Polynomial time solution to minimum forwarding set problem in wireless networks under disk coverage model

Mehmet Baysan\textsuperscript{a}, Kamil Sarac\textsuperscript{b,⇑}, R. Chandrasekaran\textsuperscript{b}

\textsuperscript{a} Neuro-Oncology Branch, National Cancer Institute, National Institute of Neurological Disease and Stroke, National Institutes of Health, Bethesda, MD, United States
\textsuperscript{b} Dept. of Computer Sci., Univ. Texas at Dallas, Richardson, TX, United States

Abstract

In this paper, we consider a practical problem, called Minimum Forwarding Set Problem (MFSP), that emerges within the context of implementing (energy efficient) communication protocols for wireless ad hoc or sensor networks. For a given node \(v\), MFSP asks for a minimum cardinality subset of 1-hop neighbors of \(v\) to cover \(v\)'s 2-hop neighbors. MFSP problem is also known as multi-point relay (MPR) problem. It is shown to be an NP-complete problem for its general case that does not consider the coverage characteristics of wireless transmissions. In this paper, we present two polynomial time algorithms to solve the MFSP problem under disk coverage model for wireless transmissions. In our earlier work, we presented a polynomial time algorithm for this problem under unit disk coverage model. In the current work, we present several observations on the geometric characteristics of wireless transmissions under disk coverage model and build two alternative dynamic programming based solutions with different run time and space complexities to the problem. Disk coverage model is a more general model because it allows nodes to use arbitrary power levels for transmissions. As a result, the presented algorithms provide a more practical solution that can be used as a building block for energy efficient communication protocols designed for wireless ad hoc and sensor networks.

1. Introduction

Energy and wireless bandwidth are two scarce resources that need to be used carefully in wireless ad hoc and sensor networks (WANETs). WANET nodes typically operate on battery power and share limited capacity wireless transmission medium to communicate with each other. Various approaches have been proposed to improve the utilization of these resources in WANETs. Localized algorithms for energy efficient communication form an important class of practical approaches for the effective and efficient use of these resources in WANETs \cite{1,2}. Localized algorithms are distributed algorithms where WANET nodes use a limited local topology information, typically 1-hop or 2-hop neighborhood information, to perform their communication actions. This is desirable for energy and bandwidth efficiency as global topology information is often expensive to collect and maintain up to date. Localized algorithms are used in building energy efficient topology structures or energy efficient network wide broadcast algorithms in WANETs \cite{1–4}. As an example, popular WANET routing protocols, including OLSR, AODV, and DSR, use broadcast to discover and maintain routes between the nodes in a WANET. A naive implementation of the broadcast operation where each 1-hop neighbor of a transmitting node involves in relaying of a broadcast message (i.e., network wide flooding) may cause a high level of energy and bandwidth consumption in WANETs \cite{5}. In this context, it becomes important to carefully select a subset of 1-hop neighbors as relay nodes \cite{6}.
1.1. Problem definition

One important primitive that is used in various localized energy efficient communication algorithms is the selection of a minimum number of 1-hop neighbors as relay nodes in reaching 2-hop neighbors. This primitive can help in building energy efficient topology structures as well as energy efficient local broadcast solutions in covering 2-hop neighbors [7,8]. This problem is referred to as Multi-point Relay (MPR) selection or Minimum Forwarding Set Problem (MFSP) [6,9] and is formally defined as follows:

**Definition 1** (Minimum Forwarding Set Problem (MFSP)). Consider a network 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 \( N(v) \cap N_2(v) = \emptyset \). 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 \( \forall v \in N_2(v), \exists y \in S(x \in N(v)) \). 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 an optimal solution, the assignment of a node \( b \in N_2(v) \) to a node \( s \in S \) requires that \( b \in N(s) \). In other words, in the context of the wireless transmission, \( b \) should be within the coverage range of \( s \). Also note that, in certain cases, multiple different optimal solutions may exist.

1.2. Existing solutions

The MFSP problem is shown to be NP-complete for arbitrary graphs with a reduction from the Set Cover problem [6]. The heuristic proposed in [6] is an application of the arbitrary graphs with a reduction from the Set Cover problem [10] and gives an approximation ratio of 2. The heuristic proposed in [6] is an application of the arbitrary graphs with a reduction from the Set Cover problem [10] and gives an approximation ratio of 2. The heuristic proposed in [6] is an application of the arbitrary graphs with a reduction from the Set Cover problem [10] and gives an approximation ratio of 2.

In our recent work [15], we presented the first polynomial time exact algorithm for the MFSP problem under the UDG model. For the UDG version of the problem, we first introduced two geometric properties named as Two-Set Property and Non-Interleaving Property that hold true for all instances of the problem under UDG model. We then presented a dynamic programming algorithm to build an optimal solution and proved its correctness. The algorithm for the UDG version has \( O(n^4m) \) time complexity where \( m = |N(v)| \) and \( n = |N_2(v)| \) for a broadcasting node \( v \). Finally, Lev Tov studied the disk graph version of the problem in the computational geometry context and developed an algorithm with \( O(n^4m) \) run-time and \( O(n^2m) \) space complexity [16].

1.3. Our contributions

In this paper, we present two polynomial time exact algorithms with different run time and space complexities to solve the MFSP problem under the disk coverage (DC) model where the broadcast coverage area of different nodes are represented by disks of different radii. The first solution has a run-time complexity of \( O(n^3m) \) and space complexity of \( O(n^2m) \) and the second solution has a run-time complexity of \( O(n^3m) \) and space complexity of \( O(n^2m) \), where \( m = |N(v)| \) and \( n = |N_2(v)| \) for a broadcasting node \( v \). Our solutions are based on the Non-Interleaving Property (see Section 3.4) that is valid for any instance of MFSP problem under DG model. After proving this property, we use dynamic programming algorithms to construct a minimum forwarding set as an optimal solution to the problem. Given that our algorithms build on important geometric properties, a significant part of the text is devoted to clearly establish the accuracy of these properties.

The current paper relaxes the UDG assumption and provides an exact solution to the problem under a more general coverage model. This is achieved with an increase in the time complexity from \( O(n^3 + n^3m) \) for the UDG model to \( O(n^3m) \) in our first algorithm presented in Section 4 for the DG model. Our second algorithm presented in Section 5 reduces this bound to \( O(n^2m) \) by an increased space requirement. Even though the DG model is a more general model than the UDG model in representing wireless coverage, it may not be applicable for all real world wireless network environments. But it is necessary to make certain assumptions to design efficient algorithms as it is difficult to model the random nature of wireless communication. In addition, the presented algorithm can be quite instrument for evaluating the performance of more practical heuristic models in simulation studies.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 introduces several definitions and properties that we use in the construction of our solution. Sections 4 and 5 present the algorithms. Finally, Section 6 concludes the paper.

2. Related work

The MFSP problem emerged within the context of energy efficient communication using local information (e.g., topology construction and network wide broadcast)
in WANETs. In this section, we present a brief summary of the related problems and refer our readers to [12,17,18,2,19] for more information on the existing literature on energy efficient communication in WANETs.

The general case of the MFSP problem is an instance of the well-known NP-complete Set Cover problem [6]. Set Cover problem has been extensively studied in the literature and early approximation algorithms have been proposed for both unweighted version by Johnson [20] and by Lovasz [21], and for weighted version by Chvatal [10]. These algorithms give an approximation ratio of $1 + \ln(n)$ where $\ln$ is the cardinality of the maximum cardinality subset $(\max_{S \subseteq \{1, \ldots, n\}} |S|)$. In [22], Hochbaum presents an algorithm for the weighted version with an approximation ratio of $\sqrt{\ln(n^k)}$ where $k$ represents the maximum number of subsets covering an element. The running time of this algorithm is $O(n^k)$. In [23], Bar-Yehuda and Even present an algorithm with a similar approximation ratio but an improved running time of $O(n^{5.5})$. We refer readers to [24] for other approximation algorithms on the Set Cover problem.

The MFSP problem becomes a geometrical problem when we use disks to model the coverage area of wireless transmissions. Disk graphs (DGs) are neither perfect nor planar graphs [14]. Thus, efficient algorithms proposed for planar and perfect graphs cannot be applied to DGs. Some general results for DGs can be found in [25]. Unit disk graphs (UDGs) are an extensively studied subset of DGs. Since the amount of information related to UDGs is much more than DGs, we provide some of these related results combined with some results for DGs as a part of related work.

MFSP problem under the disk coverage assumption resembles to the well-known Minimum Dominating Set (MDS) problem. MDS problem for UDGs has been studied extensively. The problem is shown to be NP-complete for UDGs [14]. In [26], Marathe et al. present a linear time approximation algorithm with a constant-factor performance guarantee of 5. In [27], a polynomial-time approximation scheme (PTAS) with $(k + 1)/k^2$ guarantee is given for a constant $k$ in $n^{0.5}$. Minimum Connected Dominating Set (MCDS) problem is a different version of the problem in which the dominating set should be connected. In [28], Thai et al. provide two constant-factor approximation algorithms for MCDS for DGs given that the ratio of maximum radius to minimum radius is constant. In [29], Cheng et al. presented a PTAS for MCDS problem for UDGs. In [30], Ambuhl et al. presented constant-factor approximation algorithms for the weighted versions of MDS and MCDS problems for UDGs. 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 (unit disks can be placed arbitrarily). This problem is examined in [31] and $O((l + \sqrt{2})^2 + 2n^{2/3})$ time approximation algorithm is given with a performance guarantee of $(1 + l/2)^2$. 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 neighboring nodes.

Another related problem to MFSP problem is the well-known Disk Cover (DC) problem that tries to find a minimal size set of unit disks (from a given set of unit disks) to cover a given set of points on a plane [31]. In [32], authors present an algorithm with an approximation ratio of $O(1)$ and running time of $O(c^2\log n \log(n/c))$ where $c$ represents the size of the optimal solution. MFSP problem is a special instance of the DC problem where disks are selected from a given set of 1-hop nodes.

Another related work in the context of wireless broadcast is Localized Broadcast Incremental Power Protocol (LBIP) [3]. LBIP, nodes are assumed to have variable transmission power and the goal is to cover 2-hop neighbors with minimum energy. LBIP involves selection of forwarding nodes as well as determining transmission power levels for such nodes to achieve minimum energy usage. In our current work, we assume fixed transmission power (i.e., unit disk coverage) and our goal is to choose a minimum number of 1-hop neighbors to cover all 2-hop neighbors.

The most related work to our study in this paper is the previous work by Calinescu et al. [9], by Lev Tov [16], and by us [15]. In their work [9], Calinescu et al. propose approximation algorithms to solve the MFSP problem (see Section 1.2). In her dissertation work [16], Lev Tov studies MFSP for DGs and provides an exact solution which has run-time complexity $O(n^m m)$ and space complexity $O(n^m m)$. The work presented in this paper provides two alternative solutions. One improves the run-time complexity of Lev Tov’s algorithm from $O(n^m m)$ to $O(n^m)$ by using same space complexity of $O(n^m)$. The other one has a lower space complexity $O(n^2)$ with a worse run-time performance $O(n^3)$. Finally, in our recent work [15], we presented the first polynomial time exact algorithm to solve the MFSP problem under UDG model. The current work presented in this paper is significantly different from the solution of the problem for UDG model where a node $s \in N(v)$ could be essential for at most two MCIs (see the next section for the definition of an MCI) whereas that critical property does not hold in DG model, necessitating the development of a totally new algorithm.

### 3. 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 introduce a theorem that we use in the construction of our algorithms.

#### 3.1. The practical setup of the problem

Most studies use a unit disk or a sphere to represent the shape of the effective coverage area of wireless transmissions [13]. This assumption, though may not always hold in practice, helps in gaining more insight to the problem within the practical context of wireless transmissions. In this paper, we use arbitrary disks to represent the shape of the coverage area for wireless transmission. Compared to
of the presentation. Recently, Calinescu [34] proposed methods to calculate 2-hop ultrasound receivers or directional antennas. Within an energy consumption model [33] that is representative for the environment. The angle information between neighboring nodes can be measured by using multiple ultrasound receivers or directional antennas. Recently, Calinescu [34] 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.

3.2. Definitions

In this section, we present several definitions that we use in the rest of the paper. Some of these definitions also appeared in our previous work on the UDG version of the problem [15] but are included here for the completeness of the presentation.

Let $v, N(v)$, and $N_2(v)$ represent a node, its 1-hop, and 2-hop neighbors respectively. For the simplicity of notation, let $v, N(v)$, and $N_2(v)$ also represent the locations of these nodes in a 2-dimensional space. Let $D_v$ represent the coverage area of the node $v$. $D_v$ is a disk with a radius $r_v$. Similarly, $D_b$ is a disk that represents the coverage area of a node $b \in N(v)$. Let $D_{n}(v)$ represent the area covered by 1-hop neighbors of $v$ outside of $D_b$, that is, $D_n = \left( \bigcup_{b \in N(v)} D_b \right) \setminus D_v$. By these definitions, we have $N(v) \subseteq D_v$ and $N_2(v) \subseteq D_v$. Based on this setup, we present several definitions as below.

**Definition 2** (Radial order). Radial order is the ordering of a set of points in $D_v$ (or the nodes at those points) by using the angle that they make with the origin (point) $v$. Radial order is a cyclic order. If two or more points 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 Fig. 1a where $N(v) = \{s, t\}$ and $N_2(v) = \{a, b, c, d, e\}$. Starting from the exact south position, the nodes in $N_2(v)$ form a radial order as $\{e < d < c < b < a\}$. The theorems introduced below and the algorithms presented later on use the radial ordering of the nodes in $N_2(v)$ in finding an optimal solution. As we discussed in Section 3.1, 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** (Radially Continuous Neighbor (RCN) interval). One or more points in the area $D_v$ 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.

As an example, in Fig. 1a, $(a > b > c)$ and $(e > a > b)$ form RCN intervals 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 4** (Radially Continuous Coverage Area (RCCA)). Consider a set $S \subseteq N(v)$. For a node $s \in S$, RCCA of $s$ is a continuous subarea in $D_v$. RCCA(s) $\subseteq D_v$, such that $s$ is the only node in $S$ that can cover all the points in RCCA(s). A node $s \in S$ may have zero or more RCCAs.

**Definition 5** (Maximum Coverage Interval (MCI)). An MCI of a node $s \in N(v)$ is an RCN interval $[a_i, \ldots, a] \in N_2(v)$ that is completely covered by $s$ such that $s$ cannot cover neither of $a_i$ and $a_{i+1}$. Note that $s$ can have multiple MCIs in $N_2(v)$.

**Definition 6** (Essential coverage). Consider a node $s \in S \subseteq N(v)$ that covers a point $a_i \in D_v$. $s$ is said to be essential to cover $a_i$ if no other node $t \in S$ covers $a_i$. If $s$ is essential for a node (at a point) $a_i$ in an MCI that it covers, then $s$ is essential for this MCI. Similarly, a node $s \in S \subseteq N(v)$ is said to be essential to cover an RCCA(s).

**Definition 7** (Dominance). Let $\{s, t\} \in N(v)$ and $[a_i, a_j] \in N_2(v)$ such that $[a_i, a_j] \notin N(s)$, $[a_i] \notin N(t)$, $[a_j] \notin N(t)$, and $a_i$ and $a_j$ 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 $s \geq t$.

As an example, in Fig. 1b, $N(v) = \{s, t\}$, $N_2(v) = \{a, b, c\}$, and $a$ and $b$ are in radial order as $(a > b > c)$. In this example, $s$ dominates $t$ as $N(s) = \{a, c\}$ and $b \notin N(s)$ whereas $N(t) = \{b\}$ and $\{a, c\} \notin N(t)$.

**Definition 8** (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 $a_j \in N_2(v)$, $R_{ij} = 1$ if $a_j \in N(b_i)$ and $R_{ij} = 0$ otherwise.

**Definition 9** (Interleaving coverage). Consider two nodes $\{s, t\} \in N(v)$ in an instance of MFPS 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 $(a > b > c > d)$. The coverage of this form between the nodes $s$ and $t$ is called an interleaving coverage.
As an example, in Fig. 1c, \( N(v) = \{s, t\}, N_2(v) = \{a, b, c, d\} \), and \( a, b, c, \) and \( d \) are in radially ordered as \((a > b > c > d)\). Assume, in this example, that \( N(s) = \{a, c\} \) and \( N(t) = \{b, d\} \), i.e., the solid lines between the nodes in the figure represent coverage of \( s \) and \( t \) on \( a, b, c, \) and \( d \) (note that \( \{b, d\} \not\subset N(s) \) and \( \{a, c\} \not\subset N(t) \)). In this case, \( s \) and \( t \) have an interleaving coverage. Fig. 1d presents the connectivity matrix corresponding to this coverage scenario. In Theorem 1 below, we state and prove that such an interleaving coverage is not possible among any two nodes in \( N(v) \).

### 3.3. Intersection characteristics of disks

Following the notation from the previous subsection, let \( v, s, \) and \( t \) be three points on a 2-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 periphery circle enclosing the coverage area \( D_i \). Finally, assume that \( D_{ij} = D_i \setminus D_j \neq \emptyset \) for any \( i \in N(v) \) as otherwise node \( i \) would not be used in reaching nodes in \( N_2(v) \). In the rest of this section, we study the relation between the coverage of \( D_s \) and \( D_t \) outside the coverage area \( D_v \).

Consider the periphery circles \( C_s \) and \( C_t \) that enclose \( D_s \) and \( D_t \) respectively. It is a well-known fact that two circles can intersect at most twice. If \( C_s \) and \( C_t \) have no intersections, then it is either \( D_s \cap D_t = \emptyset \) or \( D_s \subset D_t \) (or \( D_t \subset D_s \)) as shown in Fig. 2a or b respectively. If \( C_s \) and \( C_t \) have one intersection at a point \( p \), then it is either \( D_s \cap D_t = \{p\} \) (i.e., \( D_s \) and \( D_t \) intersect at a single point \( p \)) or \( D_s \subset D_t \) (or \( D_t \subset D_s \)) as shown in Fig. 2c or d respectively. Finally, if \( C_s \) and \( C_t \) have two intersections, we have three different cases that are important for us: (1) both intersection points \( p \) and \( q \) are within \( D_v \) as in Fig. 2e and f; (2) one of the intersection points, say \( p \), is out of \( D_v \) and the other one, say \( q \), is within \( D_v \) as in Fig. 2g; and (3) both intersection points \( p \) and \( q \) are outside of \( D_v \) (see Fig. 2h as an example). We analyze each case below.

**Lemma 1.** Let \( S = \{s, t\} \subseteq N(v) \) such that \( D_{qv} = D_q \setminus D_v \neq \emptyset \) and \( D_{uv} = D_u \setminus D_v \neq \emptyset \) in an instance of MFSP problem at \( v \). If \( C_s \) and \( C_t \) intersect twice in \( D_v \), then \( D_{qv} \cap D_{uv} = \emptyset \) or \( D_{qv} \subset D_{uv} \) (or \( D_{uv} \subset D_{qv} \)).

**Proof of Lemma 1.** Let \( m \) and \( n \) be the intersection points of \( C_v \) and \( C_s \); and \( m \) and \( n \) be the intersection points of \( C_v \) and \( C_t \) as shown in Fig. 2e. Assume the contrary that \( D_{slv} = D_{slv} \cup D_{slv} \) and \( D_{slv} = D_{slv} \cup D_{slv} \) where \( D_{slv} = D_{slv} \setminus D_{slv} \), \( D_{slv} = D_{slv} \setminus D_{slv} \), and \( D_{slv} = D_{slv} \cap D_{slv} \), that is, \( D_{slv} \) and \( D_{slv} \) have some overlapping coverage area in \( D_v \) without \( D_{slv} \) (or \( D_{slv} \) completely

---

**Fig. 1.** Properties of 1- and 2-hop neighbors of node \( v \).

Connectivity Matrix

\[
\begin{array}{cccc}
    a & b & c & d \\
    s & 1 & 0 & 1 \\
    t & 0 & 1 & 0 \\
\end{array}
\]
Corollary 1. Let $S = \{s, t\} \subseteq N(v)$ such that $D_{sv} = D_s \setminus D_v \neq \emptyset$ and $D_{tv} = D_t \setminus D_v \neq \emptyset$ in an instance of MFSP problem at $v$. If $C_{sv}$ has no intersection with $C_{tv}$, then each of $s$ and $t$ has at most one single RCCA w.r.t. $S$ as follows:

- If $D_{sv} \cap D_{tv} = \emptyset$, then $D_{sv}$ and $D_{tv}$ are RCCAs of $s$ and $t$ respectively w.r.t. $S$.
- If $D_{sv} \subseteq D_{tv}$, then $D_{sv}$ is an RCCA of $s$ w.r.t. $S$.
- If $D_{sv} \supseteq D_{tv}$, then $D_{tv}$ is an RCCA of $t$ w.r.t. $S$.

Lemma 2. Let $S = \{s, t\} \subseteq N(v)$ such that $D_{sv} = D_s \setminus D_v \neq \emptyset$ and $D_{tv} = D_t \setminus D_v \neq \emptyset$ in an instance of MFSP problem at $v$. If $C_s$ and $C_t$ intersect twice with one of the intersections in $D_v$ and the other in $D_v$, then each of $s$ and $t$ is essential to cover one single RCCA in $D_{sv}$ or $D_{tv}$ respectively w.r.t. $S$.

Proof of Lemma 2. When the periphery circles of two disks, $C_s$ and $C_t$, intersect twice, this results in three coverage areas for the disks $D_s$ and $D_t$, as (1) $D_s \setminus D_v \neq \emptyset$, (2) $D_t \setminus D_v \neq \emptyset$, and (3) $D_s \cap D_t = \emptyset$. We consider the parts of these coverage areas in $D_v$ denoted as $D_{sv} = (D_s \setminus D_v) \cap D_v \neq \emptyset$, $D_{tv} = (D_t \setminus D_v) \cap D_v \neq \emptyset$, and $D_{st} = (D_s \cap D_t) \setminus D_v \neq \emptyset$, respectively (see Fig. 2g). Let $p$ be the intersection point of $C_s$ and $C_t$. Consider a ray $l$ that originates at $v$ and crosses $p$ in Fig. 2g. The ray $l$ divides $(D_{sv} \cup D_{tv}) \cap D_v$ into two areas such that the radial order of the points at both sides of $l$ are disjoint from each other. In this case, $s$ is essential to cover one RCCA that includes $D_{sv}$ and part of $D_{tv}$ below ray $l$ and $t$ is essential to cover one RCCA that includes $D_{tv}$ and part of $D_{sv}$ above ray $l$ as in Fig. 2g. □

In the following, we consider the case for two intersections between $C_s$ and $C_t$ in $D_v$. First we present several geometrical facts that we use in our analysis.
Lemma 3. Let \( D_v \) be a disk with radius \( r_v \), periphery circle \( C_v \) and origin \( v \). Similarly, \( D_s \) is a disk with radius \( r_s \), periphery circle \( C_s \) and origin \( s \). If \( |v| < \min(r_v, r_s) \), then any ray \( l \) that originates at \( v \) intersects \( C_s \) at one single point \( p \).

Proof of Lemma 3. \( |v| < \min(r_v, r_s) \) implies that \( v \in D_s \) and any ray originating at any point in a disk \( D_v \) intersects the periphery circle \( C_s \) at one single point \( p \). If \( D_s \subset D_v \), then \( p \) is in \( D_s \). If \( D_v \setminus D_s = \emptyset \), then \( p \) is either in \( D_v \) or outside of \( D_v \) (i.e., \( D_v \)).

Lemma 4. Let \( D_v \) be a disk with radius \( r_v \), periphery circle \( C_v \) and origin \( v \). Similarly, \( D_s \) is a disk with radius \( r_s \), periphery circle \( C_s \) and origin \( s \). Let \( D_s \setminus D_v = \emptyset \). If \( r_v > |v| > r_s \), then a ray \( l \) that originates at \( v \) may intersect \( C_s \) zero, one, or two times with at most one of these intersections being outside of \( D_v \).

Proof of Lemma 4. \( r_v > |v| > r_s \) implies that \( s \in D_v \) and \( v \notin D_s \). Similarly, \( D_s \setminus D_v = \emptyset \) implies that \( C_v \) and \( C_s \) intersect at two points, say \( m \) and \( n \). Assume that \( v \) is at the origin of a 2-dimensional space and \( s \) is to exact east of \( v \). Consider a radial sweep operation where starting from the exact south position, we use a ray \( l \) of origin \( v \) to sweep the area in counterclockwise direction. When \( l \) is toward the exact south, it cannot intersect \( C_s \) due to the fact that \( r_s < r_v \). While we rotate \( l \) in counterclockwise direction, at a radial location \( x_p \) (the angle \( \angle pvy \) in Fig. 3a), I will intersect \( C_s \) at a single point, say \( p \) (i.e., \( l \) is tangent to \( C_s \) at \( p \)). For ease of discussion, we refer to this instance of \( l \) as \( l_p \) (see Fig. 3a). As we continue to rotate \( l \), at another angular location \( x_q \) (the angle \( \angle qvy \) in Fig. 3a), it will again intersect \( C_s \) at another single point, say \( q \). We again refer to this instance of \( l \) as \( l_q \) (see Fig. 3a). Finally, any other ray \( l_i \) that is radially between \( l_p \) and \( l_q \) will have two intersections with \( C_s \). We claim that \( \{p, q\} \in D_s \). We prove this for \( p \). Assume the contrary that \( p \notin D_s \). This requires that \( |vp| > r_v \). Note that the triangle \( \triangle vps \) is a right triangle (\( l \) is tangent to \( C_s \) at \( p \)) with \( \angle vps \) being its right angle and \( vs \) being its hypotenuse. Our assumption \( p \notin D_s \) suggests that \( |vp| > |vs| \) which is a contradiction to the Pythagorean theorem. Hence, \( p \in D_s \). Note that a similar argument holds for \( q \). In addition, since \( s \) is to the exact east of \( v \), the chord \( pq \) is parallel to \( y \)-axis.

Next, we claim that for any ray \( l \) that originates at \( v \) and intersects \( C_s \) twice, one of these intersections is on arc \( arc_c(\theta) \) of \( C_s \) (that is, the segment of \( C_s \) starting at \( q \) and ending at \( p \) in counterclockwise direction) and the other one is on arc \( arc_c(\phi) \) of \( C_s \) (that is, the segment of \( C_s \) starting at \( p \) and ending at \( q \) in counterclockwise direction).

Note that \( D_v \) is located in a region that is radially bounded by \( l_p \) and \( l_q \) on east of \((v, q)\) (see Fig. 3b). Standing at \( v \) and looking east, the visible segment of \( C_s \) is the arc \( arc_c(\theta) \). Any ray \( l \) that originates at \( v \) and intersects \( C_s \) twice has to enter \( D_v \) at a point on \( arc_c(\phi) \), cross the chord \( qp \) and exit \( D_v \) at a point on \( arc_c(\theta) \). As a result, since \( arc_c(\phi) \) is a segment of \( arc_c(\theta) \), a ray \( l \) originating at \( v \) can intersect \( arc_c(\phi) \) at most once. □

Corollary 2. Let \( s \in N(v) \) such that \( D_v = D_s \setminus D_v \neq \emptyset \) in an instance of MFSP problem at \( v \). A ray \( l \) that originates at \( v \) and intersects \( C_s \) at most once at a point \( p \) outside \( D_v \) (i.e., in \( D_s \)). If \( v \notin D_s \), then \( p \) is the only intersection point of \( l \) with \( C_s \). If \( v \notin D_v \), then \( l \) intersects \( C_v \) at another point \( p \in D_v \) besides its intersection with \( C_s \) at \( p \in D_v \).

Now, we consider the case where \( C_s \) and \( C_v \) intersect twice in \( D_v \).

Lemma 5. Let \( S = \{s, t\} \subseteq N(v) \) such that \( D_s \setminus D_v \neq \emptyset \) and \( D_v \setminus D_s \neq \emptyset \) in an instance of MFSP problem at \( v \). If \( C_s \) and \( C_v \) intersect twice with both intersection points being in \( D_v \), then one of \( s \) or \( t \) is essential to cover one RCCA and the other one is essential to cover one or two RCCAs w.r.t. \( S \).

Before we prove Lemma 5, we present some facts about intersecting two circles \( C_s \) and \( C_v \). Consider Fig. 3c where \( C_v \) and \( C_s \) intersect twice at points \( p \) and \( q \). In this figure, \( arc_c(\phi) \) is the segment of \( C_v \) between the intersection points \( p \) and \( q \) in counterclockwise direction and \( arc_c(\theta) \) is the segment of \( C_s \) between \( q \) and \( p \) in counterclockwise direction. Similarly, \( arc_c(\psi) \) is the segment of \( C_v \) between \( p \) and \( q \) in counterclockwise direction and \( arc_c(\xi) \) is the segment of \( C_s \) between \( q \) and \( p \) both in counterclockwise direction. Note that \( arc_c(\phi) \) and \( arc_c(\psi) \) together enclose the area \( D_s \cup D_v \) and we have \( arc_c(\phi) \in D_s \) and \( arc_c(\psi) \in D_v \). Now, we present the proof of the lemma.

Fig. 3. Characteristics of intersecting lines and circles.
**Proof of Lemma 5.** Let $C_i$ and $C_j$ intersect at two points $p$ and $q$ in $D_v$. Let $l_1$ and $l_2$ be two lines originating at $v$ and crossing $p$ and $q$ respectively (see Fig. 2h as an example). From the proof of Lemma 4, we know that $l_1$ cannot be tangent to $C_i$ (or $C_j$) at $p$ (i.e., if $p$ is the only intersection point between $l_1$ and $C_i$ (or $C_j$), then $p$ has to be in $D_{C_j}$). Therefore, it is either $v \in D_l$ (or $v \notin D_l$) or $l_1$ intersects $C_i$ (or $C_j$) twice one at $p \in D_l$ and the other in another point $\bar{p} \in D_h$. Similar argument applies to $l_2$.

From the intersection of $C_i$ and $C_j$, we have either $arc_{C_i}(pq)$ and $arc_{C_j}(pq)$ enclosing $D_l \cup D_l$ or $arc_{C_i}(pq)$ and $arc_{C_j}(pq)$ enclosing $D_l \cup D_j$. Assume, without loss of generality, that $arc_{C_i}(pq)$ and $arc_{C_j}(pq)$ enclose $D_l \cup D_j$ as in Fig. 2h. From Corollary 2, any ray $l_i$ originating at $v$ intersects $C_{s\ell}$ and $C_{t\ell}$ at most once. Consider a sequence of $l_i$ between $l_2$ and $l_1$ in radial order, all the point that $l_i$ intersects with in $D_{C_x}$ are also in $D_{C_x}$ as $arc_{C_x}(pq)$ encloses $D_{C_x}$ between $l_2$ and $l_1$, i.e., $D_{C_x} \subset D_{C_x}$ between $l_2$ and $l_1$. As a result, the coverage area $D_{C_x}$ between $l_2$ and $l_1$ is an RCCA for $v \text{ w.r.t. } S$.

Now, consider the coverage areas $D_{C_x}$ and $D_{C_y}$ between $l_1$ and $l_2$ in radial order. Note that, $arc_{C_y}(pq)$ encloses $D_l \cup D_l$ in this region, i.e., $D_{C_y} \subset D_{C_y}$ between $l_1$ and $l_2$ in radial order. For the coverage area of $D_{C_x}$ between $l_1$ and $l_2$, we have the following:

- If $arc_{C_x}(pq)$ has zero intersection with $C_x$, then $D_{C_x}$ between $l_1$ and $l_2$ is one single RCCA for $v \text{ w.r.t. } S$. That is, $arc_{C_x}(pq)$ encloses the coverage area $D_{C_x}$ between $l_1$ and $l_2$ without $arc_{C_y}(pq)$ intersecting with $C_x$ in that region (as in Fig. 2h).
- If $arc_{C_y}(pq)$ has one intersection with $C_y$ at a point $\bar{p}$ between $l_1$ and $l_2$, then $D_{C_y}$ is divided into two subareas enclosed by $arc_{C_y}(pq)$ and $arc_{C_y}(pq)$ each enclosing one RCCA for $v \text{ w.r.t. } S$, i.e., $s$ is essential for two RCCAs in this case (as in Fig. 2i).
- If $arc_{C_y}(pq)$ has two intersections with $C_y$ at points $p$ and $q$ between $l_1$ and $l_2$, then $D_{C_y}$ is divided into two subareas enclosed by $arc_{C_y}(pq)$ and $arc_{C_y}(pq)$ each enclosing one RCCA for $v \text{ w.r.t. } S$, i.e., $s$ is essential for two RCCAs in this case (as in Fig. 2j).

3.4. Coverage properties

In this section, we introduce two important properties that we use in building our algorithms.

**Theorem 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 $N_f(v)$), no two nodes $(s, t) \in N(v)$ can have interleaving coverage.

**Proof of Theorem 1.** Note that interleaving is considered between any two nodes $(s, t) \in N(v)$. If $C_{s\ell}$ and $C_{t\ell}$ intersect zero times, by Corollary 1, either $s$ and $t$ both have their disjoint coverage areas $C_{s\ell}$ and $C_{t\ell}$ or $D_{s\ell} \subset D_{t\ell}$ (or $D_{t\ell} \subset D_{s\ell}$). If $C_{s\ell}$ and $C_{t\ell}$ intersect once, by Lemma 2, $s$ (and $t$) covers one single RCCA. When this RCCA of $s$ (or $t$) includes some node $a_i \in N_f(v)$, then $s$ (or $t$) has one single MCI (including such node $a_i$). Finally, when $C_{s\ell}$ and $C_{t\ell}$ intersect twice, by Lemma 5, $t$ (or $s$) is essential to cover one single MCI and $s$ (or $t$) is essential to cover one or two MCLs. Note that, as shown in Fig. 2b-1, the two lines $l_1$ and $l_2$ (or $p \in D_{l_2}$) similar argument applies to $l_2$.

Note that Theorem 1 states that domination is an acyclic property. Consider an instance of MFSP problem at $v$ where $(b_x, b_y, b_z) \in N(v)$. We show that if $b_x \Delta b_y$ and $b_x \Delta b_z$ then either $b_x \Delta b_z$ or $(N(b_x) \cap N(v)) \subset (N(b_y) \cap N(v))$. By definition, $b_x \Delta b_y$ requires that there exist three nodes $(a_x, a_y, a_z) \in N(v)$ with a radial order $(a_x < a_y < a_z)$ such that $a_x$ and $a_z$ are covered by $b_x$ but not by $b_y$ and $a_y$ is covered by $b_y$ not by $b_x$. When we consider the radial order among the nodes $(a_x, a_y, a_z, a_0, a_1, a_2, a_3)$, there are four possible outcomes for the first and the last node pairs as $(1) (a_x, a_0)$, $(2) (a_x, a_0)$, $(3) (a_x, a_0)$, and $(4) (a_x, a_0)$. We now consider the overall coverage scenarios for each of these cases and show that $b_x \Delta b_y$ cannot occur in none of these cases.

Case 1 $(a_x, a_y, a_z)$: Consider the scenario in Fig. 4a that represents a segment of a connectivity matrix corresponding to this case. In this case, $b_x$ covers $a_0$ and $a_0$ as other before $b_y$ and $b_z$ have interleaving coverage contradicting with Theorem 1. In addition, $b_x$ should cover any node $a_{\text{before}} < a_0$ or $a_{\text{after}} > a_0$ that $b_x$ covers. This is because such a node $a_{\text{before}}$ or $a_{\text{after}}$ covered by $b_x$ has to be covered by $b_y$ and $b_z$ do not interleave. In this case, any node $a_{\text{before}}$ of $a_{\text{after}}$ covered by $b_x$, has to be covered by $b_y$ (so that $b_x$ and $b_y$ do not interleave). Finally, if $b_x$ covers a node $a_0$ where $(a_x < a_0 < a_z)$ such that $a_0$ is not covered by $b_y$, then $b_y \Delta b_y$ on $(a_x, a_y, a_z)$. Otherwise, $(N(b_x) \cap N(v)) \subset (N(b_y) \cap N(v))$.

Case 2 $(a_x, a_y, a_z)$: Consider the scenario in Fig. 4c that represents a segment of a connectivity matrix corresponding to this case. In this case, $b_x$ should not cover $a_0$ and $a_0$ as other otherwise $b_x$ and $b_y$ would interleave. Similarly, $b_y$ should not cover $a_0$ as otherwise $b_y$ and $b_z$ would interleave. In this scenario, $b_x$ should cover all nodes $a_{\text{before}} < a_0$ or $a_{\text{after}} > a_0$ that $b_x$ covers. This is because any $a_{\text{before}}$ or $a_{\text{after}}$ covered by $b_x$ has to be covered by $b_y$ and $b_z$ do not interleave. Finally, if $b_x$ covers a node $a_0$ where $(a_x < a_0 < a_z)$ such that $a_0$ is
interleave. In addition, 

Cases for domination relations among \( b_1, b_2 \) and \( b_3 \).

**Fig. 4.** Cases for domination relations among \( b_1, b_2 \) and \( b_3 \).

not covered by \( b_2 \), then \( b_2 \Delta b_3 \) on \( (a_v, a_p, a_g) \). Otherwise \((N(b_1) \cap N_2(v)) \subset (N(b_2) \cap N_2(v)) \).

**Case 3** \((a_v, ..., a_0)\): Consider the scenario in Fig. 4b that represents a segment of a connectivity matrix corresponding to this case. In this case, \( b_2 \) covers \( a_v \) as otherwise \( b_1 \) and \( b_2 \) would interleave. In addition, \( b_2 \) should cover any node \( a_{\text{before}} < a_0 \) or \( a_{\text{after}} > a_0 \) that \( b_1 \) covers. This is because such a node either \( a_{\text{before}} \) or \( a_{\text{after}} \) covered by \( b_1 \) has to be covered by \( b_2 \) (so that \( b_2 \) and \( b_3 \) do not interleave). In this case, any node \( a_{\text{before}} \) or \( a_{\text{after}} \) covered by \( b_2 \) has to be covered by \( b_3 \) (so that \( b_2 \) and \( b_3 \) do not interleave). Finally, if \( b_2 \) covers a node \( a_v \) where \((a_v < a_p < a_g) \) such that \( a_v \) is not covered by \( b_1 \), then \( b_2 \Delta b_3 \) on \( (a_v, a_p, a_g) \). Otherwise \((N(b_1) \cap N_2(v)) \subset (N(b_2) \cap N_2(v)) \).

**Case 4** \((a_v, ..., a_d)\): Consider the scenario in Fig. 4d that represents a segment of a connectivity matrix corresponding to this case. In this case, \( b_2 \) covers \( a_v \) as otherwise \( b_1 \) and \( b_2 \) would interleave. In this case, \( b_2 \) should not cover \( a_v \) as otherwise \( b_2 \) and \( b_3 \) would interleave. In this case, \( b_2 \) should cover all nodes \( a_{\text{before}} < a_0 \) and \( a_{\text{after}} > a_0 \) that \( b_1 \) covers. This is because any \( a_{\text{before}} \) or \( a_{\text{after}} \) covered by \( b_1 \) has to be covered by \( b_2 \) (so that \( b_2 \) and \( b_3 \) do not interleave) and any node \( a_{\text{before}} \) or \( a_{\text{after}} \) covered by \( b_2 \) should be covered by \( b_3 \) (so that \( b_2 \) and \( b_3 \) do not interleave). Finally, if \( b_2 \) covers a node \( a_v \) \((a_v < a_p < a_g) \) such that \( a_v \) is not covered by \( b_2 \), then \( b_2 \Delta b_3 \) on \( (a_v, a_p, a_g) \). Otherwise \((N(b_2) \cap N_2(v)) \subset (N(b_3) \cap N_2(v)) \).

We now show that domination is an acyclic relation among \( m \) such nodes where \( m = |N(v)| \). Assume that there is a cyclic domination relation among nodes \( b_1, b_2, b_3, ..., b_m \) in \( N(v) \) as \( (b_1 \Delta b_2, b_2 \Delta b_3, b_3 \Delta b_4, ..., b_m \Delta b_1) \). From the above discussion, we know that there are two cases for the relation between \( b_m \) and \( b_2 \), i.e., either \( b_m \Delta b_2 \) or \((N(b_2) \cap N_2(v)) \subset (N(b_m) \cap N_2(v))\). Now, consider the relation between \( b_2 \) and \( b_3 \). From the relation between \( b_m \) and \( b_2 \), if we have \( b_m \Delta b_2 \), it is either \( b_m \Delta b_3 \) or \((N(b_3) \cap N_2(v)) \subset (N(b_m) \cap N_2(v))\); on the other hand, if \((N(b_2) \cap N_2(v)) \subset (N(b_m) \cap N_2(v))\), we have the same result that either \( b_m \Delta b_3 \) or \((N(b_3) \cap N_2(v)) \subset (N(b_m) \cap N_2(v))\). If we continue to iterate the cycle, at some point we have \( b_m \Delta b_{m-1} \) or \((N(b_m) \cap N_2(v)) \subset (N(b_{m-1}) \cap N_2(v))\). But since \( b_m \Delta b_m \), \( b_m \Delta b_{m-1} \) is not possible unless \( b_m \) and \( b_{m-1} \) interleave. In addition, \((N(b_m) \cap N_2(v)) \neq (N(b_{m-1}) \cap N_2(v)) \) as \( b_{m-1} \) should cover at least two nodes in \( N_2(v) \) to satisfy \( \Delta b_{m-1} b_m \). In both cases, we have a contradiction indicating that domination relation cannot be a cyclic relation.

**4. A space efficient solution to MFSP problem**

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., \( n = |N(v)| \). 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 \((a_i, a_{i+1}, ..., a_{j-1}) \in N_2(v) \) considered by each of these subproblems. The subproblems are solved by using the solutions of the smaller size subproblems from previous rounds.

Let \( L_{\text{min}}(a_i, j) \) 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 \( L_{\text{min}}(a_i, n) \) for \( i = 1, n \). The optimal solution is given by the \( L_{\text{min}}(a_i, n) \).

**4.1. Solution approach**

Our algorithm uses a bottom-up approach to build an optimal solution. Given an RCN interval \((a_i, j)\), we know the optimal solutions for continuous subintervals of \((a_i, j)\), i.e., RCN intervals \((a_i, k)\) where \( 1 \leq k < j \), as these solutions are computed in previous rounds of the algorithm before we consider \((a_i, j)\).

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 2, we know that \( S' \) is not dominated by any other nodes in \( S \). Assume that \( b_t \) has \( t \) MCTs, \( (B_t, ..., B_i) \), where \( 1 \leq t \leq j(2t) \) in \((a_i, j)\). We refer to the RCNs that \( b_t \) does not cover as \( R_k \) for \( k \in [1, t] \). Using \( B_t \)s and \( R_k \)s, we can rep-

\(^1\) Note that the arithmetic in computing the subscripts of the nodes preserves radial circularity.
Lemma 6. For \( t = 2 \), \((a_t, j) = (B_1 \cup R_1 \cup B_2 \cup R_2, b) \) and \( S = \{ b \} \cup c(R_1 \cup R_2) \), we claim that \( |c(R_1 \cup R_2) - |c(R_1 \cup R_2) \cup b|\) is a minimum cardinality solution \( c(R_1 \cup R_2) \cup b \).\]

Proof of Lemma 6. 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 R_2) - |c(R_1 \cup R_2) \cup b|\). \tag{1}
\]

where \( |c(R_1 \cup R_2) \) represents the cardinality of the solution \( c(R_1 \cup R_2) \). Note that we know that the solution for the first case above. For the second case, the coverage of \( c(R_1 \cup R_2) \) results in a better solution. If Eq. (2) holds, then \( |c(R_1 \cup R_2) - |c(R_1 \cup R_2) \cup b|\). This is not correct only if \( \exists b \in S \) \( c(R_1 \cup R_2) \) which covers nodes from both \( R_1 \) and \( R_2 \). Assume such a \( b \) exists. By definition \( b \) does not cover any nodes in \( R_1 \) or in \( R_2 \). Now, if \( b \in S \), then \( b \) has to cover all nodes in \( R_1 \) as otherwise \( b \) dominates \( b \) (recall that \( b \) is not dominated by any node in \( S \)). Alternatively, if \( b \notin S \), then \( b \) does not cover \( a_t \). In this case, it should cover all nodes in \( R_2 \) as otherwise \( b \) and \( b \) will have an interleaving coverage. Consequently, if \( b \) covers all nodes in \( R_2 \), this results in \( |c(R_1 \cup R_2) - |c(R_1 \cup R_2) \cup b|\) which contradicts the initial assumption. As a result, if \( |c(R_1 \cup R_2) - |c(R_1 \cup R_2) \cup b|\), then \( b \) cannot cover nodes from both \( R_1 \) and \( R_2 \) simultaneously, i.e., \( |c(R_1 \cup R_2) - |c(R_1 \cup R_2) \cup b|\).\]

The above observation can be generalized for an arbitrarily large \( (R_1 \cup R_2 \cup \ldots \cup R_k) \).

Lemma 7. Given \((a_t, j) = (B_1 \cup R_1 \cup \ldots \cup B_k \cup R_k) \) and an optimal solution \( S = \{ b \} \cup c(R_1 \cup R_2 \cup \ldots \cup R_k) \), we claim that:

\[
|c(R_1 \cup R_2 \cup \ldots \cup R_k) - |c(R_1 \cup R_2 \cup \ldots \cup R_k) \cup b|\) \min\{(|c(R_1 \cup \ldots \cup R_0) + |c(R_0 \cup \ldots \cup R_j)|), \forall h \in [2, j/2]\}\]. \tag{3}
\]

Proof of Lemma 7. There are two alternatives for the solution \( c(R_1 \cup R_2 \cup \ldots \cup R_k) \):

\[
|c(R_1 \cup R_2 \cup \ldots \cup R_k) - |c(R_1 \cup R_2 \cup \ldots \cup R_k) \cup b|\) \min\{(|c(R_1 \cup \ldots \cup R_0) + |c(R_0 \cup \ldots \cup R_j)|), \forall h \in [2, j/2]\}\]. \tag{4}
\]

Similar to the basic case above, we know the solution for the case in Eq. (4). For the case in Eq. (5), we claim that we can find the optimal solution by splitting the interval \( (R_1 \cup R_2 \cup \ldots \cup R_k) \) into two as \( (R_1 \cup R_2 \cup \ldots \cup R_n) \) and \( (R_{n+1} \cup \ldots \cup R_k) \). In this case, the optimal solution is given by:

\[
|c(R_1 \cup R_2 \cup \ldots \cup R_k) - |c(R_1 \cup R_2 \cup \ldots \cup R_k) \cup b|\) \min\{(|c(R_1 \cup \ldots \cup R_0) + |c(R_{n+1} \cup \ldots \cup R_k)|), \forall h \in [2, j/2]\}\]. \tag{5}
\]

Eq. (6) may not be correct if \( \exists b \) such that \( b \in c(R_1 \cup \ldots \cup R_n) \) and \( b \notin c(R_{n+1} \cup \ldots \cup R_k) \). Assume that such a \( b \) exists. If \( b \notin S \), then \( b \) has to cover all nodes in \( B_{n+1} \), as otherwise \( b \) dominates \( b \) (recall that \( b \) is not dominated by any node in \( S \)). Alternatively, if \( b \notin S \), then \( b \) does not cover \( a_t \). But in this case, it should cover all nodes in \( B_{n+1} \), as otherwise \( b \) and \( b \) will have an interleaving coverage. Consequently, if \( b \) covers all nodes in \( B_{n+1} \), this results in \( |c(R_1 \cup R_2 \cup \ldots \cup R_k) - |c(R_1 \cup R_2 \cup \ldots \cup R_n) \cup b|\) simultaneously, i.e., \( |c(R_1 \cup R_2 \cup \ldots \cup R_k) - |c(R_1 \cup R_2 \cup \ldots \cup R_n) \cup b|\) \min\{(|c(R_1 \cup \ldots \cup R_0) + |c(B_{n+1} \cup \ldots \cup R_k)|), \forall h \in [2, j/2]\}\]. \tag{6}
\]

Using Eq. (7), we can find \( c(R_1 \cup R_2 \cup \ldots \cup R_k) \) in \( O(n^3) \) as follows. We know the optimal solutions for \( c(R_1) \), \( c(R_2) \), \ldots, \( c(R_k) \). In the second step, we find the optimal solutions for each \( c(R_k \cup R_{k+1} \cup \ldots \cup R_{k+2}) \) for \( 1 \leq k \leq t - 1 \) as:

\[
|c(R_k \cup R_{k+1} \cup \ldots \cup R_{k+2}) - |c(R_k \cup R_{k+1} \cup \ldots \cup R_{k+2}) \cup b|\) \min\{(|c(R_k \cup R_{k+1} \cup \ldots \cup R_{k+2}) + |c(R_k \cup R_{k+1} \cup \ldots \cup R_{k+2})|), \forall h \in [2, j/2]\}\]. \tag{7}
\]

and in the third step, the optimal solutions for \( c(R_k \cup R_{k+1} \cup \ldots \cup R_{k+2}) \) for \( 1 \leq k \leq t - 2 \) as:

\[
|c(R_k \cup R_{k+1} \cup \ldots \cup R_{k+2}) - |c(R_k \cup R_{k+1} \cup \ldots \cup R_{k+2}) \cup b|\) \min\{(|c(R_k \cup R_{k+1} \cup \ldots \cup R_{k+2}) + |c(R_k \cup R_{k+1} \cup \ldots \cup R_{k+2})|), \forall h \in [2, j/2]\}\]. \tag{8}
\]

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 third step, we have the optimal solution for...
the entire interval of \( R_1 \cup R_2 \cup \ldots \cup R_t \). The process is similar to building a pyramid starting with \( t \) blocks at the bottom 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 \( b_k \) prior to above calculation, we can find the optimal solution for the entire interval as above. Trials with incorrect candidates \( b_k \neq b_i \) will result in suboptimal solutions. The running time will then be \( O(n^2) \). Unfortunately, we do not know \( b_k \) 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 \( O(mn^2) \).

4.2. 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., \((a_1, 1)\) 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 \((a_n, n)\). As a result, the overall complexity of the algorithm is given by \( O(mn^2) \) where \( O(n^2) \) comes from the number of \((a_i, j)\) intervals and \( O(mn^2) \) comes from the computational complexity of the solution for each such interval.

Space complexity of this algorithm is \( O(n^2) \) since we save a solution for each subinterval. Note that there might be up to \( m \) nodes in a solution but we do not need to save all this information. Instead we can save from which sub-solutions we obtain this solution. Since we use at most two sub-solutions to construct a new solution. Space requirement to save the optimal solution of a given interval is constant and overall space complexity is \( O(n^2) \).

In the algorithm outlined in Fig. 5, the term \( L_{\text{min}}(a_i, j) \) considers the solution for an interval \((a_i, j)\) of \( N(v) \). On the other hand, the term \( P_{\text{min}}(R_\text{x}, y) \) considers the solution for \( y \) intervals that \( b_x \) does not cover, i.e., \((R_\text{x}, y) = (R_\text{x} \cup R_{\text{x}+1} \cup \ldots \cup R_{\text{y}+1})\). \( N_{\text{min}}(a_i, j) \) represents the size of the solution \( L_{\text{min}}(a_i, j) \), i.e., \( N_{\text{min}}(a_i, j) = |L_{\text{min}}(a_i, j)| \) for \((a_i, j)\) and \( K_{\text{min}}(R_\text{x}, y) \) represents the size of the solution \( P_{\text{min}}(R_\text{x}, y) \), i.e., \( K_{\text{min}}(R_\text{x}, y) = |P_{\text{min}}(R_\text{x}, y)| \). Note that for a given interval, there may be multiple optimal solutions but our algorithm maintains the set of nodes for only the first one it finds (see line 23 in Fig. 5). Finally, if there are multiple 1-hop neighbors at a location, we choose one of them with the maximum transmission range (largest disk) in a pre-processing step before running the algorithm. Similarly, if there are multiple 2-hop neighbors at a location, we eliminate all but one from the 2-hop neighbor set.

5. A faster algorithm with more space requirement

In this section we provide an alternative algorithm which has better runtime complexity. This improvement is obtained by saving more sub-solutions and using more space. As we discussed earlier, run-time complexity of our initial algorithm is \( \Theta(n \cdot n^2) \) while space complexity is \( \Theta(n^2) \). In this section we provide an alternative algorithm which runs in \( \Theta(m \cdot n^2) \) and uses \( \Theta(m \cdot n^2) \) units of space.

**Definition 11.** \( L_{\text{min}}(a_i, j, b_k) \) is the optimal solution for interval \((a_i, j)\) which assumes \( b_k \) is in \( L_{\text{min}}(a_i, j, b_k) \). In other words \( L_{\text{min}}(a_i, j, b_k) - b_k \) is the minimum cardinality subset of 1-hop nodes which covers all nodes in \((a_i, j)\) which cannot be covered by \( b_k \).

**Definition 12** (Super coverage). A 1-hop node \( b_k \) super-covers a 2-hop node \( a_i \) in solution set \( S \) if \( b_k \) has the furthest coverage on line \((v, a_i)\), \( v \) is the center node and we draw a line from \( v \) crossing \( a_i \) among nodes in \( S \). We define the furthest point covered on line \((v, a_i)\) as \( f_x \).

Algorithm is presented as Fig. 6. We prove the correctness of this algorithm by proving that it finds the optimal solution for \((a_i, j)\). Assume \( S \) is the optimal solution to cover \((a_i, j)\), \( b_k \in S \) and \( b_k \) super-covers \( a_i \). Note that \( b_k \) always exists for a given \( S \). Now we show that the algorithm finds a solution which is as good as \( S \) when it computes \( L_{\text{min}}(a_i, j, b_k) \). We assume that we keep all previously computed optimal solutions \( L_{\text{min}}(a_i, t, b_k) \) and \( L_{\text{min}}(a_i, t) \) where \( t < j \).

If \( b_k \) covers \( a_{i,j} \), \( L_{\text{min}}(a_i, j - 1, b_k) \) is an optimal solution for \( L_{\text{min}}(a_i, j, b_k) \). If \( b_k \) does not cover \( a_{i,j} \), we define the last (right most) node that \( b_k \) super covers in \((a_i, j)\) as \( h \). We assign nodes in \((a_i, j)\) to nodes in \( S \). Each 2-hop node is assigned to 1-hop node which super-covers it. Note that we do not assign any node in \((a_{i,j+1}, j - h)\) to \( b_k \) since it does not super-cover any node in this interval. We define 1-hop nodes which are used in the assignment of 2-hop nodes in \((a_i, h, b_k)\) as \( S_1 \) and 1-hop nodes which are used in the assignment of 2-hop nodes in \((a_{i,j+1}, j - h)\) as \( S_2 \).

**Lemma 8.** \( S_1 \cap S_2 = \emptyset \).

**Proof of Lemma 8.** This lemma would not be correct if there is a node \( b_k \in S \) which super-covers nodes from intervals \((a_i, h - 1)\) and \((a_{i,j+1}, j - h)\). Now we create four artificial nodes at points \( f_1, f_2, f_3, f_4 \). Note that, among these four points \( b_k \) covers only \( f_1 \) and \( f_3 \) while \( b_k \) covers only \( f_2 \) and \( f_4 \). This coverage scenario is impossible and \( b_k \) cannot exist due to Non-Interleaving Property.

**Lemma 9.** \( |L_{\text{min}}(a_i, h, b_k)| \leq |S_1| \).

**Proof of Lemma 9.** \( L_{\text{min}}(a_i, h, b_k) \) is the optimal solution to cover \((a_i, h)\) assuming \( b_k \) is in this solution. \( S_1 \) is a solution which covers \((a_i, h)\) and \( b_k \in S_1 \).

**Lemma 10.** \( |L_{\text{min}}(a_{i,j+1}, j - h)| \leq |S_2| \).

**Proof of Lemma 10.** \( L_{\text{min}}(a_i, h, b_k) \) is the optimal solution to cover \((a_{i,j+1}, j - h)\). \( S_2 \) is a solution which covers \((a_{i,j+1}, j - h)\).
Lemma 11. \[ |L_{\text{min}}(a_i, h, b_k)| + |L_{\text{min}}(a_{i+1}, j - h)| \leq |S_i| - |S_j| \]

Proof of Lemma 11. Combination of last two lemmas. □

In calculating \(L_{\text{min}}(a_i, j, b_k)\), one of the alternative solutions that algorithm checks for is the combination of \(L_{\text{min}}(a_i, h, b_k)\) and \(L_{\text{min}}(a_{i+1}, j - h)\). Since we have shown total size of this combination is less than \(|S_i|\), the solution
algorithm produces for $L_{\text{min}}(a_i, j, b_k)$ should be an optimal solution for interval $(a_i, j)$ as we defines $S$ as an optimal solution for $(a_i, j)$.

6. Conclusions

In this paper, we have presented polynomial time solutions 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 paper, we have considered arbitrary disk model as a more general model for wireless transmission and developed optimal solutions with different run time and space complexities to the problem. We believe that the results presented in this paper 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.

References


Mehmet Baysan received M.S. and Ph.D. degrees in Computer Science from The University of Texas at Dallas in 2005 and 2008. He is currently a Visiting Fellow at Neuro-Oncology Branch in National Cancer Institute in National Institutes of Health. His research interests include Bioinformatics, Algorithms, Combinatorial Optimization and Wireless Networks.
Kamil Sarac received the M.S. and Ph.D. degrees in computer science from the University of California at Santa Barbara, in 1997 and 2002, respectively. He is currently an Associate Professor in the Department of Computer Science at the University of Texas, Dallas. His research interests include computer networks and protocols; network security and Internet measurements; and energy efficient broadcast in ad hoc networks.

R. Chandrasekaran is currently Ashbel Smith Professor of Computer Science and Operations Research at The University of Texas at Dallas. He received his Ph.D. from the University of California at Berkeley. He has served as a faculty member at Case Western Reserve University and been a visitor at Northwestern and Tel Aviv Universities. His research interests are in combinatorial optimization, computational geometry, and integer programming with applications to telecommunication networks.