an interval, two MCIs that are covered by the same node $s \in S$ may seem like three MCIs. As an example, in Figure 3.8, $s$ is essential for two MCIs, $(a_1, 2)$ and $(a_4, 1)$. However, when we consider $(a_2, 5)$, it seems that $s$ is essential for three MCIs as $(a_2, 1)$, $(a_4, 1)$, and $(a_1, 1)$. Since the algorithm does not consider the circularity of the interval $(a_2, 5)$, it cannot see that $a1$ and $a_2$ are consecutive. This then results in a suboptimal solution for $(a_2, 5)$ as follows:

- From **Step 2** of the algorithm, we have four potential split cases as:
    1. For $k = 1$, we have $(a_2, 1)$ and $(a_3, 4)$ resulting a solution $N_{min}(a_2, 5) = N_{min}(a_2, 1) + N_{min}(a_3, 4) = 1 + 3 = 4$.
    2. For $k = 2$, we have $(a_2, 2)$ and $(a_4, 3)$ with a solution $N_{min}(a_2, 5) = 2 + 2 = 4$.
    3. For $k = 3$, we have $(a_2, 3)$ and $(a_5, 2)$ with a solution $N_{min}(a_2, 5) = 2 + 2 = 4$.
    4. For $k = 4$, we have $(a_2, 4)$ and $(a_1, 1)$ with a solution $N_{min}(a_2, 5) = 3 + 1 = 4$.
- From **Step 3** of the algorithm, we have $N_{min}(a_2, 5) = N_{min}(a_3, 3) + 1 = 3 + 1 = 4$. Note that in this case, $s$ is assigned to cover $a_2$ and $a_1$ as these nodes make the two ends of the interval as in Figure 3.4-d with $t = 0$.

The above computation results in $N_{min}(a_2, 5) = 4$ which is suboptimal for the overall problem instance. Finally, since the algorithm checks for all the intervals $(a_i, 5)$ for $i \in [1, 5]$, it finds an optimal solution for this interval as $N_{min}(a_1, 5) = N_{min}(a_1, 4) + N_{min}(a_5, 1) = 2 + 1 = 3$ (where $N_{min}(a_1, 4) = 2$ comes from **Step 3** of the algorithm). And the finalization part at the end of the algorithm in Figure 3.5 is then used to find the optimal result by finding the minimum one among $N_{min}(a_i, n)$ for $i \in [1, n]$.

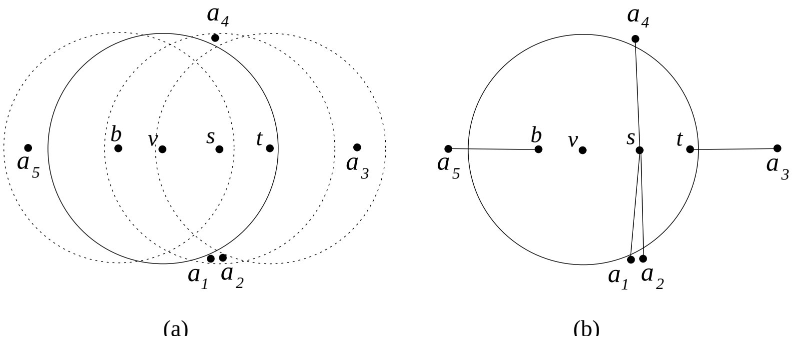


Figure 3.8. An example on finalization.

## 3.4 Evaluations

Recall that the MFSP problem is an instance of the well-known Set Cover problem. Several heuristics have been proposed for the Set Cover problem in the literature [13, 35]. In this section, we compare two of these heuristics with our algorithm. Our goal is to use our algorithm as a yard stick to evaluate the performance of these heuristics in solving the MFSP problem. The first heuristic is by Chvatal [13]. This heuristic uses an iterative approach in which a one hop node $s \in N(v)$ that covers the largest number of nodes in $N_2(v)$ is included into the solution set $MFS(v)$ at each phase and the two sets $N(v)$ and $N_2(v)$ are updated as $N(v) = N(v) - \{s\}$ and $N_2(v) = N_2(v) - \{N(s) \cap N_2(v)\}$ before starting the next phase of the iteration. The procedure terminates when $N_2(v) = \emptyset$ and the resulting solution set $MFS(v)$ gives a forwarding set of $v$.

Qayyum et al. [35] extends Chvatal's heuristic with a pre-processing step where they populate $MFS(v)$ by those one-hop neighbors that are essential for some nodes in $N_2(v)$ in a given instance of the problem. Such relay nodes are referred to as *mandatory nodes* as they have to be in any $MFS(v)$. They then remove mandatory nodes from $N(v)$ and remove all two-hop neighbors of $v$ that they cover from $N_2(v)$. If the resulting $N_2(v)$ is not an empty set, they use the Chvatal's heuristic to identify additional relay nodes to cover them.

In our evaluations, we build connected network topologies by randomly placing a number of nodes on a 7x7 grid. We then use our algorithm and the two heuristics to

Table 3.1. Average size of the relay node sets.

| Topo. size | $\lvert N_2(v) \rvert$ | #Mand.nodes | Opt.soln. | Chvatal h. | Qayyum h. |
|---|---|---|---|---|---|
| 100 | 6.66 | 1.7 | 2.3 | 2.32 | 2.3 |
| 250 | 26.61 | 3.24 | 4.35 | 4.46 | 4.4 |
| 1000 | 120.65 | 5.1 | 7.26 | 7.6 | 7.3 |

compute the $MFS(v)$ for each $v$ in the topology. We use topologies of sizes 100, 250, and 1000 nodes to consider both sparse and dense networks. For each experiment, we report the average values of $MFS(v)$ for the considered topology. The overall results are presented in Table 3.1.

The second column in the table gives the average number of two-hop neighbors for each node in the corresponding topology. The following columns give the averages of the number of mandatory nodes, the average of the optimal solution set, the average of the solution set given by Chvatal's and by Qayyum et al.'s heuristic respectively. According to the results, the number of mandatory nodes dominates the size of the optimal solution set. In addition, as the number of two-hop neighbors increase with the increasing topology size, the number of nodes in the optimal solution stays fairly small for the selected random topologies. Finally, the results demonstrate that the existing heuristics perform reasonably well with Qayyum et al.'s heuristic performing relatively better than the Chvatal's heuristic.

### 3.5 Conclusions

In this chapter, we have studied the minimum forwarding set problem (MFSP) in the context of wireless ad hoc networks. Leveraging the practical characteristics of the application environment, we have proposed a polynomial time algorithm to build an optimal solution within this class of solutions to a given instance of the MFSP problem. To the best of our knowledge, our algorithm is the first polynomial time solution to the MFSP problem under the unit disk coverage model for wireless transmission.

We expect the work presented in this chapter to have an impact on the design and development of new algorithms for several wireless network applications including energy efficient multicast and broadcast protocols; energy efficient topology control protocols; and energy efficient virtual backbone construction protocols for wireless ad hoc networks and sensor networks. Our ongoing work in this direction includes development of new energy efficient multicast and broadcast protocols; an extension of the presented algorithm for wireless network environments where nodes use non-uniform circular coverage areas for wireless transmissions; and adaptation of the algorithm for route discovery components of popular ad hoc routing protocols including OLSR and AODV.

# CHAPTER 4
# MFSP FOR VARIABLE DISKS

## 4.1 Preliminaries

In this section, we first present some preliminary information on the practical context of the MFSP problem. We then present observations on geometric relations about intersecting disks and use these relations to introduce a theorem that we use in the construction of our algorithm.

### 4.1.1 Problem Formulation

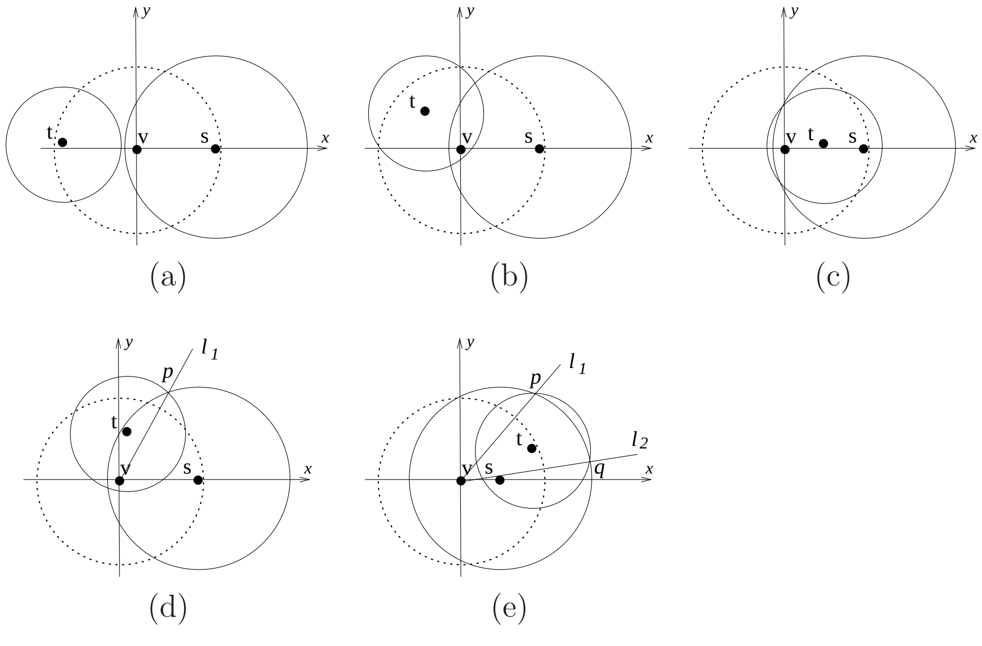
Most previous studies use a unit disk to represent the shape of the effective coverage area of wireless transmissions [26],[27]. In our previous work [6], we propose the first polynomial time algorithm to solve the MFSP problem under unit disk coverage model. In this chapter, we propose a polynomial time algorithm to solve the MFSP problem under variable size disk coverage model. The new coverage model better approximates the real life application scenarios and therefore our solution in this chapter provides better results when used in such scenarios.

Similar to the previous work, we require the availability of 2-hop neighborhood information. The required information includes (1) the identities of the 1-hop and 2-hop neighbors and (2) a radial ordering (which we define in Section 4.1.3) of the 2-hop neighbors with respect to the broadcasting node. The availability of the position information for the nodes is sufficient to compute the radial ordering of the 2-hop neighbors. Calinescu proposed methods to calculate 2-hop neighborhood information (identities and positions) for the cases where GPS or distance and angle information is available with a message complexity of $O(n)$ where $n$ is the total number of the nodes in the network [10].

### 4.1.2 Intersection Characteristics of Disks

In this section, we study some intersection characteristics of disks within the context of our problem. We then present a geometric property that we use in the formation of our algorithm.

Consider three points, $v$, $s$, and $t$, located on a two-dimensional space. In the context of wireless transmission, these points represent the locations of the three nodes named as nodes $v$, $s$ and $t$. Assume that $v$ is located at the origin of the coordinate space and $s$ is to the exact east of $v$. Assume also that both $s$ and $t$ are 1-hop neighbors of $v$, i.e., $\{s, t\} \in N(v)$. Given a node $i$, let $D_i$ represent the disk and the area covered by its wireless transmission and let $C_i$ represent the circle enclosing the coverage area $D_i$. In the rest of this section, we study the relation between $D_{s/v}$ and $D_{t/v}$ that represent the coverage of $D_s$ and $D_t$ outside the coverage area $D_v$, respectively.



(a) (b) (c)

(d) (e)

Figure 4.1. Some geometric relations of intersecting disks.

Figure 4.1 shows the possible scenarios for the coverage areas $D_s$ and $D_t$ in relation to $D_v$. Note that any two circles, $C_s$ and $C_t$, can intersect at most two times (we are not interested in the scenario where the circles $C_s$ and $C_t$ are tangent to each other - i.e., where they intersect once). Figure 4.1-a shows an example case where $C_s$ and $C_t$ intersect zero times (the case where $D_s$ (or $D_t$) completely encloses $D_t$ (or $D_s$) is not an interesting case for us). In this case, we have $D_{s/v} \cap D_{t/v} = \varnothing$. Figure 4.1-b and 4.1-c show two possible cases where $C_s$ and $C_t$ intersect twice within $D_v$. In the first case, we have $D_{s/v} \cap D_{t/v} = \varnothing$ and in the second case we have $D_{s/v} \subset D_{t/v}$ or $D_{t/v} \subset D_{s/v}$. Figure 4.1-d shows an example case where $C_s$ and $C_t$ intersect twice one within $D_v$ and the other outside $D_v$. Finally, Figure 4.1-d shows an example case where $C_s$ and $C_t$ intersect twice outside $D_v$.

### 4.1.3 Definitions

Having presented potential intersection scenarios of the disks, we now introduce a new term that we will use in the rest of the chapter. For the definitions of *radial order*, *radially continuous neighbor (RCN) interval*, *maximum coverage interval (MCI)* and *interleaving coverage* we refer reader to previous chapter.

**Definition 4.1 (Domination)** *Let* $\{s, t\} \in N(v)$ *and* $\{a_i, a_j, a_k\} \in N_2(v)$ *such that* $\{a_i, a_k\} \in N(s)$, $\{a_j\} \notin N(s)$, $\{a_j\} \in N(t)$, $\{a_i, a_k\} \notin N(t)$, *and* $a_i$, $a_j$, *and* $a_k$ *are radially ordered as* $a_i > a_j > a_k$. *Then, w.r.t. coverage relation between* $s$ *and* $t$, *we say that* $s$ *dominates* $t$ *and represent it as* $s \Delta t$.

### 4.1.4 Coverage Properties

In this section, we introduce three important properties that we use in building our algorithm for the MFSP problem.

**Theorem 4.1 (Non-Interleaving Property)** *In an instance of the MFSP problem (i.e., a node* $v$ *and its 1-hop and 2-hop neighbor sets* $N(v)$ *and* $N_2(v)$*), no two nodes* $\{s, t\} \in N(v)$ *can have interleaving coverage.*

`Proof of Theorem 4.1`: Note that interleaving is considered between two nodes $\{s, t\} \in N(v)$. We now study the intersection characteristics of $C_{s/v}$ and $C_{t/v}$ to prove the theorem. As we stated previously, there are four cases to be considered as follows:

**Case 1:** $C_{s/v}$ and $C_{t/v}$ intersect zero times as in Figure 4.1-a. In this case, $D_{s/v}$ and $D_{t/v}$ have two disjoint MCIs that are radially separate from each other.

**Case 2:** $C_{s/v}$ and $C_{t/v}$ intersect twice in $D_v$ as in Figure 4.1-b or 4.1-c. In the first case, we have $D_{s/v} \cap D_{t/v} = \varnothing$ and in the second case we have $D_{s/v} \subset D_{t/v}$ (or $D_{t/v} \subset D_{s/v}$). In both cases, there is no possibility of having interleaving coverage between $D_{s/v}$ and $D_{t/v}$.

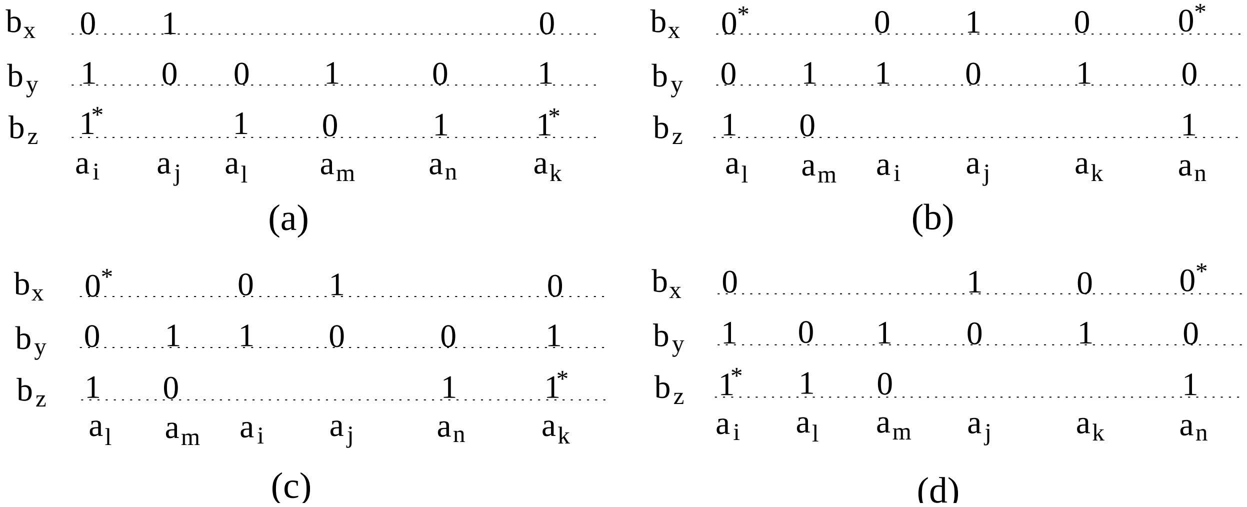
**Case 3:** $C_{s/v}$ and $C_{t/v}$ intersect twice one in $D_v$ and one outside $D_v$ as shown in Figure 4.1-d. Consider a line $l_1$ that originates at $v$ and crosses over $p$ which is the intersection point of $C_{s/v}$ and $C_{t/v}$. Observe that the nodes in $D_{s/v} \cap D_{t/v}$ can be covered by both $s$ and $t$. Also observe that the nodes in $D_{s/v/t}$ and $D_{t/v/s}$ are radially separate from each other, i.e., $l_1$ radially separates $D_{s/v/t}$ and $D_{t/v/s}$ into two non-interleaving RCNs. This then indicates that there are no two points $r_1$ and $r_2$ such that $r_1 \in D_{s/v/t}$ above $l_1$ and $r_2 \in D_{t/v/s}$ below $l_1$ to create an interleaving coverage scenario.

**Case 4:** $C_{s/v}$ and $C_{t/v}$ intersect twice outside of $D_v$ as shown in Figure 4.1-e. Similar to the previous case, let $l_1$ and $l_2$ be lines originating from $v$ and crossing the intersection points $p$ and $q$ of the two circles $C_{s/v}$ and $C_{t/v}$. From the intersection of two circles, the coverage area of $D_{t/v/s}$ is confined by $l_1$ and $l_2$. Therefore, having a single MCI to cover outside $D_{s/v}$, $t$ cannot have an interleaving coverage with $s$. $\square$

**Theorem 4.2 (Non-Domination Property)** *Given an instance of MFSP problem at* $v$*, there exists at least one node* $b_i \in N(v)$ *that is not dominated by any other node* $b_j \in N(v) - \{b_i\}$.

`Proof of Theorem 4.2`: We prove the theorem by showing that domination is an acyclic relation. Consider an instance of MFSP problem at $v$ where $\{b_x, b_y, b_z\} \in N(v)$. We show that if $b_y \Delta b_x$ and $b_z \Delta b_y$, then either $b_z \Delta b_x$ or $(N(b_x) \cap N_2(v)) \subset (N(b_z) \cap N_2(v))$.

By definition, $b_y \Delta b_x$ requires that there exist three nodes $\{a_i, a_j, a_k\} \in N_2(v)$ with a radial order $(a_i < a_j < a_k)$ such that $a_i$ and $a_k$ are covered by $b_y$ but not by $b_x$ and $a_j$ is covered by $b_x$ but not by $b_y$. Similarly $b_z \Delta b_y$ requires that there exist three nodes (not necessarily disjoint from the previous three) $\{a_l, a_m, a_n\} \in N_2(v)$ with a radial order $(a_l < a_m < a_n)$ such that $a_l$ and $a_n$ are covered by $b_z$ but not by $b_y$ and $a_m$ is covered by $b_y$ but not by $b_z$. When we consider the radial order among the nodes $\{a_i, a_j, a_k, a_l, a_m, a_n\}$, there are four possible outcomes for the first and the last node pairs as (1) $(a_i, \ldots, a_k)$, (2) $(a_l, \ldots, a_n)$, (3) $(a_l, \ldots, a_k)$, and (4) $(a_i, \ldots, a_n)$. We now consider the overall coverage scenarios for each of these cases and show that $b_x \Delta b_z$ cannot occur in none of these cases.



(a) (b)

(c) (d)

Figure 4.2. Cases for domination relations among $b_x$, $b_y$ and $b_z$.

**Case 1** $(a_i, \ldots, a_k)$**:** Consider the scenario in Figure 4.2-a that represents this case. In this case, $b_z$ covers $a_i$ and $a_k$ as otherwise $b_y$ and $b_z$ have interleaving coverage contradicting with Theorem 4.1. In addition, $b_z$ should cover any node $a_{before} < a_i$ or $a_{after} > a_k$ that $b_x$ covers. This is because such a node $a_{before}$ or $a_{after}$ covered by $b_x$ has to be covered by $b_y$ (so that $b_x$ and $b_y$ do not interleave). In this case, any node $a_{before}$ or $a_{after}$ covered by $b_y$ has to be covered by $b_z$ (so that $b_y$ and $b_z$ do not interleave). Finally, if $b_x$ covers a node $a_p$ where $(a_i < a_p < a_k)$ such that $a_p$ is not covered by $b_z$, then $b_z \Delta b_x$ on $(a_i, a_p, a_k)$. Otherwise, $(N(b_x) \cap N_2(v)) \subset (N(b_z) \cap N_2(v))$.

**Case 2** $(a_l, \ldots, a_n)$**:** Consider the scenario in Figure 4.2-c that represents this case. In this case, $b_x$ should not cover $a_l$ and $a_n$ as otherwise $b_x$ and $b_y$ would interleave. Similarly, $b_x$ should not cover $a_l$ as otherwise $b_x$ and $b_y$ would interleave. In this scenario, $b_z$ should cover all nodes $a_{before} < a_l$ or $a_{after} > a_n$ that $b_x$ covers. This is because any $a_{before}$ or $a_{after}$ covered by $b_x$ has to be covered by $b_y$ (so that $b_x$ and $b_y$ do not interleave) and any node $a_{before}$ or $a_{after}$ covered by $b_y$ should be covered by $b_z$ (so that $b_y$ and $b_z$ do not interleave). Finally, if $b_x$ covers a node $a_p$ where $(a_l < a_p < a_n)$ such that $a_p$ is not covered by $b_z$, then $b_z \Delta b_x$ on $(a_l, a_p, a_n)$. Otherwise $(N(b_x) \cap N_2(v)) \subset (N(b_z) \cap N_2(v))$.

**Case 3** $(a_l, \ldots, a_k)$**:** Consider the scenario in Figure 4.2-b that represents this case. In this case, $b_z$ covers $a_k$ as otherwise $b_y$ and $b_z$ would interleave. In addition, $b_z$ should cover any node $a_{before} < a_l$ or $a_{after} > a_k$ that $b_x$ covers. This is because such a node $a_{before}$ or $a_{after}$ covered by $b_x$ has to be covered by $b_y$ (so that $b_x$ and $b_y$ do not interleave). In this case, any node $a_{before}$ or $a_{after}$ covered by $b_y$ has to be covered by $b_z$ (so that $b_y$ and $b_z$ do not interleave). Finally, if $b_x$ covers a node $a_p$ where $(a_l < a_p < a_k)$ such that $a_p$ is not covered by $b_z$, then $b_z \Delta b_x$ on $(a_l, a_p, a_k)$. Otherwise, $(N(b_x) \cap N_2(v)) \subset (N(b_z) \cap N_2(v))$.

**Case 4** $(a_i, \ldots, a_n)$**:** Consider the scenario in Figure 4.2-d that represents this case. In this case, $b_z$ covers $a_i$ as otherwise $b_y$ and $b_z$ would interleave. In this case, $b_x$ should not cover $a_n$ as otherwise $b_x$ and $b_y$ would interleave. In this scenario, $b_z$ should cover all nodes $a_{before} < a_i$ and $a_{after} > a_n$ that $b_x$ covers. This is because any $a_{before}$ or $a_{after}$ covered by $b_x$ has to be covered by $b_y$ (so that $b_x$ and $b_y$ do not interleave) and any node $a_{before}$ or $a_{after}$ covered by $b_y$ should be covered by $b_z$ (so that $b_y$ and $b_z$ do not interleave). Finally, if $b_x$ covers a node $a_p$ where $(a_i < a_p < a_n)$ such that $a_p$ is not covered by $b_z$, then $b_z \Delta b_x$ on $(a_i, a_p, a_n)$. Otherwise $(N(b_x) \cap N_2(v)) \subset (N(b_z) \cap N_2(v))$.

We now show that domination is an acyclic relation among $n$ such nodes. Assume that there is a cyclic domination relation among $n$ nodes $\{b_1, b_2, b_3, \ldots, b_n\} \in N(v)$ as $(b_1 \Delta b_2, b_2 \Delta b_3, b_3 \Delta b_4, \ldots, b_{n-1} \Delta b_n, b_n \Delta b_1)$. From the above discussion, we know that there are two cases for the relation between $b_n$ and $b_2$, i.e., either $b_n \Delta b_2$ or $(N(b_2) \cap N_2(v)) \subset$

$(N(b_n) \cap N_2(v))$. Now, consider the relation between $b_2$ and $b_3$. From the relation between $b_n$ and $b_2$, if we have $b_n \Delta b_2$, it is either $b_n \Delta b_3$ or $(N(b_3) \cap N_2(v)) \subset (N(b_n) \cap N_2(v))$; on the other hand, if $(N(b_2) \cap N_2(v)) \subset (N(b_n) \cap N_2(v))$, we have the same result that either $b_n \Delta b_3$ or $(N(b_3) \cap N_2(v)) \subset (N(b_n) \cap N_2(v))$. If we continue to iterate the cycle, at some point we have $b_n \Delta b_{n-1}$ or $(N(b_{n-1}) \cap N_2(v)) \subset (N(b_n) \cap N_2(v))$. But since $b_{n-1} \Delta b_n$, $b_n \Delta b_{n-1}$ is not possible unless $b_n$ and $b_{n-1}$ interleave. In addition, $(N(b_{n-1}) \cap N_2(v)) \neq (N(b_n) \cap N_2(v))$ as $b_{n-1}$ should cover at least two nodes in $N_2(v)$ to satisfy that $b_{n-1} \Delta b_n$. In both cases, we have a contradiction indicating that domination relation cannot be a cyclic relation. $\square$

## 4.2 Solution

In this section, we present a polynomial time algorithm to solve the MFSP problem under disk coverage model. The main idea in our approach is to break a given problem into subproblems and use the solutions of the subproblems in building a solution to the given instance of the problem. Consider an instance of an MFSP problem at a node $v$. Let $n$ be the number of 2-hop neighbors of $v$, i.e., $n = |N_2(v)|$. Assume that nodes $a_i \in N_2(v)$ are sorted based on their radial order with respect to $v$. The algorithm presented below executes in $n$ rounds. At a round $j$, we divide $N_2(v)$ into $n$ different RCNs each covering $j$ consecutive nodes in $N_2(v)$. We use the tuple $(a_i, j)$ to represent the consecutive interval of nodes $(a_i, a_{i+1}, \ldots, a_{i+j-1}) \in N_2(v)$ considered by each of these subproblems[1]. The subproblems are then solved by using the solutions of the smaller size subproblems solved in previous rounds.

Let $L_{min}(a_i, j)$ be the list of minimum number of first hop neighbors of $v$ that are essential to cover the interval $(a_i, j)$. At round $n$, the algorithm returns $n$ solutions as $L_{min}(a_i, n)$ for $i = [1, n]$. The optimal solution is given by the $L_{min}(a_1, n)$.

---

[1] Note that the arithmetic in computing the subscripts of the nodes preserves radial circularity.

### 4.2.1 Solution Approach

Our algorithm uses a bottom-up approach to build an optimal solution. Given an RCN interval $(a_i, j)$, we assume that we know the optimal solutions for all RCNs of $(a_i, j)$. With this knowledge, we use the below approach to find the optimal solution $S$ for $(a_i, j)$.

Consider an optimal solution $S$ for $(a_i, j)$. Let $S' \subset S$ be the set of nodes that cover $a_i$ and $S'' = S - S'$. From Theorem 4.2, we know that $\exists b_x \in S'$ that is not dominated by any other nodes in $S'$. Assume that $b_x$ has $t$ MCI's, $(B_1, \ldots, B_t)$, where $(1 \leq t \leq \lfloor j/2 \rfloor)$ in $(a_i, j)$. We refer to the RCNs that $b_x$ does not cover as $R_k$ for $k \in [1, t]$. Using $B_k$s and $R_k$s, we can represent $(a_i, j)$ as $(a_i, j) = (B_1 \cup R_1 \cup B_2 \cup R_2 \cup \ldots \cup B_{t-1} \cup R_{t-1} \cup B_t \cup R_t)$. In this setup, we would like to find the solution $S$ for $(a_i, j)$ assuming that we know $b_x \in S$. Note that we do not know $b_x$ but we know that such a node exists in $S$.

**Definition 4.2** *Let $Q \subset N_2(v)$ be a subset of two-hop neighbors of $v$. Then, $c(Q) \subset N(v)$ is a minimum cardinality 1-hop node set covering all the nodes in $Q$. Note that $Q$ does not have to be a single RCN but may contain multiple RCNs.*

As introduced above, if we write $(a_i, j) = (B_1 \cup R_1 \cup B_2 \cup R_2 \cup \ldots \cup B_{t-1} \cup R_{t-1} \cup B_t \cup R_t)$, then $S$ can be rewritten as $S = c(B_1 \cup R_1 \cup B_2 \cup R_2 \cup \ldots \cup B_{t-1} \cup R_{t-1} \cup B_t \cup R_t) = c(R_1 \cup R_2 \cup \ldots \cup R_t) \cup \{b_x\}$. Note that we know the solution $c(R_1 \cup B_2 \cup R_2 \cup \ldots \cup B_{t-1} \cup R_{t-1} \cup B_t \cup R_t)$ but it may not be the same as $c(R_1 \cup R_2 \cup \ldots \cup R_t)$. Our task now is to find the optimal solution $c(R_1 \cup R_2 \cup \ldots \cup R_t)$ for $(R_1 \cup R_2 \cup \ldots \cup R_t)$. We present a dynamic programming approach to build the solution $c(R_1 \cup R_2 \cup \ldots \cup R_t)$.

Consider the basic case for $(R_1 \cup R_2)$. There are two alternative cases for the solution $c(R_1 \cup R_2)$ as

$$|c(R_1 \cup R_2)| = |c(R_1 \cup B_2 \cup R_2)|, \text{ or} \tag{4.1}$$

$$|c(R_1 \cup R_2)| = |c(R_1 \cup B_2 \cup R_2)| - 1 \tag{4.2}$$

where $|c(R_1 \cup R_2)|$ represents the cardinality of the solution $c(R_1 \cup R_2)$. Note that we know the solution for the first case above. For the second case, the coverage of $b_x$ on $B_2$ results in a better solution. The following lemma presents an observation for the second solution case above.

**Lemma 4.1** *If* $|c(R_1 \cup R_2)| = |c(R_1 \cup B_2 \cup R_2)| - 1$, *then* $|c(R_1 \cup R_2)| = |c(R_1)| + |c(R_2)|$.

`Proof of Lemma 4.1`: The above equation is not accurate if $\exists b_y \in c(R_1 \cup R_2)$ which covers nodes from both $R_1$ and $R_2$. Assume such a $b_y$ exists. Recall that by definition $b_x$ does not cover any nodes in $R_1$ or in $R_2$. Now, if $b_y \in S'$, then $b_y$ has to cover all nodes in $B_2$ as otherwise $b_y$ dominates $b_x$ (recall that $b_x$ is not dominated by any node in $S'$). Alternatively, if $b_y \in S''$, then $b_y$ does not cover $a_i$. But in this case, it should cover all nodes in $B_2$ as otherwise $b_x$ and $b_y$ will have an interleaving coverage. Consequently, if $b_y$ covers all nodes in $B_2$, this results in $|c(R_1 \cup R_2)| = |c(R_1 \cup B_2 \cup R_2)|$ which contradicts with the initial assumption. As a result, if $|c(R_1 \cup R_2)| = |c(R_1 \cup B_2 \cup R_2)| - 1$, then $b_y$ cannot cover nodes from both $R_1$ and $R_2$ simultaneously, i.e., $|c(R_1 \cup R_2)| = |c(R_1)| + |c(R_2)|$. □

The above observation can be generalized for an arbitrarily large $(R_1 \cup R_2 \cup \ldots R_t)$. In this generic case, there are two alternatives for the solution $c(R_1 \cup R_2 \cup \ldots R_t)$ as

$$|c(R_1 \cup R_2 \cup \ldots \cup R_t)| = |c(R_1 \cup B_2 \cup R_2 \cup \ldots \cup B_t \cup R_t)|, \text{ or} \tag{4.3}$$

$$|c(R_1 \cup R_2 \cup \ldots \cup R_t)| = |c(R_1 \cup B_2 \cup R_2 \cup \ldots \cup B_t \cup R_t)| - 1. \tag{4.4}$$

Similar to the basic case above, we know the solution for the case in Equation 4.3. For the case in Equation 4.4, we claim that we can find the optimal solution by splitting the interval $(R_1 \cup R_2 \cup \ldots \cup R_t)$ into two as $(R_1 \cup \ldots \cup R_h)$ and $(R_{h+1} \cup \ldots \cup R_t)$. In this case the optimal solution is given by

$$|c(R_1 \cup R_2 \cup \ldots \cup R_t)| = |c(R_1 \cup \ldots \cup R_h)| + |c(R_{h+1} \cup \ldots \cup R_t)|. \tag{4.5}$$

Equation 4.5 may not be correct if $\exists b_y$ such that $b_y \in c(R_1 \cup \ldots \cup R_h)$ and $b_y \in c(R_{h+1} \cup \ldots \cup R_t)$. Assume that such a $b_y$ exists. If $b_y \in S'$, then $b_y$ has to cover all nodes in $B_{h+1}$ as otherwise $b_y$ dominates $b_x$ (recall that $b_x$ is not dominated by any node in $S'$). Alternatively, if $b_y \in S''$, then $b_y$ does not cover $a_i$. But in this case, it should cover all nodes in $B_{h+1}$ as otherwise $b_x$ and $b_y$ will have an interleaving coverage. Consequently, if $b_y$ covers all nodes in $B_{h+1}$, this results in $|c(R_1 \cup R_2 \cup \ldots \cup R_t)| = |c(R_1 \cup R_2 \cup \ldots \cup B_{h+1} \cup \ldots \cup R_t)|$. As a result, if $|c(R_1 \cup R_2 \cup \ldots \cup R_t)| = |c(R_1 \cup B_2 \cup R_2 \cup \ldots \cup B_t \cup R_t)| - 1$ (i.e., Equation 4.4 above holds), then there exists a split point $b_{\bar{h}+1}$ such that the corresponding node $b_{\bar{y}}$ cannot cover nodes from both $c(R_1 \cup \ldots \cup R_{\bar{h}})$ and $c(R_{\bar{h}+1} \cup \ldots \cup R_t)$ simultaneously, i.e., $|c(R_1 \cup R_2 \cup \ldots \cup R_t)| = |c(R_1 \cup \ldots \cup R_{\bar{h}})| + |c(R_{\bar{h}+1} \cup \ldots \cup R_t)|$. If, on the other hand, there does not exist such a $b_{\bar{h}}$, then the solution is given by Equation 4.3.

Note that in order to find $c(R_1 \cup R_2 \cup \ldots \cup R_t)$ based on Equation 4.4, we need to try all values for $h \in [1, t]$ where $t$ is bounded by $\lfloor j/2 \rfloor$ in $(a_i, j) = (B_1 \cup R_1 \cup B_2 \cup R_2 \cup \ldots \cup B_t \cup R_t)$. Based on this the general solution for $c(R_1 \cup R_2 \cup \ldots \cup R_t)$ is given by

$$\begin{aligned} |c(R_1 \cup R_2 \cup \ldots \cup R_t)| \;=\; & min(|c(R_1 \cup B_2 \cup R_2 \cup \ldots \cup B_t \cup R_t)|, \\ & min((|c(R_1 \cup \ldots \cup R_h)| + |c(R_{h+1} \cup \ldots \cup R_t)|), \forall h \in [2, \lfloor j/2 \rfloor])). \end{aligned} \tag{4.6}$$

Using Equation 4.6, we can find $c(R_1 \cup R_2 \cup \ldots \cup R_t)$ in $O(n^3)$ as follows. We know the optimal solutions for $c(R_1), c(R_2), \ldots, c(R_t)$. In the second step, we find the optimal solutions for each $c(R_k \cup R_{k+1})$ for $1 \leq k \leq t-1$ as

$$|c(R_k \cup R_{k+1})| = min(|c(R_k \cup B_k \cup R_{k+1})|, |c(R_k)| + |c(R_{k+1})|).$$

In the third step, we find the optimal solutions for $c(R_k \cup R_{k+1} \cup R_{k+2})$ for $1 \leq k \leq (t-2)$ as

$$\begin{aligned} |c(R_k \cup R_{k+1} \cup R_{k+2})| \;=\; & min(|c(R_k \cup B_k \cup R_{k+1} \cup B_{k+1} \cup R_{k+2})|, \\ & min(|c(R_k \cup R_{k+1})| + |c(R_{k+2})|, |c(R_k)| + |c(R_{k+1} \cup R_{k+2})|)). \end{aligned} \tag{4.7}$$

We continue this process for a total of $t$ steps. In the first step, we have $t$ optimal solutions each corresponding to an interval $R_k$. In the second step, we have $t-1$ optimal solutions corresponding to consecutive interval pairs of $R_k \cup R_{k+1}$. In the $t^{th}$ step, we have the optimal solution for the entire interval of $R_1 \cup R_2 \cup \ldots \cup R_t$. Overall, the process is similar to building a pyramid starting with $t$ blocks at the bottom of it and finishing with a single block at step $t$.

There are $O(n^2)$ entries in the pyramid and we make at most $O(n)$ operations (i.e., the number of possible splits) to calculate the optimal solution for each entry. If we know which one hop node $b \in N(v)$ corresponds to $b_x$ prior to above calculation, we can find the optimal solution for the entire interval as above. Trials with incorrect candidates $b_i \neq b_x$ will result in suboptimal solutions. The running time will then be $O(n^3)$. Unfortunately, we do not know $b_x$ in advance but can try all possible candidates. The number of nodes in $N(v)$ that can cover $a_i$ is at most $|N(v)| = m$. This means that the solution for a given interval $(a_i, j)$ can be found in $O(mn^3)$.

### 4.2.2 Algorithm

So far, we presented the solution for a single interval $(a_i, j)$ for $1 \leq i, j \leq n$. We can find the solutions for all such intervals in a bottom up fashion as follows. The solution at the first step is trivial, i.e., $(a_i, 1)$ for $1 \leq i \leq n$ is trivial. The above procedure allows us to find a solution for an interval $(a_i, k)$ for $1 \leq i \leq n$ by using the optimal solutions for $(a_i, j)$ where $1 \leq j \leq k$. Finally, at the $n^{th}$ step, we find the optimal solution for $(a_1, n)$. As a result, the overall complexity of the algorithm is given by $O(mn^5)$ where $O(n^2)$ comes from the number of $(a_i, j)$ intervals and $O(mn^3)$ comes from the computational complexity of the solution for each such interval. Finally, Figure 4.3 presents a pseudocode of the overall algorithm.

In the algorithm outlined in Figure 4.3, the term $L_{min}(a_i, j)$ considers the solution for an interval $(a_i, j)$ of $N_2(v)$. On the other hand, the term $P_{min}(R_v, y)$ considers the solution for $y$ intervals that $b_s$ does not cover, i.e., $(R_v, y) = (R_v \cup R_{v+1} \cup \ldots \cup R_{v+y-1})$. Finally, $N_{min}(a_i, j)$

represents the size of the solution $L_{min}(a_i, j)$, i.e., $N_{min}(a_i, j) = |L_{min}(a_i, j)|$ for $(a_i, j)$ and $K_{min}(R_v, y)$ represents the size of the solution $P_{min}(R_v, y)$, i.e., $K_{min}(R_v, y) = |P_{min}(R_v, y)|$.

## 4.3 Evaluations

From a practicality point of view, the existing heuristics introduce smaller running time complexities as compared to the presented algorithm. Therefore, in this section, we evaluate the performance of the two popular heuristics, i.e., heuristics by Chvatal [13] and by Qayyum et al. [35] from Section 2.3, in terms of their approximation ratios under different network models and coverage ranges. The overall result of our evaluations indicate that the existing heuristics perform quite well under the considered network setups. Note that prior to our algorithm, the performance of these heuristics under practical application scenarios were not known.

In our evaluations, we build connected network topologies by placing a number of nodes on a 7x7 grid. We use topologies with different sizes but present results for 300 node topologies only. The experiments in other cases give similar results. Each node $v$ in the network has a maximum transmission range $r^v_{max}$ that is randomly selected from a transmission range $t - range$. We use several different $t - range$ values to evaluate the performance of the heuristics in different scenarios. Finally, we use two different types of node deployment strategies. The first strategy is random node placement strategy where we randomly place nodes on the grid. The second strategy is a grid-based node placement strategy where we divide the 7x7 grid deployment area into small squares and place a node at the intersection points of each square. The results of our experiments on 300 node topologies are summarized as average $MFS(v)$ values for each experiment in Table 4.3 and Table 4.3 for the two node placement strategies respectively. In the table, the columns $|N(v)|$ and $|N_2(v)|)$ represent the average numbers of one-hop and two-hop neighbors for each node $v$. The column '#Mand. Nodes' represent the average number of mandatory nodes in $N(v)$ as used in the initial part of the Qayyum et al. heuristic.

Table 4.1. Performance of heuristics under 300-node random networks.

| $T - range$ | $\|N(v)\|$ | $\|N_2(v)\|$ | $\#Mand.Nodes$ | Opt.soln. | Chvatal | Qayyum |
|---|---|---|---|---|---|---|
| [0.1-2.1] | 24.9 | 61.1 | 2.64 | 3.11 | 3.20 | 3.11 |
| [0.5-1.5] | 26.5 | 63.5 | 3.09 | 3.59 | 3.69 | 3.59 |
| [0.5-2.0] | 17.3 | 38.8 | 2.96 | 3.52 | 3.59 | 3.52 |
| [0.5-3.0] | 46.0 | 111.3 | 2.74 | 3.29 | 3.49 | 3.30 |
| [1.0-3.0] | 58.7 | 130.5 | 2.98 | 3.66 | 3.90 | 3.67 |
| [0.5-0.5] | 4.3 | 4.8 | 1.67 | 2.10 | 2.12 | 2.10 |
| [1.0-1.0] | 16.9 | 31.5 | 3.51 | 4.77 | 4.78 | 4.94 |
| [1.5-1.5] | 35.1 | 67.7 | 3.83 | 5.49 | 5.76 | 5..53 |
| [2.0-2.0] | 57.1 | 103.5 | 3.60 | 5.62 | 5.87 | 5.70 |
| [3.0-3.0] | 119.1 | 149.1 | 1.79 | 4.22 | 4.58 | 4.44 |

Table 4.2. Performance of heuristics under 300-node grid networks.

| $T - range$ | $\|N(v)\|$ | $\|N_2(v)\|$ | $\#Mand.Nodes$ | Opt.soln. | Chvatal | Qayyum |
|---|---|---|---|---|---|---|
| [0.1-2.1] | 21.4 | 55.8 | 2.82 | 3.03 | 3.09 | 3.03 |
| [0.5-1.5] | 16.9 | 41.6 | 3.51 | 3.99 | 4.07 | 3.99 |
| [0.5-2.0] | 27.8 | 69.7 | 3.71 | 4.20 | 4.36 | 4.20 |
| [0.5-3.0] | 52.2 | 127.6 | 2.95 | 3.49 | 3.62 | 3.51 |
| [1.0-3.0] | 59.3 | 132.0 | 3.24 | 3.95 | 4.18 | 3.97 |
| [0.5-0.5] | 3.8 | 7.1 | 3.53 | 3.55 | 3.55 | 3.55 |
| [1.0-1.0] | 17.5 | 35.6 | 5.46 | 6.05 | 6.11 | 6.05 |
| [1.5-1.5] | 36.1 | 69.5 | 4.04 | 5.28 | 5.45 | 5.28 |
| [2.0-2.0] | 53.1 | 96.9 | 3.32 | 4.90 | 5.21 | 4.98 |
| [3.0-3.0] | 116.7 | 149.3 | 2.48 | 4.44 | 4.96 | 4.56 |

Our main observation from these experiments is that both heuristics perform quite well under the considered scenarios. The performance of the heuristics seem to be better for arbitrary disk graphs as compared to unit disk graphs. In most of the experiments, the number of mandatory nodes make a large part of the solution set and the solution size increases slightly as the number of first hop or second hop nodes increases.

Note that prior to our solution presented in this chapter, quantifying the performance of the existing heuristics were difficult. Therefore, our algorithm also provides a strong base in developing new solutions and heuristics and measuring their performance with respect to the optimal solution.

## 4.4 Conclusions

In this chapter, we have presented a polynomial time solution to the minimum forwarding set problem (MFSP), a practical problem that appears in developing efficient algorithms for several communication applications in wireless ad hoc networks and sensor networks. In our earlier work, we presented a solution to the problem under unit disk coverage model for wireless transmission. In this chapter, we have considered arbitrary disk model as a more general model for wireless transmission and developed an optimal solution to the problem. We believe that the results presented in this chapter can be used in building several efficient communication services including energy efficient multicast and broadcast, energy efficient topology control protocols, and energy efficient virtual backbone construction protocols for wireless ad hoc networks and sensor networks.

```
**INPUT:** An instance of MFSP problem at a node $v$:
  **1.** $N(v) := \{b_1, \dots, b_m\}$ **2.** $N_2(v) := \{a_1, \dots, a_n\}$
  **3.** An $m \times n$ coverage matrix $A$ as defined
**OUTPUT:** $MFS(v) = L_{min}(a_1, n)$
**FOR** $i := 1$ to $n$ /*initialization*/
  $N_{min}(a_i, 1) := 1$, $L_{min}(a_i, 1) := (b)$ where $b \in N(v)$ and $a_i \in N(b)$
**FOR** $j := 2$ to $n$ /* for each round */
  **FOR** $i := 1$ to $n$ /* for each interval $(a_i, j)$*/
    $N_{min}(a_i, j) := \infty$, $L_{min}(a_i, j) := \emptyset$
    **FOR** $s := 1$ to $m$ /*for all candidates to be $b_x$*/
      **IF** $A_{si} = 0$ **THEN break** /* it should cover $a_i$ */
      Choose $b_s$ as $b_x$
      Determine $B_1, R_1, B_2, R_2, B_3, ..., R_{t-1}, B_t, R_t$ in $O(n)$
      Representation: $R_p^s$ := first node of $R_p$ , $R_p^l$ :=length of interval $R_p$
      **FOR** $p := 1$ to $t$ /*initialization*/
        $K_{min}(R_p, 1) := N_{min}(R_p^s, R_p^l), P_{min}(R_p, 1) := L_{min}(R_p^s, R_p^l)$
      **FOR** $y := 2$ to $t$ /* for each round */
        **FOR** $v := 1$ to $t - y + 1$ /* for each starting $R_v$*/
          let $k = |(R_v \cup B_{v+1} \cup \ldots \cup B_{v+y-1} \cup R_{v+y-1})|$
          $K_{min}(R_v, y) := N_{min}(R_v^s, k)$
          $P_{min}(R_v, y) := L_{min}(R_v^s, k)$
          **FOR** $h := 2$ to $y - 1$ /* for each possible split */
            **IF** $K_{min}(R_v, y) > K_{min}(R_v, h) + K_{min}(R_{v+h}, y - h)$
              $K_{min}(R_v, y) := K_{min}(R_v, h) + K_{min}(R_{v+h}, y - h)$
              $P_{min}(R_v, y) := P_{min}(R_v, h) + P_{min}(R_{v+h}, y - h)$
    **IF**$N_{min}(a_i, j) > K_{min}(R_1, t) + 1$
      $N_{min}(a_i, j) := K_{min}(R_1, t) + 1$
      $L_{min}(a_i, j) := P_{min}(R_1, t) \cup \{b_s\}$
return $L_{min}(a_1, n)$
```

Figure 4.3. Outline of the algorithm.

# CHAPTER 5

# MFSP WITH COLLISION AWARENESS

## 5.1 Preliminaries

Ad hoc networks consist of nodes which communicate with each other via radio frequency. If two wireless nodes make transmission in the same frequency and if a third node receives these messages at the same time, these messages will collide and the third node will not be able to process these messages properly. Original MFSP does not consider the effect of these collisions. In this study we examine the effects of collisions. Message should be successfully delivered to all 2-hop neighbors. Since messages might collide, we need to design a proper schedule to determine the transmission periods for 1-hop neighbors. One simple way of achieving this is designing rounds. Each round represents a certain amount of delay. In each round only a subset of 1-hop neighbors make transmissions. If a 1-hop neighbor is selected for a round, it waits for the predefined amount of time and make transmission after that. The delay between two rounds should be arranged such that two transmissions from two consecutive rounds should not collide even if they cover the same area.

There are two new problems. In the first one the input is an instance of MFSP and a minimum cardinality forwarding set. Problem asks for a schedule with minimum number of rounds where each selected 1-hop neighbor makes transmission in only one round. In this schedule each 2-hop neighbor should receive the message from exactly one 1-hop neighbor for at least one round. We call this problem MFSPWCA(Minimum Forwarding Set Problem with Collision Awareness). The second problem accepts an instance of MFSP as input and asks for a schedule with minimum number of rounds.The first problem optimizes the number of transmissions and battery power whereas the second one optimizes the overall delay. In this study we focus on the first problem.

## 5.2 Reduction from Graph Coloring

MFSPWCA is NP-hard. In this section we present a reduction from Graph Coloring. The input of the Graph Coloring is a graph G:(V,E) with vertex set V and edge set E. A Coloring is a function F from V to C, where C is the set of colors used. In a valid coloring, if two vertices are connected they should be colored with different colors. Problem asks for a valid coloring with minimum number of colors. For a given instance of Graph Coloring G:(V,E) an instance of MFSPWCA can be created as follows. For each vertex in V a corresponding 1-hop neighbor is placed for MFSPWCA. Moreover a 2-hop neighbor is created for each such 1-hop neighbor which is uniquely covered by this 1-hop neighbor. For each edge (u,v) of G, a 2-hop neighbor is constructed. This 2-hop neighbor should only be connected to 1-hop neighbors corresponding to u and v. In this setting all 1-hop neighbors should be in the forwarding set because each one covers a unique 2-hop neighbor. Two 1-hop neighbors can be assigned to the same color if and only if corresponding vertices are not connected. Since each 1-hop neighbor can only be assigned to a single round, the optimal solution for MFSPWCA can be used as an optimal solution for Graph Coloring. If two 1-hop neighbors assigned to same round, corresponding vertices should be colored by the same color. This concludes the reduction. Note that, this reduction does not consider the geometrical properties of this problem. The availability of a polynomial time solution for the UDG or DG model is still unknown.

## 5.3 Heuristics

MFSPWCA is NP-hard and it is application-wise important. We have developed practically efficient heuristics as best effort solutions. We have developed two different heuristics for two different models. The first model assumes the availability of the location information for the nodes in the 2-hop neighborhood. The second model is more general, it only assumes the availability of 2-hop neighborhood information. We used first model as a tool to understand the characteristics of the problem. We used this information to develop efficient heuristics for the second model. First model is also useful for the environments where location information is readily available.

Developed heuristics accept a forwarding set $F$ as the input. The output is an assignment from forwarding set $F$ to round set $R$. In this assignment each element of $F$ should be assigned to an element of $R$. If two elements in $F$ are connected to same 2-hop neighbor, they should not be assigned to same element in $R$. The cardinality of $R$ should be minimized in this setting for best performance.

### 5.3.1 Heuristic A

In this heuristic we assume the availability of location information. We observed that the nodes assigned to the same round should be as far as possible to minimize the collisions. To utilize this information we ordered the elements in $F$. This ordering is with respect to the angle they make with the center node. This order is used in the algorithm as a measure to maximize the distance between forwarding nodes. We believe this radial order is suitable to separate forwarding nodes due to the nature of the problem.

We can find the radial order of forwarding nodes by using the location information. Moreover the angle information between center and 1-hop neighbors is enough to find the radial order. Angle information between neighboring nodes can be calculated by using multiple ultrasound receivers or directional antennas.

This heuristic works in an incremental fashion. In the first step it checks if one round is enough or not. In the second step it checks for the two rounds. In the third it checks for three rounds. If more than one round is being used, nodes are assigned to rounds in the radial order. For example in the third step, first node is assigned to round one, second node is assigned to round two, third node is assigned to round three and fourth node is assigned to round one again. By using such an assignment we try to maximize the minimum angular distance between two nodes which are assigned to the same round. In the assignment of nodes an exception should be made for the last node. In the radial order, last node might be assigned to round one. This creates high risk for a collision between first and last nodes. To handle this situation last node is assigned to the half of the round count. For example if round count is three and there are seven nodes, last node will be assigned to round two instead of round one.

At the end of each step, the algorithm performs a validity check. If two nodes in the same round, have a common 2-hop neighbor, validity check fails and algorithm continues with next step. If assignment is valid then algorithm outputs the assignment. Note that if there are f forwarding nodes algorithm ends in f steps in the worst case.

### 5.3.2 Heuristic B

This heuristic is designed for the environments where location information is not available. Since we don't have location information we can not use the optimal solution in Chapter- 3. In this heuristic the forwarding set is created with the algorithm proposed in [35]. In the first heuristic, we have used the radial order of 1-hop neighbors as a distance measure. The challenge for this model is, finding a similar distance measure with only using neighborhood information. In the radial order if two 1-hop neighbors are next to each other, they are expected to have lots of common 2-hop neighbors. We used this information to determine an imaginary radial order. Only difference between these heuristics is the construction of order. Remaining operations are identical with first heuristic.

Order construction algorithm first adds all 1-hop neighbors into a candidate list. Then it selects the 1-hop neighbor which is connected to maximum number of 2-hop neighbors. This becomes the first node in imaginary radial order and it is removed from the candidate list. After that algorithm puts a sign on the 2-hop neighbors first node covers. Then it selects the 1-hop neighbor which covers the maximum cardinality subset of these signed 2-hop neighbors. This selected 1-hop neighbor becomes the second node in the radial order and it is removed from the candidate list. After this, 2-hop neighbors covered by the second 1-hop neighbor is signed. The third node is the one which covers the maximum cardinality

Table 5.1. Results for Heuristic A.

| Topo. size | $\lvert N_1(v)\rvert$ | $\lvert N_2(v)\rvert$ | #For. Nodes | #Rounds |
|---|---|---|---|---|
| 250 | 6.83 | 10.06 | 3.21 | 1.74 |
| 500 | 14.51 | 27.06 | 4.62 | 2.32 |
| 750 | 21.09 | 43.10 | 5.42 | 2.47 |
| 1000 | 28.70 | 61.58 | 6.13 | 2.63 |
| 1250 | 35.80 | 78.38 | 6.66 | 2.75 |

subset of these signed 2-hop neighbors. This process is repeated until there is no node in the candidate list.

## 5.4 Evaluations

In this section we present the simulation results of the two heuristics explained. For these simulations the environment used is a 10 by 10 square. Selected number of nodes are placed on random locations on this square. The effective coverage is a unit disk around the center for each node.

After the construction of the network Heuristic A is applied for every node in the network. Then a new network with the same parameters is created and Heuristic B is applied. Both heuristics output the schedule for the selected forwarding nodes. Experiments are repeated for networks with different number of nodes. Results of these experiments are represented in Table- 5.1 and Table- 5.2. Results in Table- 5.1 are acquired by Heuristic A and results in Table- 5.2 are acquired by Heuristic B.

In these tables, first two column represents the average number of 1-hop and 2-hop neighbors respectively. Third column represents the average cardinality of forwarding sets. Finally, last column shows the average number of rounds used for scheduling.

These results suggest that heuristics reduce the round count significantly. In the worst case for each round only one forwarding node should be assigned. In this case umber of rounds should be equal to the number of forwarding nodes. Heuristics reduce this amount significantly. Forwarding set sizes do not vary a lot, between two forwarding set selection

Table 5.2. Results for Heuristic B.

| Topo. size | $\lvert N_1(v)\rvert$ | $\lvert N_2(v)\rvert$ | #For. Nodes | #Rounds |
|---|---|---|---|---|
| 250 | 7.08 | 9.87 | 3.09 | 1.69 |
| 500 | 14.14 | 26.14 | 4.60 | 2.40 |
| 750 | 21.14 | 42.57 | 5.49 | 2.69 |
| 1000 | 28.81 | 61.37 | 6.09 | 2.94 |
| 1250 | 35.42 | 78.04 | 6.73 | 3.09 |

algorithms. This information is consistent with our previous evaluations. A more interesting result is about the number of rounds used for each heuristic. Even the forwarding set sizes are very close for two heuristics, there is a significant gap between the number of rounds used. Moreover this gap increases with the density of the network. This outcome implies a trade-off between the availability of location information and the average delay.

## 5.5 Conclusions

In this study we have defined a new version of MFSP in which the effect of collision is considered. Collision is a natural property of wireless transmission. We have defined a simple model which assigns delays to selected forwarding neighbors. These delays are defined as rounds. In this model one simple solution is assigning a single forwarding neighbor for each round. Using the heuristics that we have designed, we have improved the performance of this simple solution. We have also demonstrated the complexity of this problem by presenting a reduction from Graph Coloring.

# CHAPTER 6

# SUMMARY

In this thesis, I have studied different versions of forwarding set problems. I developed two exact algorithms and two heuristics for these problems. These solutions are designed to be used as building blocks for the routing protocols in future.

I model MFSP as a geometrical problem. The nature of wireless communication suggests this assumption. It is difficult to predict the effective coverage area of a wireless transmission. Lots of factors affect the exact effective coverage such as obstacles in the environment and it is almost impossible to model all these factors. If environment related factors are ignored, the natural shape of a wireless transmission is a disk around the center in two dimensional space.

Based on unit disc model, I developed the first polynomial time solution for MFSP. The time and space complexity of my algorithm is $O(n^3 + n^2m)$ and $O(n^2)$ respectively. I have also studied the problem for heterogenous networks. I have developed another polynomial time solution for DGs which is a more general setup. For this model problem space complexity remains same as $O(n^2)$ but runtime complexity increases to $O(n^5m)$.

Finally I have studied the problem for a more realistic setup by taking collisions into consideration. I have developed a new system to solve this problem by defining rounds for the delay assignment. I have developed two practical heuristics for this setup and demonstrated their efficiency with simulations.

Local algorithms are desirable for WANET applications. In this study, I developed optimal solutions to predefined local problems using geometrical structures of these problems. I believe such local algorithms with proven efficiency will play an important role in the development of future protocols for WANETs.

# REFERENCES

[1] C. Adjih, P. Jacquet, and L. Viennot. Computing connected dominated sets with multipoint relays. *Ad Hoc and Sersor Networks*, 1:27–39, May 2005.

[2] C. Ambuhl, T. Erlebach, M. Mihalak, and M. Nunkesser. Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In *Proc. of the 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, Barcelona, Spain, August 2006.

[3] J. Aspnes, D. Goldenber, and R. Yang. On the computational complexity of sensor network localization. In *Algorithmic Aspects of Wireless Sensor Networks: First International Workshop*, Turku, Finland, July 2004.

[4] B. Baker. Approximation algorithms for np-complete problems on planar graphs. *Journal of the ACM (JACM)*, 41(1):153–180, 1994.

[5] R. Bar-Yehuda and S. Even. A Linear-Time Approximation Algorithm for the Weighted Vertex Cover Problem. *J. Algorithms*, 2(2):198–203, 1981.

[6] M. Baysan, K. Sarac, S. Bereg, and R. Chandrasekaran. A polynomial time solution to minimum forwarding set problem in wireless networks under unit disk coverage model. Technical Report UTDCS-09-07, Univ. Texas Dallas, February 2007.

[7] H. Breu and D. Kirkpatrick. Unit disk graph recognition is np-hard. *Computational Geometry*, 9(1-2):3–24, 1998.

[8] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete & Computational Geometry*, 14(4):463–479, 1995.

[9] J. Bruck, J. Gao, and A. Jiang. Localization and routing in sensor networks by local angle information. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, Urbana-Champaign, IL, USA, 2005.

[10] G. Calinescu. Computing 2-hop neighborhoods in ad hoc wireless networks. In *2nd Int. Conference on Ad-Hoc, Mobile, and Wireless Networks*, Montreal, Canada, October 2003.

[11] G. Calinescu, I. Mandoui, P.-J. Wan, and A. Zelikovsky. Selecting forwarding neighbors in wireless ad hoc networks. *Mobile Networks and Applications*, 9:101–111, 2004.

[12] X. Cheng, X. Huang, D. Li, W. Wu, and D. Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 42(4):202–208, 2003.

[13] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operation Research*, 4(3):233–235, 1979.

[14] B. Clark, C. Colbourn, and D. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

[15] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit Disk Graphs. *Discrete Mathematics*, 86(1-3):165–177, 1991.

[16] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). Internet Engineering Task Force (IETF) - RFC 3561, July 2003.

[17] L. Feeney. An energy-cosumption model for performance analysis of routing protocols for mobile ad hoc networks. *ACM Journal of Mobile Netowrks and Applications*, 3(6):239–249, 2001.

[18] P. Hliněňý and J. Kratochvíl. Representing graphs by disks and balls (a survey of recognition-complexity results). *Discrete Mathematics*, 229(1-3):101–124, 2001.

[19] D. Hochbaum. Approximation Algorithms for the Set Covering and Vertex Cover Problems. *SIAM J. Comput.*, 11(3):555–556, 1982.

[20] D. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, (1):130–136, 1985.

[21] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *Journal of the ACM (JACM)*, 32(1):130–136, 1985.

[22] H. Hunt, M. Marathe, V. Radhakrishnan, S. Ravi, D. Rosenkrantz, and R. Stearns. Nc-approximation schemes for np-and pspace-hard problems for geometric graphs. *Journal of Algorithms*, 26(2):238–274, 1998.

[23] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic. *Resource Management in Wireless Networking*, chapter Energy-efficient broadcasting in wireless mobile ad hoc networks. Kluwer Academic Press, 2004.

[24] D. Johnson, D. Maltz, and J. Broch. *Ad Hoc Networking*, chapter Chapter 5: DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks, pages 139–172. Addison-Wesley, 2001.

[25] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[26] X. Li. *Ad Hoc Wireless Networking*, chapter Applications of Computational Geomety in Wireless Ad Hoc Networks. Kluwer, 2003.

[27] X.-Y. Li. Algorithmic, geometric and graphs issues in wireless networks. *Wireless Communications and Mobile Computing*, 3(2):119–140, March 2003.

[28] L. Lovasz. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13(4):383–390, 1975.

[29] M. Marathe, H. Breu, H. Hunt, S. Ravi, and D. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59–68, 1995.

[30] J. Matousek, R. Seidel, and E. Welzl. How to net a lot with little: Small e-nets for disks and halfspaces. In *Proceedings of the sixth annual symposium on Computational geometry*, pages 16–22, Berkeley, CA, USA, 1990.

[31] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, Seattle, WA, USA, 1999.

[32] V. Paschos. A survey of approximately optimal solutions to some covering and packing problems. *ACM Computing Surveys (CSUR)*, 29(2):171–209, 1997.

[33] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.

[34] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, USA, February 1999.

[35] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *Proceedings of 35th Hawaii International Conference on System Sciences*, Big Island, HI, USA, January 2002.

[36] M. Thoppian and R. Prakash. A distributed protocol for dynamic address assignment in mobile ad hoc networks. *IEEE Tran. on Mobile Computing*, 5(1):4–19, January 2006.

[37] J. Wu, W. Lou, and F. Dai. Extended multipoint relays to determine connected dominating sets in MANETs. *IEEE Transactions on Computers*, 55(3):334–347, March 2006.

# VITA

Mehmet Baysan was born and raised in Izmir, Turkey. He graduated from Istanbul Fatih Science Lycee in 1999. He received his B.S. degree in Computer Science from Bilkent University, Ankara, Turkey in 2003. He received his M.S. degree in Computer Science from University of Texas at Dallas in 2005.