Placement and Range Assignment in Power-Aware Radio Networks

Doctoral dissertation submitted by

Gautam Kumar Das

for award of the Ph.D. degree of the

Indian Statistical Institute, Kolkata

Advisor :

Professor Subhas Chandra Nandy

Advanced Computing and Microelectronics Unit

Indian Statistical Institute

203 B. T. Road, Kolkata 700 108, India

Placement and Range Assignment in Power-Aware Radio Networks

Doctoral dissertation submitted by

Gautam Kumar Das

for award of the Ph.D. degree of the

Indian Statistical Institute, Kolkata

Advisor :

Professor Subhas Chandra Nandy

Advanced Computing and Microelectronics Unit

Indian Statistical Institute

203 B. T. Road, Kolkata 700 108, India

Submission date of the first version: June 28, 2007

Submission date of the revised version: February 18, 2008

DEDICATION

to my grandmother

Laxmi Rani Das
(1921-1990)

Acknowledgement

My languages are inadequate to convey my gratitude towards my supervisor Professor Subhas Chandra Nandy. He had introduced me to the field of geometric algorithms and its application in mobile computing. It has been a wonderful experience, ever since. His immaculate conception and tireless effort inspired me a lot. I express my deepest gratitude for his invaluable advice and guidance that made me more matured, both academically and personally.

It has been my privilege to have valuable suggestions and novel ideas from Dr. Sandip Das. I am thankful for his uninterrupted help throughout my research period.

My special thanks to Professor Bhabani P. Sinha who greatly enriched my knowledge with his exceptional insight in Computer Science. I am also thankful for his support and advice.

I owe a huge debt of gratitude to Professor Bhargab B. Bhattacharya for his valuable comments, suggestions and encouragement. His comments and criticisms have led a vast improvement in the quality of the thesis.

I am seriously indebted to the Examiner(s) of this thesis for their valuable suggestions for improving the quality of the thesis.

I must also mention the encouragement, advice, and support I received from Professor Jayasree Dattagupta, Professor Nabanita Das, Dr. Susmita Sur-kolay, Dr. Krishnendu Mukhopadhyaya, Dr. Partha Pratim Goswami and Dr. Srabani Mukhopadhyaya. My heartfelt thanks to all of them. I would also like to express my sincere thanks to Dr. Arijit Bishnu of Indian Institute of Technology Kharagpur for constantly encouraging me and providing me help in all respect. I am also thankful to Mrs. Sukriti Nandy for her nonstop encouragement and help throughout my research.

I cherished my stay in the institute largely due to my friends. I thank Subhasisda, Sasthi-da, Partha-da, Pritha-di, Buddha-da, Sasanka, Debasis, Sandeep, Arindam, Sumana, Ritwik, Debasri, Alokananda, Bibhas, Bidyut, Suman, Rahul, Debapriyaya and many others for their encouragement and delightful company. I am specially thankful to Sasthi-da for helping me to do research in early stage. I express my sincere thanks to Pujari-da and other office staffs of Indian Statistical Institute for assistance in official matters.

I would like to thank the Indian Statistical Institute for providing me a good research environment, the opportunity to get acquainted with distinguished researchers and financial support during this research works.

I feel a deep sense of gratitude for my parents and other family members. Specially, I thank my elder brother Mr. Debabrata Das who always encourages me in my study since childhood.

(Gautam Kumar Das)
February 18, 2008
Advanced Computing and Microelectronics Unit
Indian Statistical Institute
Kolkata 700 108

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Due to the extraordinary growth of demand in mobile communication facility, design of efficient systems for providing specialized services has become an important issue in wireless mobility research. Broadly speaking, there are two major models for wireless networking: *single-hop* and *multi-hop*. The single-hop model [110] is based on the cellular network, and it provides one-hop wireless connectivity between the host and the static nodes known as base stations. single-hop networks rely on a fixed backbone infrastructure that interconnects all the base stations by high speed wired links. On the other hand, the multi-hop model requires neither fixed wired infrastructure nor predetermined interconnectivity [83]. Two main examples where the multi-hop model is adopted, are ad hoc network and sensor network [111].

Ad hoc wireless networking is a technology that enables untethered wireless networking in the environments where no wired or cellular infrastructure is available, or if available, is not adequate or cost-effective. Indeed, in an ad hoc wireless network, the wireless links are established based on the ranges assigned to the radio stations. Ad hoc networking is the most popular type of multi-hop wireless network because of its simplicity (see Haas and Tabrizi [71]). This type of networking is useful in many practical applications, for example in a battlefield, for disaster management, etc. One of the main challenges in ad hoc wireless networks is the minimization of energy consumption. This can be achieved in several ways. The most important issues in this context are range assignment to the radio stations, and efficient routing of the packets as described below:

Range assignment: Assigning range (a non-zero real number) to the radio stations in the network. This enables each radio station to transmit packets to the other radio stations within its range. Here, the goal is to assign ranges to the radio stations such that the desired communication among the radio stations can be established, and the total power consumption of the entire network is minimized.

Routing: Transmission of packets from the source radio station to the destination radio station. Here, the power consumption of the network can be minimized by the choice of an appropriate path from source radio station to destination radio station.

On the other hand, a *wireless sensor network* (WSN) consists of large collection of co-operative small-scale nodes which can sense, perform limited computation, and can communicate over a short distance via wireless media. A WSN is self-organized in nature, and its members use short range broadcast communication to send the collected information to the base station in multi-hop fashion.

In this thesis, we deal with the algorithmic aspects of the range assignment problem with a focus on the minimization of the total power requirement of the network maintaining its desired connectivity property. Two important sub-problems in this area are:

- The radio stations are pre-placed, and the objective is to assign ranges to the radio stations such that the network maintains some specific connectivity property. This problem is referred to as the *range assignment problem.*

- The radio stations are not pre-placed; the objective is to compute the positions and ranges of the radio stations such that the entire network maintains the desired connectivity property, and the total cost of range assignment in the entire network is minimized. This problem is referred to as the *base station placement problem*.

Specifically, we consider the range assignment problem for broadcasting and all-to-all communication, when the radio stations are placed on a line and on a 2D plane. In the base station placement problem, we consider both the unconstrained and constrained version. In the unconstrained version, the base stations can appear any where inside the desired (convex) region. In the constrained version, base stations can appear only on the boundary of the desired (convex) region.

In the next two sections of this chapter, we discuss these two problems in detail. We formulate these optimization problems, and deduce geometric characterizations to develop efficient algorithms for solving these problems. In Section 1.3, a detailed literature survey on both the range assignment and base station placement problems are given. The explicit mention of the scope of this thesis appears in Section 1.4.

1.1 Range assignment problem

A radio network is a finite set $S = \{s_1, s_2, \ldots, s_n\}$ of radio stations located in a geographical region. These radio stations can communicate with each other by transmitting and receiving radio signals. Each radio station $s_i \in S$ is assigned a range $\rho(s_i)$ (a non-negative real number) for communication with the other radio stations. This range assignment $\mathcal{R} = \{\rho(s_1), \rho(s_2), \ldots, \rho(s_n)\}$ defines a directed graph $G = (V, E)$, where $V = S = \{s_1, s_2, \ldots, s_n\}$, and $E = \{(s_i, s_j) | s_i, s_j \in S, d(s_i, s_j) \leq \rho(s_i)\}$. From now onwards, the graph G will be referred to as the *communication graph*.

A directed edge $(s_i, s_j) \in E$ indicates that $d(s_i, s_j) \leq \rho(s_i)$ and hence s_i can communicate (i.e., send a message) directly (i.e., in 1 hop) to any other radio station s_j. If s_i

cannot communicate directly with s_j because of the insufficiency of its assigned range, then communication between them can be achieved by multi-hop transmission along a path from s_i to s_j in G, where the intermediate radio stations on the path cooperate with the source node and forward the message till its destination s_j is reached. Sometimes in a radio network, link failure occurs with some probability and all such failures occur independently. In multi-hop transmission, the probability of link failure on a transmission path increases with the number of hops. Thus, for multi-hop transmission, the reliability of communication can be ensured by bounding the number of hops for communication, considering the probability of failure of the radio stations. Several other problems related to bounded hop communication are available in the literature [14, 57, 63, 127]. If the maximum number of hops (h) allowed is small, then communication between a pair of radio stations is established very quickly, but the power consumption of the entire radio network may become very high [35]. On the other hand, if h is large, then the power consumption decreases, but communication delay is likely to increase. The impact of tradeoff between the power consumption of the radio network and the maximum number of hops needed between a communicating pair of radio stations has been studied extensively [41, 76, 90]. For more information about the tradeoff between connectivity and power consumption of the network, see [25, 42, 98, 112, 125]. The power required by a radio station s_i (denoted as $power(s_i)$) to transmit a message to another radio station s_j should satisfy

$$power(s_i) > \gamma \times (d(s_i, s_j))^{\beta} \tag{1.1}$$

where $d(s_i, s_j)$ is the Euclidean distance between s_i and s_j, β is referred to as the distance-power gradient, and $\gamma(\geq 1)$ is the transmission quality of the message [101]. In the ideal case (i.e., free-space environment without any obstruction in the line of sight, and in the absence of reflections, scattering, diffraction caused by buildings, terrains etc.), we may assume $\beta = 2$ and $\gamma = 1$. Note that, the values of β may vary from 1 to 6 depending on various environmental factors, and the value of γ may also vary based on several other environmental factors, for example, noise, weather condition, etc. The more realistic model is to consider γ as a function of the radio station s_i. Here $\gamma(s_i)$ is referred as the weight of the radio station s_i and it depends on the positional parameters of s_i. Thus

$$power(s_i) = \gamma(s_i) \times (\rho(s_i))^2 \tag{1.2}$$

and the total cost of a range assignment $\mathcal{R} = \{\rho(s_i) \mid s_i \in S\}$ is written as

$$cost(\mathcal{R}) = \sum_{s_i \in S} power(s_i) = \sum_{s_i \in S} \gamma(s_i) \times (\rho(s_i))^2 \tag{1.3}$$

This version of the range assignment problem is referred to as the *weighted range assignment problem*. Unless otherwise specified, we assume that $\gamma(s_i) = 1$ for all $s_i \in S$, and hence the cost of the range assignment $\mathcal{R} = \{\rho(s_i) \mid s_i \in S\}$ is

$$cost(\mathcal{R}) = \sum_{s_i \in S} power(s_i) = \sum_{s_i \in S} (\rho(s_i))^2 \tag{1.4}$$

Note that, Equation 1.2 accounts for only the transmission power, i.e., the power consumed by the sender radio stations. In practice, a non-negligible amount of energy is also consumed at the receiver end to receive and decode the radio signals. Throughout this thesis, we consider only the energy consumed by the transmitting radio stations, since most of the existing literature do not account for the energy consumed for receiving a message.

If the radio stations are pre-placed, then the following three types of range assignment problem are considered in the literature depending on the communication criteria:

1. Range assignment problem for broadcasting a message from a source radio station to all the target radio stations,

2. Range assignment problem for all-to-all communication,

3. Range assignment problem for accumulation of messages to a target radio station from all other radio stations in the network.

We assume two variations of the problem depending on whether the radio stations are arranged on a straight line or on a 2D plane. These are referred to as 1D- and 2D-version respectively. The simple 1D model produces more accurate analysis of some typical situation arising in vehicular technology applications [90]. For an example, consider the road traffic information system where the vehicles follow roads, and messages are broadcasted along lanes [16, 39, 48, 77, 90]. Typically, the curvature of a road is small in comparison to the transmission range; so we may consider the road is a straight line. For several other vehicular technology applications of this problem, see [31, 61, 76].

The 2D version of the range assignment problems are more realistic, but are often computationally hard in nature. We could propose efficient algorithm for some restricted variation of those problems.

1.2 Base station placement problem

In this sub-problem, the objective is to identify the locations for placing the base stations and to assign ranges to the base stations for efficient radio communication. Each mobile terminal communicates with its nearest base station, and the base stations communicate with each other over scarce wireless channels in a multi-hop fashion by receiving and transmitting radio signals. Each base station emits signal periodically and all the mobile terminals within its range can identify it as its nearest base station after receiving such radio signal. Here, the problem is to position the base stations such that a mobile terminal at any point in the entire area can communicate with at least one base station, and the total power required for all the base stations in the network is minimized. Another variation of this problem arises when there are forbidden zones for placing the base stations, but communication is to be provided over these regions. Example of such forbidden regions may include large water bodies, or stiff mountain terrains. In such cases, we need some specialized algorithms for efficiently placing the base stations on the boundary of the forbidden zone to provide services within that region. These two variations of base station placement problem are referred to as

(i) Unconstrained version of the base station placement problem, and

(ii) Constrained version of the base station placement problem.

1.3 Review of Related Works

The range assignment problem for ad hoc wireless networks is studied extensively in the context of all-to-all communication, information broadcast and information accumulation [111]. In this context, it is very much important to minimize power consumption while maintaining the aforesaid properties of the network.

1.3.1 Broadcast range assignment problem

The objective of the broadcast range assignment problem is to assign transmission ranges $\rho(s_i)$ to the radio stations $s_i \in S$ so that a dedicated radio station (say $s^* \in S$) can transmit messages to all other radio stations, and the total power consumption of the entire network is minimum. The graph-theoretic formalization of the problem is as follows:

> Compute a range assignment $\mathcal{R} = \{\rho(s_1), \rho(s_2), \ldots, \rho(s_n)\}$ such that there exists a directed spanning tree rooted at s^* in the communication graph G, and the total cost of the range assignment $\sum_{i=1}^{n}(\rho(s_i))^2$ is minimum.

The directed spanning tree rooted at s^* is referred to as the *broadcast tree.* In the bounded hop broadcast range assignment problem, the objective is to compute a range assignment $\mathcal{R}$ of minimum cost that realizes a broadcast tree of height bounded by a pre-specified integer h. The hardness result of the broadcast range assignment problem depends on different parameters, namely, the distance power gradient β in the cost function (Equation 1.1), h the maximum number of hops allowed, the dimension (d) of the plane where the radio stations are located, and the edge weight function. In general, we assume that $w(s_i, s_j)$ is equal to the Euclidean distance between the radio stations s_i and s_j. If $0 < \beta \leq 1$ or $h = 1$ the problem is trivially polynomial time solvable because it suffices to set the range of the source s^* equal to the maximum weight among the edges incident on it. Or equivalently, assign the range of s^* equal to the distance of the furthest radio station from it.

For the unbounded case ($h = n-1$), the broadcast range assignment problem is proved to be NP-hard for any $\beta \geq 2$ [24, 28]. The authors of [24] also suggested a very high constant factor approximation algorithm for this problem. Fuchs [53] studied this problem in a restricted setup, called the *well-spread* instances, which is defined in [41] as follows:

> Let $\Delta(S) = max\{d(u,v) \mid u,v \in S\}$, $\delta_s(S) = min\{d(s,v) \mid v \in S \setminus \{s\}\}$, and $\delta(S) = min\{\delta_s(S) \mid s \in S\}$. A set of radio stations in S are said to be *well-spread* if there exists some positive constant c such that $\delta(S) \geq \frac{c\Delta(S)}{\sqrt{|S|}}$.

In that paper, it is proved that for any $\beta > 1$, the broadcast range assignment problem is NP-hard for a set of radio stations which are well-spread in 2D. Clementi et al. [24] considered the general combinatorial optimization problem, called *minimum energy consumption broadcast subgraph* (MECBS) problem, which is stated as follows:

> Given a weighted directed graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$, and each edge $(v_i, v_j) \in E$ is attached with a positive weight $w(v_i, v_j)$. The *transmission graph* induced by a range assignment $\mathcal{R} = \{\rho(v_1), \rho(v_2), \ldots, \rho(v_n)\}$ is a subgraph $G_{\mathcal{R}}(V, E_{\mathcal{R}})$ of G, where $E' = \{(v_i, v_j) | w(v_i, v_j) \leq \rho(v_i)\}$.

The MECBS problem is then defined as follows: *given a source node* $v_i \in V$*, find a range assignment* $\mathcal{R}$ *such that* $G_{\mathcal{R}}$ *contains a spanning tree rooted at* v_i*, and* $cost(\mathcal{R}) = \sum_{i=1}^{n} \rho(v_i)$ *is minimum.*

The bounded hop version of MECBS problem is named as h-MECBS problem, and is defined as follows: *given a source node* $v_i \in V$*, find a range assignment* $\mathcal{R}$ *such that* $G_{\mathcal{R}}$ *contains a spanning tree of height at most* h *rooted at* v_i*, and* $cost(\mathcal{R}) = \sum_{i=1}^{n} \rho(v_i)$ *is minimum.*

It is proved that, both the MECBS and h-MECBS problems are NP-hard [24, 28]. But, this does not imply that the h-hop broadcast range assignment problem is NP-hard. The reason is that, here the weight of each edge (v_i, v_j) in G is equal to the Euclidean distance of the radio stations in S corresponding to the nodes v_i and v_j. To our knowledge, no hardness result is available for the bounded hop broadcast range assignment problem.

For the 1D version of the problem, a dynamic programming based algorithm is proposed in [31], which runs in $O(hn^2)$ time. In this thesis, we propose an improved algorithm for this problem, which is based on the geometric properties of the problem, and the running time of this algorithm is $O(n^2)$.

Several researchers studied on developing good approximation/heuristic algorithms for the broadcast range assignment problem. The most popular heuristic for this problem is based on the minimum spanning tree (MST), and is stated below.

Construct a weighted complete graph $G = (V, E)$, where $V = S$ (the set of radio stations), and the edge weight $w(s_i, s_j) = d(s_i, s_j)$. Compute the MST T of G. Assign the range of a radio station $s_i = \rho(s_i) = \max\{w(s_i, s_j) | (s_i, s_j) \in T\}$.

As MST is always connected, the communication graph derived from this range assignment is also connected. Wieselthier et al. [130] proposed a greedy heuristic, called *broadcast incremental power* (BIP), which is a variant of Prim's algorithm for MST, and is applicable for any arbitrary dimension d ($d > 1$). At each step, instead of adding the edge having minimum weight in the MST, a node that needs minimum extra energy is added. This formulation is obvious due to the broadcast nature of the problem, where increasing the radius of an already emitting node to reach a new node is less expensive than creating a new emitting node. Some small improvements of this method was proposed by Marks et al. [88]. A different heuristic paradigm, namely *embedded wireless multicast advantage* (EWMA) is described by Cagalj et al. [28], which is an improvement over MST based algorithm. It takes the MST as the initial feasible solution, and builds an energy efficient broadcast tree. In EWMA, every forwarding node in the initial solution is given a chance to increase its power level. This may decrease the power level of some other nodes maintaining the network connectivity. This assignment of new power level to the concerned node is acceptable if cost of the tree decreases. Each node finally chooses the power level at which the overall decrease in cost of the final tree is maximized. Assuming complete knowledge of distances for all pairs of radio stations, Das et al. [49] proposed three different integer programming (IP) formulations for the minimum energy broadcast problem.

The distributed version of this problem was studied by Wieselthier et al. [131]. Although it works well for small instances, its performance degrades when the number of radio stations becomes large. The reason is that, it needs communication for exchanging data in distributed environment for constructing the global tree. Ingelrest and Simplot-Ryl [72] proposed a localized version of the BIP heuristic in the distributed set up. Here, each node apply the BIP algorithm on its 2-hop neighbors, and then include the list of its neighbors who need to retransmit, together with the transmission ranges with the broadcast packet. It is experimentally observed that the result offered by this algorithm is very close to the one obtained by BIP with global knowledge of the network. Below, we provide a scheme for computing 1-hop and 2-hop neighbors of each node.

Let the radio stations be distributed in a 2D region. Each radio station knows its position using a location system (say GPS) [75]. Each radio station broadcasts a "HELLOW" message with its own coordinate. A radio station that receives such a message can identify the sender and it notes that the sender is in its 1-hop neighborhood. Using the 1-hop neighbor information, the 2-hop neighbors of each node can be computed after the second round of exchange. After knowing the positions of 1-hop and 2-hop neighbors, each node can easily compute the distances of its 1-hop and 2-hop neighbors.

Cartigny et al. [44] proposed a distributed algorithm for the broadcast range assignment problem that is based on the *Relative Neighborhood Graph* (RNG) [123]. The RNG preserves connectivity, and based on the local information, the range assignment is done as follows: for each node s_i, compute its furthest RNG neighbor s_j, excepting the one from which the message is received, and assign a range $d(s_i, s_j)$ to the node s_i. They experimentally demonstrated that their algorithm performs better than the solution obtained by the sequential version of the BIP algorithm. Cartigny et al. [43] described localized energy efficient broadcast for wireless networks with directional antennas. This is also based on RNG. Messages are sent only along RNG edges, and the produced solution requires about 50% more energy than BIP. More reviews on broadcast problem are available in [85, 122]. Now we mention the state-of-the-art performance bound of the MST and BIP based algorithms for the broadcast range assignment problem.

Wan et al. [129] proved that the approximation ratio of the MST based heuristic is between 6 and 12, whereas the approximation ratio of the centralized BIP is between $\frac{13}{3}$ and 12. Unfortunately, there was a small error in [129]. Klasing et al. [79] corrected the analysis and proved that the upper bound of the approximation ratio of the MST based algorithm is actually 12.15. Further, Flammini et al. [55] proved that the approximation ratio of the MST based algorithm is 7.6. Navarra [95] improved the approximation ratio to 6.33. Finally, Ambhul [1] improved the approximation factor of the MST based algorithm to 6; thus it attains the lower bound proposed in [129].

Calamoneri et al. [23] proved an almost tight asymptotic bound on the optimal cost for the minimum energy broadcast problem on the square grid. Finally Calinescu et al. [36] presented $(O(\log n), O(\log n))$ bicriteria approximation algorithm for *h-hop broadcast range assignment problem*. The solution produced by this algorithm needs $O(h \log n)$ number of hops, and cost is at most $O(\log n)$ times the optimum solution. They also presented an $O(\log^{\beta} n)$-approximation algorithm for the same problem, where the radio stations are installed in d-dimensional Euclidean spaces, and β is the distance-power gradient.

Clementi et al. [24] considered the general combinatorial optimization problem, called *minimum energy consumption broadcast subgraph* (MECBS) problem and proved that MECBS is not approximable within a sub-logarithmic factor. They also suggested a polynomial time approximation algorithm for a special case where the radio stations are distributed in the Euclidean space. The first logarithmic factor approximation algorithm for MECBS problem was proposed by Caragiannis et al. [34], where an interesting reduction to the *node-weighted connected dominating set problem* is used. This algorithm achieves a $10.8 \ln n$ factor approximation ratio for the symmetric instances of MECBS problem. Latter, Papadimitriou and Geordiadis [100] addressed the minimum energy broadcast problem where the broadcast tree is to be constructed in such a way that different source nodes can broadcast using the same broadcast tree, and the overall cost of the range assignment is minimum. This approach differs from the most commonly used one where the determination of the broadcast tree depends on the fixed source node. It is proved that, if the same broadcast tree is used, the total power consumed is at most twice the total power consumed for creating the broadcast tree with any node as the source. It is also proved that the total power consumed for this common broadcast tree is less than $2H(n-1) * opt$, where *opt* denotes the minimum cost of broadcast tree with this node as the source, and $H(n)$ is the harmonic function involving n.

Chlebikova et al. [46] and Kantor and Peleg [80] independently studied h-hop broadcast range assignment problem on an arbitrary graph where the weight of edges (transmission distances) can violate triangle inequality. By approximating edge weighted graph by paths, the authors presented a probabilistic $O(\log n)$ factor approximation algorithm, which matches with the lower bound proposed in [104]. If a graph does not contain a complete bipartite subgraph $K_{r,r}$ with $r > 2$, and $\beta \leq O(\frac{\log \log n}{\log \log \log n})$, then the approximation ratio can be improved to $O((\log \log n)^{\beta})$ [46], where β is the distance power gradient of the cost function. It also needs to be mentioned that Chlebikova et al. [46] presented an exact algorithm for h-hop broadcast, where the graph G is a tree. The running time of this algorithm is $O(hn^4)$.

For the 2-hop broadcast in the plane, the optimum range assignment can be obtained in $O(n^7)$ time using an algorithm based on dynamic programming paradigm [4]. In the same paper, a polynomial-time approximation scheme for the h-hop broadcast range assignment problem was suggested for any $h \geq 1$ and $\epsilon > 0$. The time complexity of the proposed algorithm is $O(n^{\alpha})$, where $\alpha = O((8h^2/\epsilon)^{2^h})$. Calinescu et al. [38] discussed 2-hop broadcast problem, where the range of a radio station is either a specified value ρ or 0. They presented a 6-factor approximation algorithm with running time $O(n \log n)$, and a 3-factor approximation algorithm with running time $O(n \log^2 n)$. Bronnimann and Goodrich [18] considered the *circle cover problem*, where a set S of n points is given in the plane, and a family of circles C is also given; the problem is to find a minimum number of circles in C that covers all the points in S. The circle cover problem can be easily mapped to the 2-hop broadcast problem. The proposed algorithm for the circle cover problem produces $O(1)$-approximation results in $O(n^3 \log n)$ time. Thus the algorithm proposed in [38] is an improvement over [18] in terms of both the time complexity results, and approximation factor. In this thesis, we present two algorithms in connection with 2-hop broadcast range assignment problem in 2D. These are (i) find the minimum cost homogeneous range assignment for 2-hop broadcast from a given source, and (ii) given a range ρ compute the minimum set of radio stations (if possible) whom range ρ is to be assigned for the 2-hop broadcast from the source radio station. For Problem (i), our proposed algorithm returns the optimum cost range assignment in $O(n^{2.376} \log n)$ time, and for the Problem (ii), we proposed a 2-approximation algorithm which runs in $O(n^2)$ time.

Although the unweighted broadcast range assignment problem has been studied extensively, little is known for the case of the weighted version. The first work on this problem for linearly arranged n radio stations was discussed in [5]. A number of variations of the problem have been studied, and the algorithms are proposed using dynamic programming. These are as follows:

- The unbounded case (i.e., $h = n - 1$), for which the time and space complexities of the proposed algorithm are $O(n^3)$ and $O(n^2)$ respectively;
- h-hop broadcast, for which the time and space complexities of the proposed algorithms are $O(hn^4)$ and $O(hn^2)$ respectively;
- The unbounded multi-source broadcast, where the time and space complexities are $O(n^6)$ and $O(n^2)$ respectively;
- In higher dimension (i.e., $d > 2$) and $\beta = 1$, the problem is formulated as a shortest path problem in a graph, and the proposed algorithm produces a 3-approximation result in $O(n^3)$ time;

For a detailed survey in the broadcast range assignment problem, see [29, 102].

In this thesis, we consider both the bounded and unbounded version of the weighted broadcast range assignment problem in 1D. Our proposed algorithm for the unbounded version of the problem output the optimum result in $O(n^2)$ time. The proposed algorithm for the bounded (h) hop broadcast problem produces the optimum solution in $O(hn^2 \log n)$ time.

1.3.2 All-to-all range assignment problem

The objective of the range assignment problem for h-hop all-to-all communication is to assign transmission range $\rho(s_i)$ to each radio station $s_i \in S$ so that each pair of members in S can communicate using at most h hops, and the total power consumption by the entire radio network is minimized. Typically, h can assume any value from 1 to $n-1$, where $n = |S|$. For $h = 1$, the problem is trivial. Here, for each radio station $s_i \in S$, $\rho(s_i) = Max_{s_j \in S} d(s_i, s_j)$. Basically, the hardness of the all-to-all range assignment problem depends on two parameters, namely, the distance power gradient (β in Equation 1.1) of the cost function and dimension (d) in which the radio stations are located. For the linear radio network ($d = 1$), the problem can be solved in polynomial time [76], but if $d > 1$, then the problem becomes NP-hard [40, 41, 53]. In particular, if $\beta = 1$, then the problem can be shown to be 1.5-APX hard [7]. For $\beta > 1$, the problem is APX-hard, and so it does not admit a PTAS unless P = NP [76].

For linear radio network, the problem becomes relatively simple, but it results in a more accurate analysis of the situation arising in vehicular technology application [90]. In a linear radio network, several variations of the 1D range assignment problem for h-hop all-to-all communication are studied by Kirousis et al. [76]. For the uniform chain case, i.e., where each pair of consecutive radio stations on the line is at a distance δ, tight upper bound on the minimum cost of range assignment is shown to be $OPT_h = \Theta(\delta^2 n^{\frac{2^{h+1}-1}{2^h-1}})$ for any fixed h. In particular, if $h = \Omega(\log n)$ in the uniform chain case, then $OPT_h = \Theta(\delta^2 \frac{n^2}{h})$. For the general problem in 1D, i.e., where the radio stations are arbitrarily placed on a line, a 2-approximation algorithm for the range assignment problem for h-hop all-to-all communication is proposed by Clementi et al. [39]. The worst case running time of this algorithm is $O(hn^3)$. For the unbounded case ($h = n-1$), a dynamic programming based $O(n^4)$ time algorithm is available [76] for generating the minimum cost. In this thesis, we proposed an improved algorithm for this problem. The running time of our algorithm is $O(n^3)$.

Carmi and Katz [32] proved that the all-to-all range assignment problem remains NP-hard when the range of each radio station is either ρ_1 or ρ_2 with $\rho_2 > \sqrt{\frac{3}{2}}\rho_1$. In the same paper, they also provided an $\frac{11}{6}$-approximation algorithm. Fuchs [53] studied the range assignment problem for all-to-all communication where the radio stations are well-spread [41] with $\beta > 0$. Under the assumption of symmetric connectivity as stated below, the problem is shown to be NP-hard in both 2D and 3D. It is also shown that the problem is APX-hard in 3D.

In the symmetric connectivity model, the minimum transmission power needed for a radio station s_i to reach a radio station s_j is assumed to be equal to the minimum transmission power needed for s_j to reach s_i. In other words, the symmetric connectivity means a link is established between two radio stations $s_i, s_j \in S$ only if both radio stations have transmission range at least as big as the distance between them.

Althaus et al. [6] presented an exact branch and cut algorithm based on an integer linear programming formulation for solving the unbounded (i.e., $h = n - 1$) version of the 2D all-to-all range assignment problem with symmetric connectivity assumption, and their algorithm takes 1 hour for solving instances with up to 35-40 nodes. In the same paper, a minimum spanning tree (MST) based 2-approximation algorithm has also been presented with symmetric connectivity assumption; here range of a radio station is equal to the length of the longest edge of the Euclidean MST attached with that radio station. Under the assumption of symmetric connectivity, Krumke et al. [78] presented an $(O(\log n), O(\log n))$ bicriteria approximation algorithm for h-hop all-to-all range assignment problem, i.e., their algorithm produces a solution having $O(h \log n)$ number of hops and costs at most $O(\log n)$ times the optimum solution. Latter, Calinescu et al. [36] studied the same problem independently, and provided an algorithm with same approximation result. Recently, Kucera [74] presented an algorithm for the all-to-all range assignment problem in 2D. Probabilistic analysis says that the average transmission power of the radio stations produced by this algorithm is almost surely constant if the radio stations appear in a square region of a fixed size. This algorithm can also work in any arbitrary dimension.

Santi et al. [115] studied the homogeneous (range of radio stations are equal) version of the 2D h-hop range assignment problem using a probabilistic approach, and established lower and upper bounds on the probability of connectedness of the communication graph. Under the assumption of asymmetric connectivity, Clementi et al. [40] presented a lower and an upper bound on the minimum cost of the h-hop range assignment of a radio network in 2D. They also proved that for the well-spread instances (defined in [41]) of this problem, these two bounds remain same.

Chlebikova et al. [46] studied the range assignment problem for h-hop all-to-all communication in static ad hoc networks using a graph-theoretic formulation where the edge weights can violate triangle inequality. They presented a probabilistic algorithm to approximate any edge-weighted graph by a collection of paths such that for any pair of nodes, the expected distortion of shortest path distance is at most $O(\log n)$. The paths in the collection and the corresponding probability distribution are obtained by solving a packing problem defined by Plotkin et al. [103], and using a *minimum linear arrangement problem* solver of Robinovich and Raz [108] as an oracle. With this algorithm, they approximated a 2D static ad hoc network as a collection of paths. Then it runs the polynomial time algorithm for the minimum range assignment problem in 1D [39]. Therefore, this strategy leads to a probabilistic $O(\log n)$ factor approximation algorithm for the h-hop all-to-all range assignment problem for the static ad hoc network in 2D. A polynomial time constant factor approximation algorithm for this problem on general metrics is given in [80]. The approximation ratio of the proposed algorithm is $(\frac{1}{\sqrt[h]{2}-1})^{\beta}(1+3^{\beta})(3^{\beta+1})^{h-2}$.

In [5], the weighted version of the all-to-all range assignment problem is studied only for q-spread instances in 2D, where the notion of *q-spread* instances in 2D is defined as follows:

Let s_i be a radio station in S. Consider a maximum size convex polygon containing only the radio station s_i and whose vertices are in the set $S \setminus \{s_i\}$. Let $H(s_i)$ be the vertices of this convex polygon. Now we define two quantities $\Delta(H(s_i)) = \max_{s_j \in H(s_i)}(d(s_i, s_j))^\beta$ and $\delta(H(s_i)) = \min_{s_j \in H(s_i)}(d(s_i, s_j))^\beta$. Finally, we choose $s_k \in S$ such that $\Delta(H(s_k)) = \max_{i=1}^{n} \Delta(H(s_i))$. The instance S is said to be q-spread if $\Delta(H(s_k)) \leq q \times \delta(H(s_k))$.

The proposed algorithm can work for any arbitrary distance power gradient $\beta > 1$, and produces a q-approximation result.

In this thesis, we propose an efficient heuristic for the h-hop all-to-all range assignment problem in 2D. The experimental evidences demonstrate that it produces near optimum result in reasonable time.

1.3.3 Accumulation range assignment problem

The objective of *h-hop accumulation range assignment problem* is to assign transmission range $\rho(s_i)$ to the radio station $s_i \in S$ such that each radio station $s_i \in S$ can send message to a dedicated radio station $s^* \in S$ using at most h hops and the total cost $\sum_{s_i \in S}(\rho(s_i))^2$ of the network is minimum.

Clementi et al. [39] discussed h-hop accumulation range assignment problem for 1D radio network and proposed an algorithm based on dynamic programming which can produce optimum solution in $O(hn^3)$ time. This algorithm can be used to design a trivial 2-approximation algorithm for the range assignment of h-hop all-to-all communication in a 1D radio network.

In a general graph, the h-hop accumulation range assignment problem is equivalent to finding the minimum spanning tree of height h with a designated node as the root. This problem is referred to as the h-MST problem in the literature. Experimentally tested exact super-polynomial time algorithms for the h-MST problem is already available [58, 62]. Althaus et al. [9] presented a polynomial time algorithm for this problem with running time $n^{O(h)}$. The 2-MST problem can be easily reduced to the classical *Uncapacitated Facility Location Problem* (UFLP). Thus all the approximation algorithms for UFLP apply to the 2-MST as well. As for the metric FLP, several polynomial time approximation algorithm based on linear programming (LP) relaxations have been presented in the literature [22, 60, 81, 94, 121]. The best known approximation factor is 3.16 due to Mahdian et al. [94]. Alfandari and Paschos [11] proved that metric 2-MST is MAX SNP-hard and hence PTAS cannot be found for this problem unless P = NP. As for the Euclidean case, the best result is a PTAS given by Arora et al. [12].

Several heuristic algorithms for the range assignment problem in 2D are available in the literature. Raidl and Julstrom [106] presented an evolutionary-based heuristic for the 2D Euclidean 2-MST problem and experimentally demonstrate that their algorithm performs better that the two existing greedy heuristics based on the classical Prim's algorithm for the MST problem [8, 47]. Clementi et al. [30] presented a fast and easy-to-implement heuristics for the 2D h-hop accumulation range assignment problem, and investigated its behavior on the instances obtained by choosing n points at random in a square region. They have also presented two simple heuristics based on Prim's and Kruskal's algorithms for the MST problem, and performed a comparative study among these three heuristics.

In connection with the accumulation range assignment problem, we have studied a little bit. We have proposed an algorithm for the unbounded case, which can produce the optimum solution in $O(n^2)$ time. It needs to be mentioned that, our algorithm can work in arbitrary dimension.

1.3.4 Unconstrained base station placement problem

The base station placement problem involves placing multiple base stations within a specific deployment site, with an aim to provide an acceptable quality of service to the mobile users. Here, the formulation of objective function depends on the hardware limitations of the specific wireless system and the particular application for which the system is to be designed.

Several authors [26, 54, 116, 120, 128, 132] studied the issues of optimal base station placement in an indoor micro-cellular radio environment with an aim to optimizing several objective criteria. Most of them have primarily used local optimization strategies for optimizing the desired objective function. Stamatelos and Ephremides [116] formulated the objective function as the maximization of coverage area along with the minimization of co-channel interference under the stipulated constraint of spatial diversity. Choong and Everitt [26] investigated the role of frequency-reuse across multiple floors in a building while solving the base station placement problem to minimize co-channel interference. Howitt and Ham [66] pointed out the limitations of using local optimization algorithms for solving the base station placement problem; finally they proposed a global optimization technique based algorithm, where the objective function is modeled as a stochastic process. The authors of [54, 128] indicated that the simplex method is well suited for the base station placement problem because the corresponding objective function is non-differentiable and so quasi-Newton optimization methods are not well-suited.

In this thesis, we consider the following problem: place a given number of base stations in a given convex region in 2D, and assign ranges to each of these base stations such that every point in the region is covered by at least one base station, and the maximum assigned range is minimized. We may assume that the range of all the base stations are same, say ρ. Since a base station with range ρ can communicate with all the mobile terminals present in the circular region of radius ρ and centered at the position where the base station is located, our problem reduces to the traditional *covering by circle problem*, available in the literature.

In the covering by circle problem, the following two variations are important: (i) find the minimum number of unit-radius circles that are necessary to cover a given polygon, and (ii) given a constant k, compute a radius ρ, such that an arrangement of k circles of radius ρ exists which can cover the entire polygon, and there does not exists any arrangement of k circles of a radius $\rho' < \rho$ which can cover the entire polygon. In this problem, we needs to report the centers of the k circles (of optimum radius) also.

Verblunsky [126] proposed a lower bound for the first problem; it says that if m is the minimum number of unit circles required for covering a square where each side is of length σ, then $\frac{3\sqrt{3}}{2}m > \sigma^2 + c\sigma$, where $c > \frac{1}{2}$. Substantial studies have been done on the second problem. Several researcher tried to cover a unit square region with a given number (say k) of equal radius circles with the objective to minimize the radius. Tarnai and Gasper [124] proposed graph theoretic approach to obtain a locally optimal covering of a square with up to 10 equal circles. No proof for optimality was given, but later it was observed that their solution for $k = 5$ and $k = 7$ are indeed optimal. The same idea was then extended by Heppes and Melissen [67] for covering a rectangle with up to 5 equal circles. Several results exist on covering squares and rectangles with k equal circles for small values of k ($= 6, \ldots 10$, etc.) [92, 93]. For a reasonably large value of k, the problem becomes more complex. Nurmela and Ostergard [97] adopted simulated annealing approach to obtain near-optimal solutions for the unit square covering problem for $k \leq 30$. As it is very difficult to get a good stoping criteria for a stochastic global optimization problem, they used heuristic approach to stop their program. It is mentioned that, for $k = 27$ their algorithm runs for about 2 weeks to achieve the stipulated stopping criteria. For $k \geq 28$, the time requirement is very high. So, they changed their stopping criteria, and presented the results. Nurmela [96] adopted the same approach for covering a equilateral triangle of unit edge length with circles of equal radius, and presented the results for different values of k less than or equal to 36. No results on the covering by circles problem are available where the region is an arbitrary simple polygon, or even for an arbitrary convex polygon.

The discrete version of the covering by circle problem is the well-known k-center problem. Here we need to place k supply points to cover a set of n demand points on the plane such that the maximum Euclidean distance of a demand point from its nearest supply point is minimized. The simplest form of this problem is the Euclidean 1-center problem which was originally proposed by Sylvester [114] in 1857. The first algorithmic result on this problem is due to Elzinga and Hearn [51], which gives an $O(n^2)$ time algorithm. Later, Shamos and Hoey [117] improved the time complexity of this problem to $O(n \log n)$. Lee [84] proposed the furthest point Voronoi diagram, which also can be used to solve the 1-center problem in $O(n \log n)$ time. Finally Megiddo [87] found an optimal $O(n)$ time algorithm for solving this problem using prune-and-search technique.

Jaromczyk and Kowaluk [73] studied the 2-center problem, and proposed a simple algorithm with running time $0(n^2 \log n)$. Later, Sharir [113] improved the time complexity to $O(n \log^9 n)$. The best known algorithm for this problem was proposed by Chan [21]. He suggests two algorithms. The first one is a deterministic algorithm, and it runs in $O(n \log^2 n (\log \log n)^2)$ time; the second one is a randomized algorithm that runs in $O(n \log^2 n)$ time with high probability.

A variation of this problem is the discrete 2-center problem, where the objective is to find two closed disks whose union can cover the given set P of n points, and whose centers are a pair of points in P. Kim and Shin [82] considered both the standard and discrete versions of the 2-center problem where the points to be covered are vertices of a convex polygon. Their algorithms run in $O(n \log^3 n \log \log n)$ and $O(n \log^2 n)$ time respectively.

For a given set of n demand points, the general version of both the k-center problem and the discrete k-center problem are NP-complete [56, 86]. But for a fixed value of k, both the problems can be solved in $O(n^{O(\sqrt{k})})$ time (see [65]). Therefore, it makes sense to search for efficient approximation algorithms and heuristics for the general version [69, 99]. Detailed review on this topic can be found in [119]. Another variation of this problem, available in the literature, is that the center and radius of the (equal-radius) circles are fixed and the objective is to cover the points in S with minimum number of circles. Stochastic formulations of different variations of this problem appeared in [15].

Apart from the base stations placement for mobile communication, the proposed problems find relevant applications in energy-aware strategic deployment of the sensor nodes in wireless sensor networks (WSN) [13, 17]. In particular, Boukerche et al. [17] studied the case where the sensor nodes are already placed. A distributed algorithm is proposed in that paper which can activate the sensors such that the entire area is always covered, and the total lifetime of the network is maximized. Voronoi diagram [19] is also an useful tool for dealing with the coverage problem for sensor networks, where the sensors are distributed in $I\!R^2$. Meguerdichian et al. [89] considered the problem where the objective is to find a sensor avoiding path between a pair of points s and t such that for any point p on the path, the distance of p from its closest sensor is maximized. Several other application specific covering problems related to sensor network are mentioned in [13].

1.3.5 Constrained base station placement problem

In general, every point inside the desired region may not be suitable for installing a base station. Some specific situations were mentioned in Section 1.2. In this thesis, we shall consider a constrained variation of the base station placement problem where the base stations can be erected only on the boundary of the given convex polygonal region. This problem can also be viewed as a constrained variation of k-center problem.

Several constrained versions of the Euclidean 1-center problem are studied in the literature. Megiddo [87] studied the case where the center of the smallest enclosing circle must lie on a given straight line. Bose and Toussaint [20] addressed a constrained variation of 1-center problem where instead of the entire region, a given set Q of m points is to be covered by a circle whose center is constrained to lie on the boundary of a given simple polygon P of size n. They provided an output sensitive $O((n+m)\log(n+m)+k)$ time algorithm for this problem, where k is the number of intersection points of the farthest-point Voronoi diagram of Q with the edges in P; this may be $O(nm)$ in the worst case. Constrained variations of the 1-center and 2-center problems are studied by Roy et al. [105], where the target region is a convex polygon with n vertices, and the center(s) of the covering circle(s) is/are constrained to lie on a specific edge of the polygon. The time complexities of both these problems are $O(n)$. Hurtado et al. [70] used linear programming to give an $O(n+m)$ time algorithm for solving minimum enclosing circle problem for a set of points whose center satisfies m linear inequality constraints. The query version of the minimum enclosing circle problem is studied by Roy et al. [107], where the given point set needs to be preprocessed such that given an arbitrary query line, the minimum enclosing circle with center on the query line can be reported efficiently. The preprocessing time and space of this algorithm are $O(n \log n)$ and $O(n)$ respectively, and the query time complexity is $O(\log^2 n)$.

Several other constrained variations of the k-center problem can be found in the domain of mobile communication and sensor network. Recently, Alt et al. [2] considered the problem of computing the centers of k circles on a line to cover a given set of points in 2D. The radius of the circles may not be the same. The objective is to minimize the sum of radii of all these k circles. They proposed an $O(n^2 \log n)$ time algorithm for solving this problem.

Sohn and Jo [118] considered a different variation of the problem. It assumes that two sets of points $B = \{b_1, b_2, \ldots, b_m\}$ and $R = \{r_1, r_2, \ldots, r_n\}$, called *blue* and *red* points, are given. The objective is to cover all the red points with circles of radius ρ (given a priori) centered at minimum number of blue points. Here the blue points indicate the possible positions of base stations, and red points indicate the target locations where the message need to be communicated. A heuristic algorithm using integer linear programming is presented along with experimental results. Azad and Chockalingam [3] studied a different variation where n base stations (of same range ρ) are placed on the boundary of a square region, and m sensors are uniformly distributed inside that region. The sensors are also allowed for limited movement. The entire time span is divided into slots. At the beginning of each time slot, depending on the positions of the sensors, k base stations need to be activated. The proposed algorithm finds a feasible solution (if exists) in time $O(mn + n \log n)$ time.

1.4 Scope of the Thesis

In this section, we summarize the list of problems considered in this thesis, and the results obtained on those problems.

In Chapter 2, we study the unweighted version of the broadcast range assignment problem in a linear radio network. Here, we assume that the members in the set S of n radio stations are located on a straight line, and one of them (say s^*) is designated as the source station. An integer h ($1 \leq h \leq n-1$) is also given. The objective is to assign ranges to the members in S so that s^* can send message to all other members in S using at most h hops, and the total power consumption (see Equation 1.4) is minimum. We propose an $O(n^2)$ time algorithm for this problem.

In Chapter 3, we consider the weighted version of the broadcast range assignment problem in a linear radio network. Here, the parameter γ of each radio station $s_i \in S$ may differ from that of the other members in S. Thus, here the cost function is given by Equation 1.3. Efficient algorithms have been designed for both the unbounded and bounded-hop broadcast range assignment problems in this environment. In the unbounded case ($h = n-1$), the proposed algorithm runs in $O(n^2)$ time, whereas for the h-hop broadcast, the time complexity is $O(hn^2 \log n)$.

In Chapter 4, we consider the broadcast range assignment problem in 2D. Here the members in the set S of n radio stations are placed in a 2D region, and a source radio station $s^* \in S$ is given. We consider the following two variations of minimum cost homogeneous range assignment problem for 2-hop broadcast from s^*: (i) find the value of ρ such that 2-hop homogeneous broadcast from s^* is possible with minimum cost, and (ii) given a real number ρ, check whether homogeneous 2-hop broadcast from s^* to all members in S is possible with range ρ, and if so, then identify the smallest subset of S, to which the range ρ is to be assigned to accomplish the broadcast. The first problem is solved in $O(n^{2.376} \log n)$ time and $O(n^2)$ space. The second problem seems to be hard. We present a 2-factor approximation algorithm for this problem, which runs in $O(n^2)$ time.

In Chapter 5, we consider the range assignment problem for all-to-all communication in linear radio network. We have considered only the unbounded version of the problem. As in Chapters 2 and 3, here also the radio stations in S are placed arbitrarily on a line. The objective is to assign ranges to these radio stations such that each of them can communicate with every other member in S, and the total power consumption of the entire network is minimized. Since, no restriction on the number of hops is imposed, communication may be done with at most $n-1$ hops. A simple incremental algorithm for this problem is proposed which produces the optimum solution in $O(n^3)$ time and $O(n^2)$ space.

In Chapter 6, we extend the all-to-all range assignment problem in 2D. As in Chapter 4, here the radio stations in S are distributed on a 2D plane, and an integer h is given. The objective is to assign range to each member in S such that each radio station in S can communicate with every other member in S using at most h hops, and the total power consumption of the entire network is minimum. The general 2D h-hop all-to-all range assignment problem is known to be NP-hard. We first consider the homogeneous version of the problem, where the range assigned to each radio station is same (say ρ), and the objective is to compute the minimum feasible value of ρ for the all-to-all communication.

We propose an algorithm for this problem which needs $O(n^3(\frac{\log \log n}{\log n})^{\frac{5}{4}}\log n)$ time in the worst case. In addition, if we consider the unbounded version of the homogeneous range assignment problem, then the minimum feasible value of ρ can be obtained in $O(n^2\log n)$ time. Finally, we propose an efficient heuristic algorithm for the general h-hop all-to-all range assignment problem in 2D, where the range of the radio stations may not be equal. Experimental results demonstrate that our heuristic algorithm runs fast and produces near-optimal solutions on randomly generated instances.

In Chapter 7, we consider the base station placement problem in the context of mobile communication. The unconstrained version of the problem is studied first in a restricted setup, where the 2D region under consideration is a convex polygon. The objective is to place a given number of base stations inside a convex region, and to assign range to each of them such that every point in the region is covered by at least one base station, and the maximum range assigned is minimum. Existing results for this problem are known for the case where the region is a square or an equilateral triangle. The minimum radius obtained by our method favorably compares with the existing results. The execution time of our algorithm is a fraction of a second in a SUN Blade 1000 computing platform with 750 MHz CPU speed, whereas the existing methods may even take about two weeks' time for a reasonable value of the number of circles (≥ 27), as reported in [97].

In Chapter 8, we study the constrained version of the base station placement problem, where the base station can be placed only on the boundary of the given convex region P. Here the objective is to determine the positions of k base stations (of equal range) on the boundary of P such that each point inside P is covered by at least one base station. We name this problem as *region-cover*(k) problem. A simplified form of this problem is the *vertex-cover*(k) problem, where the objective is to establish communication among only the vertices of P instead of covering the entire polygon. This problem is also useful in some specified applications as mentioned in Chapter 8. We first present efficient algorithms for *vertex-cover(2)* and *region-cover(2)* problems, where the base stations are to be installed on a pair of specified edges. The time complexities of these algorithms are $O(n \log n)$ and $O(n^2)$ respectively. Next, we consider the case where $k \geq 3$. We first concentrate on the restricted version of the *vertex-cover(k)* and *region-cover(k)* problems, where all the k base stations are to be placed on the same edge of P. Our proposed algorithm for the restricted *vertex-cover(k)* problem produces the optimum result in $O(min(n^2, nk \log n))$ time, whereas the algorithm for the restricted *region-cover(k)* problem produces an $(1+\epsilon)$-factor approximation result in $O((n+k)\log(n+k) + n\log(\lceil\frac{1}{\epsilon}\rceil))$ time. Finally, we propose an efficient heuristic algorithm for the general *region-cover(k)* problem, for $k \geq 3$. Experimental results demonstrate that our proposed algorithm runs fast and produces near-optimum solutions.

Finally, the concluding remarks on our studies in this thesis appear in Chapter 9. Here, once again we discuss our proposed results on different problems along with their possible extensions.

Chapter 2

Broadcast in Linear Radio Networks

2.1 Introduction

In this chapter, we study the *h-hop broadcast range assignment problem* in linear radio network. Here a set S of n radio stations are placed on a straight line, a source node s^* ($\in S$), and an integer h ($1 \leq h \leq n-1$) are given. The objective is to assign range $\rho(s)$ to each radio station $s \in S$ such that s^* can transmit message to each member in S using at most h hops, and the total power requirement for all the members in S ($\sum_{s \in S}(\rho(s))^2$) is minimum. We propose an $O(n^2)$ time algorithm for this problem. The earlier result on this problem was $O(hn^2)$ [31]. Thus, we have an improvement over the existing result by a factor of h.

2.2 Preliminaries

We assume that the radio stations $S = \{s_1, s_2, \ldots, s_n\}$ are ordered on the x-axis from left to right, with s_1 positioned at 0 (the origin). The position of s_i will be denoted by $x(s_i)$.

Thus, the distance between two radio stations s_i and s_j is $d(s_i, s_j) = |x(s_i) - x(s_j)|$. Let $\mathcal{R} = \{\rho(s_1), \rho(s_2), \ldots, \rho(s_n)\}$ be a range assignment, where $\rho(s_i)$ is the range assigned to s_i. The cost of this range assignment $\mathcal{R}$ is equal to $\sum_{s_i \in S}(\rho(s_i))^2$ and denoted by $cost(\mathcal{R})$. As mentioned earlier in Section 1.1, the directed graph $G = (V, E)$ with $V = S$ and $E = \{(s_i, s_j) | d(s_i, s_j) \leq \rho(s_i)\}$ is referred to as the *communication graph* for the range assignment $\mathcal{R}$.

For each radio station $s_i \in S$, let D_i be an array of size n which contains the distances $\{d(s_i, s_j), j = 1, \ldots, n\}$ in sorted order. Note that, $\min(D_i) = \min_{j=1}^{n} D_i[j] = 0$, because $d(s_i, s_j) = 0$ if $i = j$ and $d(s_i, s_j) > 0$ if $i \neq j$. Now we have the following lemma.

Lemma 2.1 *For any given h, if $\mathcal{R} = \{\rho_1, \rho_2, \ldots, \rho_n\}$ denotes the optimum range assignment of $\{s_1, s_2, \ldots, s_n\}$ for h-hop broadcast, then $\rho_i \in D_i$ for all $i = 1, 2, \ldots, n$.*

Proof: Let us assume that $\rho_i = r$ for some i, and $r \notin D_i$. Let $G_{\mathcal{R}}$ be the corresponding communication graph. Clearly, $r > \min(D_i)$, since $r \notin D_i$ and $\min(D_i) = 0$. Now we consider two cases: (i) $r > \max(D_i)$ and (ii) there exist a pair of consecutive elements $a, b \in D_i$ such that $a < r < b$.

Consider a different range assignment $\mathcal{R}' = \{\rho_1, \rho_2, \ldots, \rho_{i-1}, r^*, \rho_{i+1}, \ldots, \rho_n\}$, where $r^* = \max(D_i)$ in Case (i), and $r^* = a$ in Case (ii). In both the cases, the communication graph G corresponding to $\mathcal{R}'$ remains same as it was for the range assignment $\mathcal{R}$, because this change in the range assignment does not delete any edge from G. Thus, the desired connectivity of each vertex in S to all other vertices is maintained for the range assignment $\mathcal{R}'$. Again, $cost(\mathcal{R}') = cost(\mathcal{R}) - r^2 + (r^*)^2 < cost(\mathcal{R})$. Hence we have the contradiction that $\mathcal{R}$ is the optimum range assignment. $\square$

Note: The result stated in Lemma 2.1 is valid if these range assignment problems are considered in any arbitrary dimension. It will be used to justify the correctness of the proposed algorithms for different problems considered in this thesis.

We will use $\mathcal{C}(S, s^*, h)$ to denote the minimum among the costs of the range assignments of the members in S for broadcasting message from the source radio station s^* ($\in S$) to all other radio stations in S using at most h hops. There may be several range assignments of S having cost $\mathcal{C}(S, s^*, h)$. We will use $\mathcal{R}(S, s^*, h)$ to denote one such range assignment, and will refer it as *optimal range assignment*.

Definition 2.1 *In the communication graph G corresponding to a h-hop broadcast range assignment, an edge $e = (s_i, s_j)$ is said to be* functional *if the removal of this edge indicates that there exists a radio station $s_k \in S$ which is not reachable from s^* (source) using a h-hop path.*

Definition 2.2 *In a h-hop broadcast range assignment, a* right-bridge $\overleftarrow{s_\ell s_r}$ *corresponds to a pair of radio stations (s_ℓ, s_r) such that s_ℓ is to the left of s^*, s_r is to the right of s^*, and s_r can communicate with s_ℓ in 1-hop due to its assigned range, but it can not communicate with $s_{\ell-1}$ in 1-hop.*

A right-bridge $\overleftarrow{s_\ell s_r}$ (if exists) is said to be a *functional right-bridge* if the edge (s_r, s_ℓ) is functional (see Definition 2.1) in the communication graph G. Similarly, one can define a *left-bridge* $\overrightarrow{s_\ell s_r}$ and a *functional left-bridge* in a h-hop range assignment, where s_ℓ and s_r are respectively to the left and right sides of s^*.

Theorem 2.1 *[31] Given a set of radio stations $S = \{s_1, s_2, \ldots, s_n\}$, a source node $s^* \in S$, and an integer h $(1 \leq h \leq n-1)$, the optimal h-hop broadcast range assignment $\mathcal{R}(S, s^*, h)$ contains at most one functional bridge.*

The algorithm proposed in [31] solves the problem in three phases. It computes optimal solutions having (i) no functional (left/right) bridge, (ii) one functional left-bridge only, and (iii) one functional right-bridge only. Finally, the one having minimum total cost is reported. Our algorithm is based on the same principle as in [31], but it considers the inherent geometry of the range assignment problem for obtaining the optimal solution in each of the three cases mentioned in (i)-(iii) in a careful manner; this leads to an algorithm with improved time complexity.

2.3 Geometric properties

Lemma 2.2 *In a linearly ordered set of radio stations* $\{s_a, s_{a+1}, \ldots, s_b\} \subseteq S$*, if the source station* s^* *is at one end of the above set (i.e.,* $s^* = s_a$*), then for any* $1 \leq \mu \leq b-a$*, an optimum* μ*-hop broadcast range assignment* $\mathcal{R}(\{s_a, s_{a+1}, \ldots, s_b\}, s_a, \mu)$ *should satisfy* $\sum_{k=a}^{b-1} \rho(s_k) = x(s_b) - x(s_a)$*.*

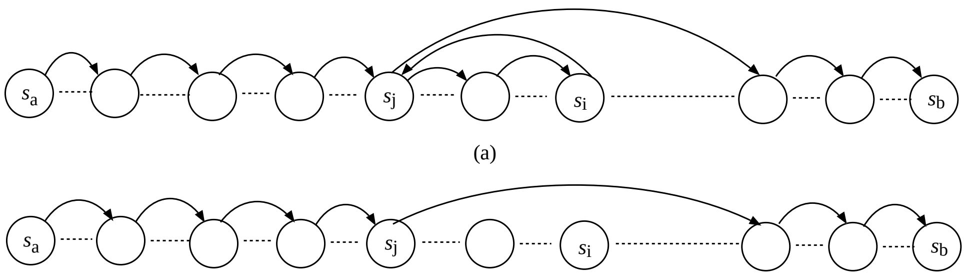


Figure 2.1: Proof of Lemma 2.2

Proof: Consider the μ-hop path for communication from s_a to s_b as shown in Figure 2.1(a). Note that, one can reduce the total cost of range assignment $(\sum_{k=a}^{b-1}(\rho(s_k))^2)$ by setting $\rho(s_i) = 0$ (see Figure 2.1(b)). This maintains μ-hop connections from s_a to all other members in the set S. □

Lemma 2.3 *For a set of radio stations* $S = \{s_1, s_2, \ldots, s_n\}$*,* $\mathcal{C}(S, s_1, \mu) = \mathcal{C}(S, s_n, \mu)$*.*

Proof: Let $\{a_0, a_1, \ldots, a_{\mu-1}\} \subseteq S$ be the sequence of radio stations having non-zero ranges in $\mathcal{R}(S, s_1, \mu)$. Here $a_0 = s_1$, and let us denote $a_\mu = s_n$. By Lemma 2.2, $\rho(a_i) =$

$x(a_{i+1}) - x(a_i)$, for $i = 0, 1, \ldots, \mu - 1$. A feasible range assignment for communicating from s_n to s_1 using at most μ hops is $\rho(a_i) = x(a_i) - x(a_{i-1})$, for $i = 1, 2, \ldots, \mu$, and its cost is same as $\mathcal{C}(S, s_1, \mu)$. Thus $\mathcal{C}(S, s_n, \mu) \leq \mathcal{C}(S, s_1, \mu)$. Following the same method, we can prove that $\mathcal{C}(S, s_1, \mu) \leq \mathcal{C}(S, s_n, \mu)$. Hence the result follows. $\square$

Lemma 2.4 *In an optimum μ-hop broadcast range assignment $\mathcal{R}(S, s_1, \mu)$, if the range assigned to s_1 is $\rho(s_1) = d(s_1, s_j)$ for some $j > 1$, then there exists a μ-hop broadcast range assignment $\mathcal{R}(S \backslash \{s_1\}, s_2, \mu)$ for broadcasting from s_2, where $\rho(s_2) \geq d(s_2, s_j)$.*

Proof: In $\mathcal{R}(S, s_1, \mu)$, $\rho(s_1) = d(s_1, s_j)$ implies that $\rho(s_2) = \rho(s_3) = \ldots = \rho(s_{j-1}) = 0$. Thus, if $\mathcal{C}(S, s_1, \mu) = c$ then $\mathcal{C}(S \setminus \{s_1, s_2, \ldots, s_{j-1}\}, s_j, \mu - 1) = c - (d(s_1, s_j))^2$. In other words, the range assignments of the radio stations $S \setminus \{s_1, s_2, \ldots, s_{j-1}\}$ in $\mathcal{R}(S, s_1, \mu)$ are such that, it supports broadcasting from s_j to all the radio stations $\{s_{j+1}, \ldots, s_n\}$ in $(\mu - 1)$ hops with minimum cost.

Now, let us assume that the range assigned to s_2 in $\mathcal{R}(S \backslash \{s_1\}, s_2, \mu)$ is $\rho(s_2) = d(s_2, s_k)$. We need to prove that $k \geq j$.

Let us assume $\mathcal{C}(S \setminus \{s_1\}, s_2, \mu) = c'$. This implies, $\mathcal{C}(S \setminus \{s_1, s_2, \ldots, s_{k-1}\}, s_k, \mu - 1) = c' - (d(s_2, s_k))^2$. Thus, $\{d(s_1, s_k), \underbrace{0, 0, \ldots, 0}_{k-2}, \mathcal{R}(S \setminus \{s_1, s_2, \ldots, s_{k-1}\}, s_k, \mu - 1)\}$ is a feasible range assignment (may not be optimum) for the μ-hop broadcast from s_1 to all the nodes in $S \setminus \{s_1\}$, and its cost is equal to $(d(s_1, s_k))^2 + (c' - (d(s_2, s_k))^2) \geq c$. This implies, $c - c' \leq (d(s_1, s_2))^2 + 2d(s_1, s_2)d(s_2, s_k)$.

By a similar argument, $\{d(s_2, s_j), \underbrace{0, 0, \ldots, 0}_{j-3}, \mathcal{R}(S \setminus \{s_1, s_2, \ldots, s_{j-1}\}, s_j, \mu - 1)\}$ is a feasible range assignment for the μ-hop broadcast from s_2 to the members in $S \backslash \{s_1, s_2\}$, and its cost is equal to $(d(s_2, s_j))^2 + (c - (d(s_1, s_j))^2) \geq c'$. This implies, $c - c' \geq (d(s_1, s_2))^2 + 2d(s_1, s_2)d(s_2, s_j)$.

Combining these two inequalities, we have

$$(d(s_1, s_2))^2 + 2d(s_1, s_2)d(s_2, s_j) \leq c - c' \leq (d(s_1, s_2))^2 + 2d(s_1, s_2)d(s_2, s_k).$$

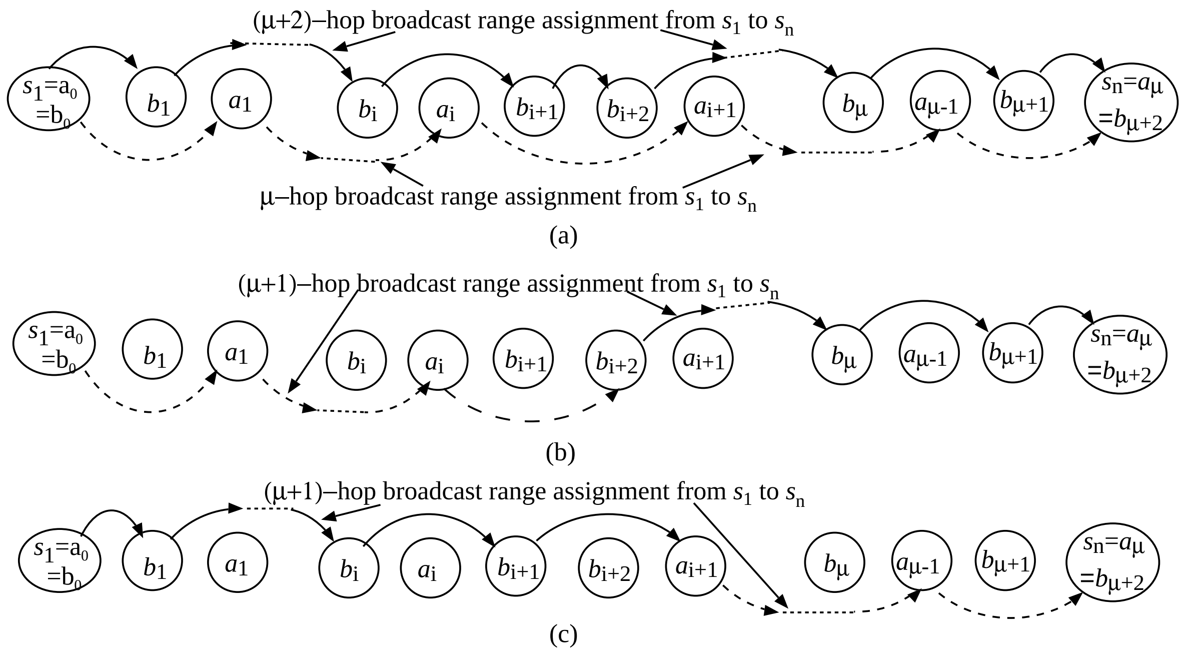


Figure 2.2: Proof of Lemma 2.5

This implies $k \geq j$. □

In the following lemma, we prove that if we increase the number of allowable hops for broadcasting from a fixed radio station, say s_1, to all the radio stations to its right, then the gain in the cost obtained in two consecutive steps are monotonically decreasing.

Lemma 2.5 $\mathcal{C}(S, s_1, \mu) - \mathcal{C}(S, s_1, \mu+1) \geq \mathcal{C}(S, s_1, \mu+1) - \mathcal{C}(S, s_1, \mu+2)$.

Proof: Let $A = \{a_0 = s_1, a_1, a_2, \ldots a_{\mu-1}\}$ denote the subsequence (radio stations) of S having non-zero ranges in $\mathcal{R}(S, s_1, \mu)$. We use a_μ to denote the radio station s_n and $cost(A)$ to denote $\mathcal{C}(S, s_1, \mu)$. Here, the range assigned to $a_i \in A$ is $(x(a_{i+1}) - x(a_i))$ for $i = 0, 1, 2, \ldots, \mu - 1$. Again, let $B = \{b_0 = s_1, b_1, b_2, \ldots b_{\mu+1}\}$ denote the set of radio stations having non-zero ranges in $\mathcal{R}(S, s_1, \mu+2)$, i.e., $cost(B) = \mathcal{C}(S, s_1, \mu+2)$. As earlier, s_n is denoted by $b_{\mu+2}$, and the ranges assigned to $b_i (\in B)$ are $(x(b_{i+1}) - x(b_i))$ for $i = 0, 1, 2, \ldots, \mu+1$. The two range assignments (A and B) are shown in Figure 2.2(a) using solid and dashed lines. Observe that, $x(a_0) - x(b_1) < 0$, and $x(a_\mu) - x(b_{\mu+1}) > 0$. This implies, there exists at least one $i \in \{0, 1, \ldots, \mu - 1\}$ such that $x(a_i) - x(b_{i+1}) \leq 0$ and $x(a_{i+1}) - x(b_{i+2}) \geq 0$. We consider the smallest $i \geq 0$ such that $x(a_{i+1}) - x(b_{i+2}) \geq 0$, and construct two subsequences of radio stations $C = \{a_0 = b_0 = s_1, a_1, \ldots, a_i, b_{i+2}, b_{i+3}, \ldots, b_{\mu+1}\}$ and $D = \{a_0 = b_0 = s_1, b_1, b_2, \ldots, b_{i+1}, a_{i+1}, a_{i+2}, \ldots, a_{\mu-1}\}$, each of length $\mu + 1$. The ranges assigned to the members in C and D are respectively

- $\{(x(a_1) - x(a_0)), \ldots, (x(a_i) - x(a_{i-1})), (x(b_{i+2}) - x(a_i)), (x(b_{i+3}) - x(b_{i+2})), \ldots, (x(b_{\mu+2}) - x(b_{\mu+1}))\}$ (see Figure 2.2(b)), and
- $\{(x(b_1) - x(b_0)), \ldots, (x(b_{i+1}) - x(b_i)), (x(a_{i+1}) - x(b_{i+1})), (x(a_{i+2}) - x(a_{i+1})), \ldots, (x(a_\mu) - x(a_{\mu-1}))\}$ (see Figure 2.2(c)).

The corresponding costs of the range assignments are

$cost(C) = \sum_{j=0}^{j=i-1}(x(a_{j+1}) - x(a_j))^2 + (x(b_{i+2}) - x(a_i))^2 + \sum_{j=i+2}^{j=\mu+1}(x(b_{j+1}) - x(b_j))^2$, and

$cost(D) = \sum_{j=0}^{j=i}(x(b_{j+1}) - x(b_j))^2 + (x(a_{i+1}) - x(b_{i+1}))^2 + \sum_{j=i+1}^{j=\mu-1}(x(a_{j+1}) - x(a_j))^2$.

Thus, $cost(C) + cost(D)$
$= (\sum_{j=0}^{j=\mu-1}(x(a_{j+1}) - x(a_j))^2 - (x(a_{i+1}) - x(a_i))^2) + (\sum_{j=0}^{j=\mu+1}(x(b_{j+1}) - x(b_j))^2 - (x(b_{i+2}) - x(b_{i+1}))^2) + (x(b_{i+2}) - x(a_i))^2 + (x(a_{i+1}) - x(b_{i+1}))^2$.

$= cost(A) + cost(B) + 2(x(a_i) - x(b_{i+1}))(x(a_{i+1}) - x(b_{i+2}))$
$\leq cost(A) + cost(B)$ (since $(x(a_i) - x(b_{i+1})) \leq 0$ and $(x(a_{i+1}) - x(b_{i+2})) \geq 0$ by the choice of i as mentioned above).

Note that, $\mathcal{C}(S, s_1, \mu + 1)$ is the cost of the *minimum cost range assignment* for sending message from s_1 to s_n in $\mu + 1$ hops (or equivalently to all members in S in at most $\mu + 1$ hops). Thus we have, $2 \times \mathcal{C}(S, s_1, \mu + 1) \leq cost(C) + cost(D) \leq cost(A) + cost(B)$.
$\square$

2.4 Algorithm

Let $s_\alpha \in S$ be the given source radio station (not necessarily the left-most/right-most in the ordering of S), i.e., $s^* = s_\alpha$. Our algorithm for broadcasting from s_α to all other radio stations $s_j \in S$ consists of three phases. Phase 1 prepares four initial matrices. These are used in Phases 2 and 3 for computing optimal solution with no functional bridge, and exactly one functional bridge respectively.

For notational convenience, if the source radio station s_α is at one end of a linearly ordered destination radio stations $\{s_a, s_{a+1}, \ldots, s_b\}$ (i.e., $s_\alpha = s_a$), then we will use $R(s_b, s_a, \mu)$ and $\mathcal{C}(s_b, s_a, \mu)$ to denote the optimal range assignment $\mathcal{R}(\{s_a, s_{a+1}, \ldots, s_b\}, s_a, \mu)$ and the corresponding cost $\mathcal{C}(\{s_a, s_{a+1}, \ldots, s_b\}, s_a, \mu)$ respectively.

2.4.1 Phase 1

In this phase, we prepare the following four initial matrices. These will be extensively used in Phases 2 and 3. Recall that s_α is the source radio station.

M_1: It is a $h \times (\alpha - 1)$ matrix. Its (m, j)-th element $(1 \leq j < \alpha)$ indicates the optimum cost of sending message from s_j to s_α (source radio station) using at most m hops. In other words, $M_1[m, j] = \mathcal{C}(s_\alpha, s_j, m)$, where $1 \leq m \leq h$ and $1 \leq j < \alpha$.

M_2: It is a $h \times (\alpha - 1)$ matrix. Its (m, j)-th element $(1 < j \leq \alpha)$ indicates the optimum cost of sending message from s_j to s_1 (left-most radio station in S) using at most m hops. In other words, $M_2[m, j] = \mathcal{C}(s_1, s_j, m)$, where $1 \leq m \leq h$ and $1 < j \leq \alpha$.

M_3: It is a $h \times (n - \alpha)$ matrix. Its (m, j)-th element $(\alpha < j \leq n)$ indicates the optimum cost of sending message from s_j to s_α using at most m hops. In other words, $M_3[m, j] = \mathcal{C}(s_\alpha, s_j, m)$, where $1 \leq m \leq h$ and $\alpha < j \leq n$.

M_4: It is a $h \times (n-\alpha)$ matrix. Its (m, j)-th element $(\alpha \leq j < n)$ indicates the optimum cost of sending message from s_j to s_n (right-most radio station in S) using at most m hops. In other words, $M_4[m, j] = \mathcal{C}(s_n, s_j, m)$, where $1 \leq m \leq h$ and $\alpha \leq j < n$.

Note that, the columns of M_1 are indexed as $[1, 2, \ldots, \alpha - 1]$, whereas those in M_2 are indexed as $[2, 3, \ldots, \alpha]$. Similarly, the columns of M_3 are indexed as $[\alpha + 1, \alpha + 2, \ldots, n]$, whereas those in M_4 are indexed as $[\alpha, \alpha + 1, \ldots, n-1]$. We explain an incremental approach (in terms of hops) for constructing M_1. Similar procedure works for constructing the other three matrices.

Each entry of the matrix M_1 contains a tuple (χ, ptr), where $M_1[m, j].\chi$ contains $\mathcal{C}(s_\alpha, s_j, m)$, and its $M_1[m, j].ptr$ contains the index of the first radio station (after s_j) on the m-hop path from s_j to s_α. We will interchangeably use, $M_1[m, j]$ and $M_1[m, j].\chi$ to denote $\mathcal{C}(s_\alpha, s_j, m)$. After computing up to row m of the matrix M_1, the elements in the $(m + 1)$-th row can easily be obtained as follows:

Consider an intermediate matrix A of size $(\alpha - 1) \times (\alpha - 1)$. Its (j, k)-th element contains the cost of $(m+1)$-hop communication from s_j to s_α with first hop at s_k. Thus, $A[j, k] = (d(s_j, s_k))^2 + M_1[m, k]$. After computing the matrix A, we compute $M_1[m + 1, j].\chi = \min_{k=j+1}^{\alpha-1} A[j, k]$, and $M_1[m + 1, j].ptr$ = the index k for which $A[j, k]$ is contributed to $M_1[m + 1, j].\chi$.

Straight forward application of the above method needs $O(\alpha^2)$ time. For each entry $M_1[m+1, j]$, it needs computation of the minimum value in the j-th row of the matrix A. But, Lemma 2.4 says that, if in the optimum $(m+1)$-hop path from s_j to s_α, the first hop is at node s_k, then for any node $s_{j'}$ with $j' > j$, we have the optimum $(m+1)$-hop path from s'_j to s_α with first hop at some node $s_{k'}$, where $k' \geq k$. A simple method for computing the minimum of every row in the matrix A (without enumerating all the entries in A) needs a total of $O(\alpha \log \alpha)$ time as follows:

Compute all the entries in the $\frac{\alpha}{2}$-th row of the matrix A, and find the minimum. Let it corresponds to $A[\frac{\alpha}{2}, \beta]$. Next, compute the minimum entry in $\frac{\alpha}{4}$-th row of A by considering $\{A[\frac{\alpha}{4}, j], j = 1, 2, \ldots, \beta\}$, and compute the minimum entry in $\frac{3\alpha}{4}$-th row of A by considering $\{A[\frac{3\alpha}{4}, j], j = \beta, \beta+1, \ldots, \alpha-1\}$. The process continues until all the rows of A are considered.

Definition 2.3 *[10] A matrix M is said to be a* monotone matrix *if for every j, k, j', k' with $j < j'$, $k < k'$, if $M[j, k] \geq M[j, k']$ then $M[j', k] \geq M[j', k']$.*

Lemma 2.6 *The matrix A is a monotone matrix.*

Proof: Given $A[j, k] \geq A[j, k']$, where $A[j, k] = (d(s_j, s_k))^2 + M_1[m, k]$ and $A[j, k'] = (d(s_j, s_{k'}))^2 + M_1[m, k']$. Thus, $M_1[m, k] - M_1[m, k'] \geq (d(s_j, s_{k'}))^2 - (d(s_j, s_k))^2$.

Now, $A[j', k] - A[j', k'] = (d(s_{j'}, s_k))^2 - (d(s_{j'}, s_{k'}))^2 + M_1[m, k] - M_1[m, k']$
$\geq (d(s_{j'}, s_k))^2 - (d(s_{j'}, s_{k'}))^2 + (d(s_j, s_{k'}))^2 - (d(s_j, s_k))^2 \geq 0$ (on simplification). □

A recursive algorithm for monotone matrix searching is described in [10], which can compute the minimum entry in each row of a $\alpha \times \alpha$ monotone matrix in $O(\alpha)$ time provided each entry of the matrix can be computed in $O(1)$ time. Using that algorithm, the matrix M_1 can be computed in $O(\alpha h)$ time.

Lemma 2.7 *Phase 1 needs $O(nh)$ time.*

Proof: Follows from the fact that M_1, M_2 can be constructed in $O(\alpha h)$ time, and M_3, M_4 needs $O((n - \alpha)h)$ time. □

Lemma 2.8 *If the source station $s_\alpha = s_1$ or s_n, then the h-hop broadcast range assignment can be performed in $O(nh)$ time.*

Proof: The optimum cost of the broadcast range assignment with $s_\alpha = s_1$ can be obtained from $M_4[h, 1].\chi$. The corresponding range assignments of the radio stations are obtained as follows:

Set $i = 1$ and $\mu = h$.

While $i < n$ execute the following steps:

- Set $j = M[\mu, i].ptr$, and $\rho(s_i) = d(s_i, s_j)$
- Set $i = j$ and $\mu = \mu - 1$.

Computing the matrix M_4 needs $O(nh)$ time (see Lemma 2.7). The range assignments can be done in $O(n)$ time. The case for $s_\alpha = s_n$ can be solved in a similar way consulting the matrix M_3. $\square$

2.4.2 Phase 2

In this phase, we compute the optimal functional bridge-free solution for broadcasting message from s_α to the other nodes in S. Here, the range to be assigned to s_α is at least $\max(d(s_\alpha, s_{\alpha-1}), d(s_\alpha, s_{\alpha+1}))$ as there is no functional left/right bridges in this case.

Without loss of generality, we assume that $d(s_\alpha, s_{\alpha-1}) \leq d(s_\alpha, s_{\alpha+1})$. Thus, $\rho(s_\alpha)$ is initially assigned to $d(s_\alpha, s_{\alpha+1})$, and let s_k ($k < \alpha$) be the farthest radio station such that s_α can communicate with s_k in 1-hop (i.e., $d(s_k, s_\alpha) \leq d(s_\alpha, s_{\alpha+1})) < d(s_{k-1}, s_\alpha)$). If we use $\mathcal{R}(S, s_\alpha, h|\rho(s_\alpha) = \delta)$ to denote the optimum range assignment for the h-hop broadcast from s_α to all the nodes in S subject to the condition that the range assigned to s_α is δ, then

$$\mathcal{R}(S, s_\alpha, h|\rho(s_\alpha) = d(s_\alpha, s_{\alpha+1}))$$
$$= \{\mathcal{R}(\{s_1, \ldots, s_k\}, s_k, h-1), \underbrace{0, 0, \ldots, 0}_{\alpha-k-1}, d(s_\alpha, s_{\alpha+1}), \mathcal{R}(S \setminus \{s_1, \ldots, s_\alpha\}, s_{\alpha+1}, h-1)\},$$
$$= \{R(s_1, s_k, h-1), \underbrace{0, 0, \ldots, 0}_{\alpha-k-1}, d(s_\alpha, s_{\alpha+1}), R(s_n, s_{\alpha+1}, h-1)\}$$

and its cost is

$$\mathcal{C}^* = \mathcal{C}(S, s_\alpha, h|\rho(s_\alpha) = d(s_\alpha, s_{\alpha+1})) = (d(s_\alpha, s_{\alpha+1}))^2 + M_2[h-1, k] + M_4[h-1, \alpha+1].$$

This can be computed in $O(1)$ time using the matrices M_2 and M_4. We use two temporary variables TEMP_Cost and TEMP_id to store $\mathcal{C}^*$ and $s_{\alpha+1}$.

Next, we increment $\rho(s_\alpha)$ to $\min(d(s_\alpha, s_{k-1}), d(s_\alpha, s_{\alpha+2}))$, and apply the same procedure to calculate the optimum cost of the h-hop broadcast from s_α. This may cause update of TEMP_Cost and TEMP_id. The same procedure is repeated by incrementing $\rho(s_\alpha)$ to its next choice in the set $\{d(s_\alpha, s_k), k = 1, 2, \ldots, \alpha - 1\} \bigcup \{d(s_\alpha, s_j), j = k+1, \ldots, n\}$ so that it can communicate directly with one more node than its previous choice. At each step, the TEMP_Cost and TEMP_id are adequately updated. The procedure is repeated for $O(n)$ times, and the time complexity of this phase is $O(n)$. The stepwise description of the algorithm is given below.

Step 1: Compute the matrices M_2 and M_4 using the method described in Phase 1.

Step 2: Initialize TEMP_Cost and TEMP_id by ∞ and $NULL$ respectively. Set $i = \alpha - 1$ and $j = \alpha + 1$.

Step 3: $\rho(s_\alpha) = \max(d(s_\alpha, s_i), d(s_\alpha, s_j))$.

Step 4: If $\rho(s_\alpha)$ corresponds to i, **then**

if m is the maximum index such that $d(s_\alpha, s_m) \leq \rho(s_\alpha)$, then set $j = m$, $count = j - i + 1$, and TEMP_id $= s_i$;

else (* If $\rho(s_\alpha)$ does not correspond to i *)

if m be the minimum index such that $d(s_\alpha, s_m) \leq \rho(s_\alpha)$, then set $i = m$, $count = j - i + 1$, and TEMP_id $= s_j$.

Step 5: while ($count \leq n$) perform the following steps:

Step 5.1: $cost = (\rho(s_\alpha))^2 + M_2[h-1, i].\chi + M_4[h-1, j].\chi$.

Step 5.2: If $d(s_\alpha, s_{i-1}) < d(s_\alpha, s_{j+1})$, then $\rho(s_\alpha) = d(s_\alpha, s_{i-1})$, $i = i - 1$;

else $\rho(s_\alpha) = d(s_\alpha, s_{j+1})$, $j = j + 1$

Step 5.3: $count = count + 1$

Step 6: (* Range assignment *) Thus, *TEMP_Cost* is the cost of optimum solution having no functional bridge, and range of s_α is $\rho(s_\alpha) = d(s_\alpha, TEMP\_id)$. The range of the other radio stations are computed as follows:

Let $\rho(s_\alpha) = d(s_\alpha, s_j)$, $j < \alpha$ and $s_{j'}$ ($j' > \alpha$) be the right-most radio station such that $d(s_\alpha, s_{j'}) \leq \rho(s_\alpha)$, or in other words, s_α can communicate with $s_{j'}$ in 1-hop. We assign range of s_j and $s_{j'}$ by $d(s_j, s_{M_2[j].ptr})$ and $d(s_{j'}, s_{M_4[j'].ptr})$ respectively. Next, we proceed further in both left and right directions separately. At each move towards left (resp. right) we update $j = M_2[j].ptr$ (resp. $j' = M_4[j'].ptr$) and assign the range of the corresponding s_j (resp. $s_{j'}$) as mentioned above, until $j = 1$ (resp. $j' = n$) is achieved. Range of the other radio stations are assigned to zero.

2.4.3 Phase 3

In this phase, we compute an optimal range assignment for the h-hop broadcast from s_α to all other nodes in S where the solution contains a functional *right-bridge*. Similar method will be adopted to compute the optimal solution with one functional *left-bridge*. The one having minimum cost is chosen as the optimal solution obtained in this phase.

Let us consider a range assignment which includes a right-bridge $\overleftarrow{s_i s_j}$, where $i < \alpha < j$. Here, the range of s_j is assigned to $\rho(s_j) = d(s_j, s_i)$). For this range assignment of s_j, it can communicate with at most s_k to its right. In other words, $d(s_j, s_k) \leq d(s_j, s_i) < d(s_j, s_{k+1})$, $k \geq j$. A right-bridge $\overleftarrow{s_i s_j}$ can be realized in the following two ways:

Scheme 1: Assign $\rho(s_j) = d(s_j, s_i)$.

Scheme 2: If $d(s_j, s_k) \leq d(s_j, s_i) < d(s_j, s_{k+1}) < d(s_j, s_{i-1})$, then assign $\rho(s_j) = d(s_j, s_{k+1})$.

We assume that s_j is reached from s_α using m hops. Thus, using Scheme 1, ($\leq h$)-hop path from s_α to all the nodes in S is achieved by (i) reaching s_1 from s_i in $(h-m-1)$ hops, and (ii) reaching s_n from s_k in $(h-m-1)$ hops. Here the cost of range assignment is $B_1 = \mathcal{C}(s_j, s_\alpha, m) + (d(s_i, s_j))^2 + \mathcal{C}(s_1, s_i, h-m-1) + \mathcal{C}(s_n, s_k, h-m-1)$.

In Scheme 2, s_j can directly communicate with s_{k+1} to the right, and s_i to the left. Thus, the ($\leq h$)-hop path from s_α to all the nodes in S is established by (i) reaching s_1 from s_i in $(h-m-1)$ hops, and (ii) reaching s_n from s_{k+1} in $(h-m-1)$ hops. Here the cost of range assignment is $B_2 = \mathcal{C}(s_j, s_\alpha, m) + (d(s_j, s_{k+1}))^2 + \mathcal{C}(s_1, s_i, h-m-1) + \mathcal{C}(s_n, s_{k+1}, h-m-1)$.

Denoting by $B(\overleftarrow{s_i s_j}, m)$ the cost of range assignment with a right bridge $\overleftarrow{s_i s_j}$ where s_j is reached from s_α using m hops, we have $B(\overleftarrow{s_i s_j}, m) = \min(B_1, B_2)$.

Apart from identifying s_k, $B(\overleftarrow{s_i s_j}, m)$ can be calculated in $O(1)$ time, because

(i) $\mathcal{C}(s_j, s_\alpha, m) = \mathcal{C}(s_\alpha, s_j, m) = M_3[m, j]$ (by Lemma 2.3),

(ii) $\mathcal{C}(s_1, s_i, h-m-1) = M_2[h-m-1, i]$,

(iii) $\mathcal{C}(s_n, s_k, h-m-1) = M_4[h-m-1, k]$, and

(iv) all these matrices are already calculated in Phase 1.

To get an optimal solution with a right-bridge, we need to find $\min_{i=1}^{\alpha-1} \min_{j=\alpha+1}^{n} \min_{m=1}^{h-1} B(\overleftarrow{s_i s_j}, m)$.

In our algorithm, we fix each s_i and compute $\min_{j=\alpha+1}^{n} \min_{m=1}^{h-1} B(\overleftarrow{s_i s_j}, m)$ using Lemma 2.9, stated below.

Lemma 2.9 *If $s_j \in S \setminus \{s_1, s_2, \ldots, s_\alpha\}$, then*

$$\mathcal{C}(s_j, s_\alpha, \mu-1) - \mathcal{C}(s_j, s_\alpha, \mu) \leq \mathcal{C}(s_{j+1}, s_\alpha, \mu-1) - \mathcal{C}(s_{j+1}, s_\alpha, \mu).$$

Proof: Let $A = \{a_0 = s_\alpha, a_1, a_2, \ldots a_{\mu-2}\}$ denote the subsequence (radio stations) of S having non-zero ranges in $R(s_{j+1}, s_\alpha, \mu-1)$. We use $a_{\mu-1}$ to denote s_{j+1}. Thus, the range assigned to $a_i \in A$ is $(x(a_{i+1}) - x(a_i))$ for $i = 0, 1, 2, \ldots, \mu-2$. We use $cost(A)$ to denote $\mathcal{C}(s_{j+1}, s_\alpha, \mu-1)$. Again, let $B = \{b_0, b_1, b_2, \ldots b_{\mu-1}\}$ denote the set of radio stations having non-zero ranges in $R(s_j, s_\alpha, \mu)$, i.e., $cost(B) = \mathcal{C}(s_j, s_\alpha, \mu)$. The range assigned to $b_i (\in B)$ is $(x(b_{i+1}) - x(b_i))$ for $i = 0, 1, 2, \ldots, \mu-1$.

Let us now observe the pairs (a_i, b_{i+1}) for $i = 0, 1, 2, \ldots, \mu-1$. Note that, $x(a_0) - x(b_1) < 0$, and $x(a_{\mu-1}) - x(b_\mu) > 0$. This implies, there exists at least one $i \in \{1, 2, \ldots, \mu-1\}$ such that $x(a_{i-1}) - x(b_i) \leq 0$ and $x(a_i) - x(b_{i+1}) \geq 0$. We consider the smallest $i \geq 1$ such that $x(a_i) - x(b_{i+1}) \geq 0$, and construct two subsequences of radio stations, namely $C = \{a_0 = b_0, a_1, \ldots, a_{i-1}, b_{i+1}, b_{i+2}, \ldots, b_{\mu-1}\}$ and $D = \{a_0 = b_0, b_1, b_2, \ldots, b_i, a_i, a_{i+1}, \ldots, a_{\mu-2}\}$, where length of C is $\mu-1$ and that of D is μ. The ranges assigned to the members in C and D are respectively

- $\{(x(a_1) - x(a_0)), \ldots, (x(a_{i-1}) - x(a_{i-2})), (x(b_{i+1}) - x(a_{i-1})), (x(b_{i+2}) - x(b_{i+1})), \ldots, (x(b_\mu) - x(b_{\mu-1}))\}$, and
- $\{(x(b_1) - x(b_0)), \ldots, (x(b_i) - x(b_{i-1})), (x(a_i) - x(b_i)), (x(a_{i+1}) - x(a_i)), \ldots, (x(a_{\mu-1}) - x(a_{\mu-2}))\}$.

The corresponding costs of these range assignments are

$cost(C) = \sum_{j=0}^{j=i-2} (x(a_{j+1}) - x(a_j))^2 + (x(b_{i+1}) - x(a_{i-1}))^2 + \sum_{j=i+1}^{j=\mu-1} (x(b_{j+1}) - x(b_j))^2$, and

$cost(D) = \sum_{j=0}^{j=i-1} (x(b_{j+1}) - x(b_j))^2 + (x(a_i) - x(b_i))^2 + \sum_{j=i}^{j=\mu-2} (x(a_{j+1}) - x(a_j))^2$.

Thus, $cost(C) + cost(D) = (\sum_{j=0}^{j=\mu-2} (x(a_{j+1}) - x(a_j))^2$ - $(x(a_i) - x(a_{i-1}))^2) + (\sum_{j=0}^{j=\mu-1} (x(b_{j+1}) - x(b_j))^2$ - $(x(b_{i+1}) - x(b_i))^2) + (x(b_{i+1}) - x(a_{i-1}))^2 + (x(a_i) - x(b_i))^2$.

$= cost(A) + cost(B) + 2(x(a_i) - x(b_{i+1}))(x(a_{i+1}) - x(b_i))$
$\leq cost(A) + cost(B)$ (due to the choice of i as mentioned above).

The lemma follows from the fact that $\mathcal{C}(s_j, s_\alpha, \mu - 1) \leq cost(C)$ and $\mathcal{C}(s_{j+1}, s_\alpha, \mu) \leq cost(D)$. $\square$

Lemma 2.10 *While using the bridge $\overleftarrow{s_i s_j}$, $i < \alpha < j$, if $B(\overleftarrow{s_i s_j}, \mu) \leq B(\overleftarrow{s_i s_j}, \mu + 1)$ then $B(\overleftarrow{s_i s_j}, \mu + 1) \leq B(\overleftarrow{s_i s_j}, \mu + 2)$.*

Proof: The gain in cost for increasing the number of hops from μ to $\mu + 1$ to reach from s_α to s_j is $a_1 = (\mathcal{C}(s_j, s_\alpha, \mu) - \mathcal{C}(s_j, s_\alpha, \mu + 1)) \geq 0$. In order to maintain h-hop reachability from s_α to s_1 and s_n, we need to reach both from s_i to s_1 and from s_k to s_n using at most $(h - \mu - 2)$ hops instead of $(h - \mu - 1)$ hops. Thus, the amount of increase in the corresponding costs are $a_2 = \mathcal{C}(s_1, s_i, h - \mu - 2) - \mathcal{C}(s_1, s_i, h - \mu - 1) \geq 0$ and $a_3 = \mathcal{C}(s_n, s_k, h - \mu - 2) - \mathcal{C}(s_n, s_k, h - \mu - 1) \geq 0$.

As stated in the lemma, $B(\overleftarrow{s_i s_j}, \mu) - B(\overleftarrow{s_i s_j}, \mu + 1) \leq 0$ implies $a_1 - a_2 - a_3 \leq 0$.

Now, the gain in cost for increasing the number of hops from $\mu + 1$ to $\mu + 2$ to reach from s_α to s_j is $a'_1 = (\mathcal{C}(s_j, s_\alpha, \mu + 1) - \mathcal{C}(s_j, s_\alpha, \mu + 2)) \geq 0$. This causes the reduction in number of hops from $(h - \mu - 2)$ to $(h - \mu - 3)$ for reaching s_1 from s_i and s_n from s_k. The reduction in the corresponding costs are $a'_2 = \mathcal{C}(s_1, s_i, h - \mu - 3) - \mathcal{C}(s_1, s_i, h - \mu - 2) \geq 0$ and $a'_3 = \mathcal{C}(s_n, s_k, h - \mu - 3) - \mathcal{C}(s_n, s_k, h - \mu - 2) \geq 0$.

By Lemma 2.5, $a'_1 \leq a_1$, $a'_2 \geq a_2$ and $a'_3 \geq a_3$.

Thus, $B(\overleftarrow{s_i s_j}, \mu + 1) - B(\overleftarrow{s_i s_j}, \mu + 2) = a'_1 - a'_2 - a'_3 \leq a_1 - a_2 - a_3 \leq 0$ $\square$

Lemma 2.10 implies that while using the right-bridge $\overleftarrow{s_i s_j}$, we vary the number of hops m to reach s_j from s_α, and compute the corresponding cost $B(\overleftarrow{s_i s_j}, m)$. As soon as $m = \mu$ is reached such that $B(\overleftarrow{s_i s_j}, \mu) < B(\overleftarrow{s_i s_j}, \mu + 1)$, there is no need to check the costs by increasing m beyond $\mu + 1$.

After computing the optimum range assignment with the right-bridge $\overleftarrow{s_i s_j}$, we proceed to compute the same with right-bridge $\overleftarrow{s_i s_{j+1}}$. The following lemma says that if the optimum $B(\overleftarrow{s_i s_j}, m)$ is achieved for $m = \mu$ then while considering the right-bridge

$\overleftarrow{s_i s_{j+1}}$, the optimum $B(\overleftarrow{s_i s_{j+1}}, m)$ will be achieved for some $m \geq \mu$. Here, it needs to be mentioned that, we could not explore any relationship among the optimum costs of range assignments using the right-bridges $\overleftarrow{s_i s_j}$ and $\overleftarrow{s_i s_{j+1}}$.

Lemma 2.11 *For a given $s_i \in S$, $i < \alpha$, if $\min_{m=1}^{h} B(\overleftarrow{s_i s_j}, m)$ and $\min_{m=1}^{h} B(\overleftarrow{s_i s_{j+1}}, m)$ are achieved for $m = \mu$ and ν respectively, then $\nu \geq \mu$.*

Proof: As s_i is fixed, we compute the optimal range assignment $R(s_1, s_i, h - m - 1)$ to reach from s_i to s_1.

While using $\overleftarrow{s_i s_j}$ as the right-bridge, we have $\rho(s_j) = d(s_j, s_i)$, and this enables s_j to reach s_k to its right (i.e. $d(s_j, s_i) \geq d(s_j, s_k)$). Similarly, while using $\overleftarrow{s_i s_{j+1}}$ as the right-bridge, we have $\rho(s_{j+1}) = d(s_{j+1}, s_i)$, and this enables s_{j+1} to reach s_ℓ to its right (i.e. $d(s_{j+1}, s_i) \geq d(s_{j+1}, s_\ell)$). Here $j + 1 \leq k \leq \ell$.

In order to prove the lemma, we need only to show that $B(\overleftarrow{s_i s_{j+1}}, \mu - 1) \geq B(\overleftarrow{s_i s_{j+1}}, \mu)$.

Lemma 2.10 implies that $B(\overleftarrow{s_i s_{j+1}}, m - 1) \geq B(\overleftarrow{s_i s_{j+1}}, m)$ for all $m \leq \mu$. Thus, if $\min(B(\overleftarrow{s_i s_{j+1}}, m))$ is achieved for $m = \nu$, then $\nu > \mu$.

To prove the above inequality, let
$a_1 = \mathcal{C}(s_j, s_\alpha, \mu - 1) - \mathcal{C}(s_j, s_\alpha, \mu)$,
$a_1' = \mathcal{C}(s_\alpha, s_{j-1}, \mu - 1) - \mathcal{C}(s_\alpha, s_{j-1}, \mu)$,
$a_2 = \mathcal{C}(s_1, s_i, h - \mu - 1) - \mathcal{C}(s_1, s_i, h - \mu)$,
$a_3 = \mathcal{C}(s_n, s_k, h - \mu - 1) - \mathcal{C}(s_n, s_k, h - \mu)$ and
$a_3' = \mathcal{C}(s_n, s_\ell, h - \mu - 1) - \mathcal{C}(s_n, s_\ell, h - \mu)$.

As $B(\overleftarrow{s_i s_j}, \mu - 1) > B(\overleftarrow{s_i s_j}, \mu)$, we have $a_1 - a_2 - a_3 > 0$. By Lemma 2.9, $a_1' \geq a_1$ and $a_3' \leq a_3$. Hence, the amount of gain in cost for increasing the number of hops from $\mu - 1$ to μ for reaching from s_α to s_{j+1} and then using the bridge $\overleftarrow{s_i s_{j+1}}$ for broadcasting to the other nodes in S is equal to $B(\overleftarrow{s_i s_{j+1}}, \mu - 1) - B(\overleftarrow{s_i s_{j+1}}, \mu) = a_1' - a_2 - a_3' \geq a_1 - a_2 - a_3 \geq 0$.
$\square$

Given a source-station s_α and another station s_i ($i < \alpha$), the optimal range assignment of the members in S consisting of a functional right-bridge incident at s_i, can be computed using the following algorithm:

Algorithm Range_Assign_using_Right_Bridge(s_i)

Step 1: We initialize $OPT\_j = \alpha$, $OPT\_cost = \infty$ and $k\_store = \alpha$, and $\mu = 1$.

(* μ stores the number of hops allotted to reach s_j from s_α. The role of k_store will be clear in the procedure **compute** invoked from this algorithm.*)

Start with $m = 1$ and $j = \alpha + 1$.

Step 2: At each j, we execute **compute**($B(\overleftarrow{s_i s_j}, m)$, k_store) by incrementing m from its current value upwards until

(i) $B(\overleftarrow{s_i s_j}, m) > B(\overleftarrow{s_i s_j}, m-1)$ is achieved (see Lemma 2.10) or

(ii) m attains its maximum allowable value $\min(h-2, j-\alpha)$.

Step 3: Update OPT_cost and OPT_j observing the value of $B(\overleftarrow{s_i s_j}, m-1)$ or $B(\overleftarrow{s_i s_j}, m)$ depending on whether Step 2 has terminated depending on Case (i) or Case (ii).

Step 4: For the next choice of j, update μ by $m-1$ or m depending on whether Case (i) or (ii) occurred in Step 2 (see Lemma 2.11).

Procedure **compute**($B(\overleftarrow{s_i s_j}, m)$, k_store)

- Initialize $k = k\_store$.
- Increment k to identify the right-most radio station s_k such that $d(s_j, s_k) \leq \rho(s_j)$ ($= d(s_j, s_i)$).
- Set $k\_store = k$ for further use. (* i.e., for next j, the search for k will start from k_store *)
- Compute $B(\overleftarrow{s_i s_j}, m) = (\rho(s_j))^2 + R(s_j, s_\alpha, m) + R(s_1, s_i, h-m-1) + R(s_n, s_k, h-m-1)$; the last three terms are available in $M_3[m, j]$, $M_2[h-m-1, i]$ and $M_4[h-m-1, k]$ respectively.

Let $\overleftarrow{s_i s_j}$ be the functional right-bridge corresponding to the optimum solution. Here, $\rho(s_j) = d(s_i, s_j)$. The range assignment of the radio stations (i) $\{s_\alpha, s_{\alpha+1}, \ldots, s_{j-1}\}$ can be obtained from the matrix M_3, (ii) $\{s_1, s_2, \ldots, s_{\alpha-1}\}$ can be obtained from the matrix M_1, and (iii) $\{s_j, s_{j+1}, \ldots, s_n\}$ can be obtained from the matrix M_4 as described in Step 6 of the algorithm proposed in Subsection 2.4.2 and using the Lemma 2.3.

Theorem 2.2 *For a given s_i $(i < \alpha)$, algorithm Range_Assign_using_Right_Bridge needs $O(n - \alpha + h)$ time in the worst case.*

Proof: Follows from Lemmas 2.10 and 2.11, and the role of *k_store* in the procedure **compute** for locating right-most s_k such that $d(s_j, s_k) \geq \rho(s_j)$. □

2.4.4 Complexity analysis

Theorem 2.3 *Given a set of radio station S and a source station $s_\alpha \in S$, the optimum range assignment for broadcasting message from s_α to all the members in S using at most h-hops can be computed in $O(n^2)$ time and using $O(nh)$ space.*

Proof: Phase 1 needs $O(nh)$ time for initializing the matrices. Optimum functional bridge-free solution can be obtained in $O(n)$ time as described in Phase 2. Finally in Phase 3, we fix s_i to the left of s_α and identify the optimum solution with a functional right-bridge incident at s_i in $O(n - \alpha + h)$ time (see Theorem 2.2). For $(\alpha - 1)$ such s_i's, the total time required in this phase is $O(\alpha(n - \alpha + h))$. Similarly, the worst case time required for finding the optimum range assignment with exactly one functional left-bridge is $O((n - \alpha)(\alpha + h))$. Thus, the result follows. □

2.5 Summary

An easy to implement algorithm for the h-hop broadcast range assignment problem for the linear radio network is presented. The worst case time complexity of our algorithm is $O(n^2)$. This is an improvement over the existing result on this problem by a factor of h [31]. Further reduction in the time complexity of the problem is an interesting open problem. It is observed that if the source radio station is at one end, then the time complexity of the h-hop broadcast range assignment is $O(nh)$. Thus, it seems that, one may improve the time complexity to $O(nh\ polylog(h))$ by efficiently designing Phase 3 of the algorithm.

Chapter 3

Weighted Broadcast in Linear Radio Networks

3.1 Introduction

In this chapter, the weighted version of the range assignment problem is studied in the context of information *broadcast* and *accumulation* in a linear radio network. Efficient algorithms for the unbounded and bounded-hop *broadcast* problems are presented. Unlike the problem in the previous chapter, here each radio station $s \in S$ is attached with a weight $w(s)$ ($= \gamma(s)$). Thus the cost function is $\Sigma_{s \in S} w(s) \times (\rho(s))^2$, where $\rho(s)$ is the range assigned to s. The objective is to assign transmission ranges to the members in S such that each member in S is reachable from the source radio station s^* in at most h hops, and the value of the cost function is minimum. Our proposed algorithm is based on dynamic programming, and it runs in $O(hn^2 \log n)$ time. In the unbounded case (i.e., $h = n-1$), we have used graph-theoretic formulation of the problem and proposed an $O(n^2)$ time algorithm. We have also studied the unbounded version of the weighted *accumulation* range assignment problem. Here the objective is to assign ranges to the radio stations in S such that all the radio stations can send message to the target radio station s^*, and the cost $\Sigma_{s \in S} w(s) \times (\rho(s))^2$ of the entire network is minimum. The time complexity of our algorithm is $O(n^2)$. Our proposed algorithm for accumulation problem can work when the radio stations are placed in any arbitrary dimension.

3.2 Preliminaries

Let $S = \{s_1, s_2, \ldots, s_n\}$ be the set of n radio stations on a straight line from left to right. An weight $w_i > 0$ is assigned with each s_i, $i = 1, 2, \ldots, n$. Let $s^* = s_\theta \in S$ be the source radio station where from the message needs to be broadcast. The communication graph for a range assignment $\mathcal{R} = \{\rho(s_1), \rho(s_2), \ldots, \rho(s_n)\}$, as defined in Section 1.1, is a directed graph $G = (V, E)$ with $V = S$ and $E = \{(s_i, s_j) | d(s_i, s_j) \leq \rho(s_i)\}$. As in Chapter 2, here also we can define an edge to be *functional* with respect to the h-hop broadcast path in G (see Definition 2.1).

Figure 3.1 demonstrates an instance of the unbounded weighted broadcast range assignment problem with five radio stations, along with the optimum solution. Here, $d(s_1, s_2) = 8$, $d(s_2, s_3) = 2$, $d(s_3, s_4) = 1$ and $d(s_4, s_5) = 4$. The weight of the radio stations $s_1, s_2, \ldots, s_5$ are 10, 1, 10000, 100 and 0.01 respectively, source is s_3 (=s^*), the range of each radio station is indicated by the arrow-headed line as shown in the figure, and the cost of the optimum range assignment is 10,951.25 units.

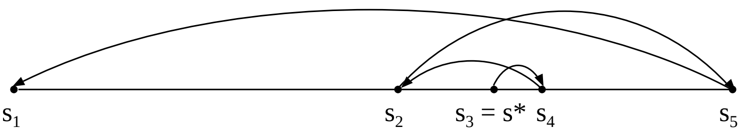


Figure 3.1: An example of linear weighted broadcast problem

Definition 3.1 *Let* $\Pi = \{s_\theta = s_{i_1}, s_{i_2}, \ldots, s_{i_k} = s_j\}$ *be a path from the source* $s^* = s_\theta$ *to a radio station* s_j *(*$j > \theta$*) in the broadcast communication graph* G *corresponding to* *a range assignment $\mathcal{R}$. An edge $(s_{i_a}, s_{i_{a+1}})$ is said to be a* right back edge *if $i_{a-1} < i_a$ and $i_{a+1} < i_a$ (see the dashed edge (s_i, s_j) in Figure 3.2).*

Similarly, on a path from s_θ to a radio station s_j $(j < \theta)$ a *left back edge* can be defined.

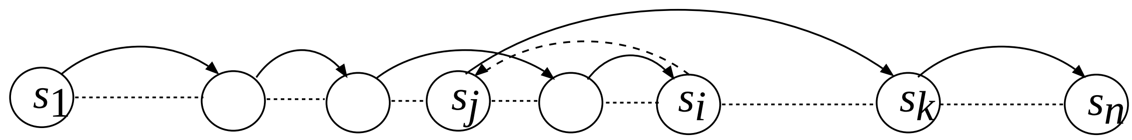


Figure 3.2: An example of right back edge

Lemma 3.1 *If $\theta = 1$ (resp. $\theta = n$), then in the minimum cost h-hop weighted broadcast range assignment, there is no functional right (resp. left) back edge.*

Proof: Similar to the proof of Lemma 2.2. □

As in Chapter 2, here also the optimum broadcast range assignment may consists of functional left-bridge(s) and/or functional right-bride(s) (see Definitions 2.1 and 2.2 and the subsequent discussions). But unlike the earlier problem, this weighted version of the problem may consist of many such bridges in the optimum solution (see the example in Figure 3.1). It needs to be mentioned that, the results stated in Lemmas 2.1 and 2.2 (of Chapter 2) hold for this weighted version also. The following three lemmas, numbered 3.2, 3.3 and 3.4, characterize a feasible solution of the h-hop weighted broadcast problem.

Lemma 3.2 *If $\overleftarrow{s_a s_b}$ and $\overleftarrow{s_{a'} s_{b'}}$ are two functional right-bridges present in a h-hop weighted broadcast range assignment $\mathcal{R}$, then (i) $b \neq b'$, (ii) $a \neq a'$, and (iii) if $b < b'$, then $a' < a$.*

Proof: (i) If $b = b'$ then trivially one of these two bridges will not remain functional. Same argument holds for part (ii) also.

(iii) On the contrary, let $a' \geq a$. Now any path from source s_θ to $s_{b'}$ implies that there is also a path from s_θ to s_b as $b' > b > \theta$. Since $a' \geq a$, all the radio stations s_k $(a' \leq k < \theta)$ are reachable using the right-bridge $\overleftarrow{s_a s_b}$. Thus, the right-bridge $\overleftarrow{s_{a'} s_{b'}}$ will no longer remain functional (see Figure 3.3(a)). $\square$

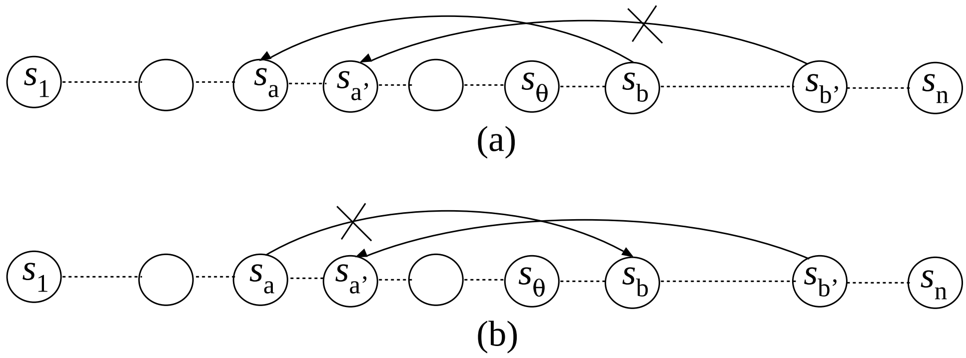


Figure 3.3: Illustrations of (a) Lemma 3.2 and (b) Lemma 3.3

Lemma 3.3 *Let $\overrightarrow{s_a s_b}$ be a functional left-bridge and $\overleftarrow{s_{a'} s_{b'}}$ be a functional right-bridge in a h-hop weighted broadcast range assignment $\mathcal{R}$. Now, (i) if $a \leq a'$ then $b' < b$, and (ii) if $a > a'$ then $b' \geq b$.*

Proof: (i) On the contrary, let $a \leq a'$, but $b' \geq b$. Now, the path from the source s_θ to s_a may or may not use the right-bridge $\overleftarrow{s_{a'} s_{b'}}$. If that path uses the right-bridge $\overleftarrow{s_{a'} s_{b'}}$, then obviously left-bridge $\overrightarrow{s_a s_b}$ will not remain functional (see Figure 3.3(b)). Again, if the path does not use right-bridge $\overleftarrow{s_{a'} s_{b'}}$, then obviously it is not functional.

(ii) Proof of this part is similar to part (i). $\square$

Lemma 3.4 *Let $\overrightarrow{s_a s_b}$ be a left-bridge and $\overleftarrow{s_{a'} s_{b'}}$ be a right-bridge such that $a' < a < \theta < b' < b$. Now, if both the bridges $\overrightarrow{s_a s_b}$ and $\overleftarrow{s_{a'} s_{b'}}$ are functional in a h-hop weighted broadcast range assignment, then (i) there is no functional right-bridge $\overleftarrow{s_p s_q}$ such that $p \leq a'$ and $q \geq b$ (see Figure 3.4(a)) and (ii) there is no functional left-bridge $\overrightarrow{s_{p'} s_{q'}}$ such that $p' \leq a'$ and $q' \geq b$ (see Figure 3.4(b)).*

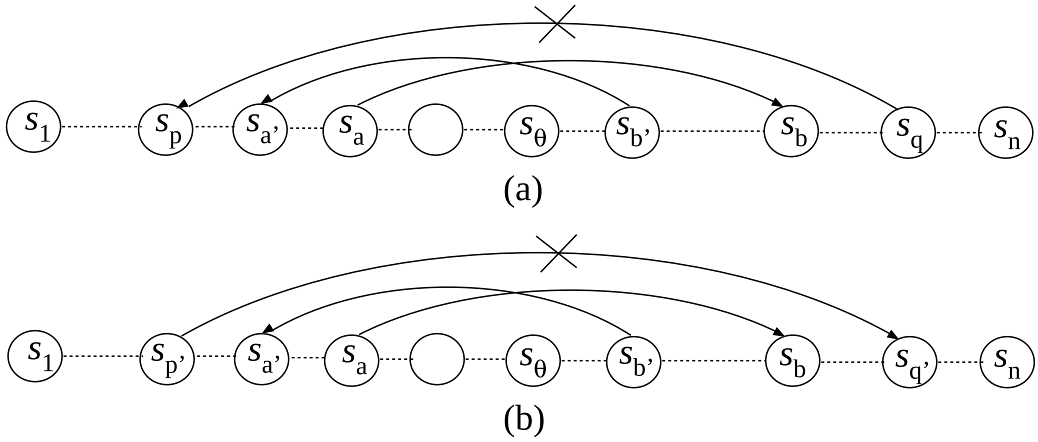


Figure 3.4: Illustration of Lemma 3.4

Proof: On the contrary, let one of the following two situations be present in the range assignment:

Situation 1: $\overleftarrow{s_p s_q}$ is a functional right-bridge such that there is no functional left-bridge $\overrightarrow{s_e s_f}$ with $p < e \leq a'$ and $b < f \leq q$. The justification of the inequalities in the choice of e and f are as follows: $e = p$ is not possible since $\overleftarrow{s_p s_q}$ is functional, and by Lemma 3.2, $b = f$ is also not possible.

Situation 2: $\overrightarrow{s_{p'} s_{q'}}$ is a functional left-bridge such that there is no functional right-bridge $\overleftarrow{s_{e'} s_{f'}}$ with $p' \leq e' < a'$ and $b \leq f' < q$.

Situation 1 (resp. Situation 2) produces an affirmative instance of Case (i) (resp. Case (ii)) of the lemma. We shall now prove the Case (i) of the lemma. Case (ii) can be similarly proved.

Let there is an instance satisfying Situation 1. Since there is no 1-hop path from any radio station in $\{s_{p+1}, s_{p+2}, \ldots, s_{a'}\}$ to any radio station in $\{s_{b+1}, s_{b+2}, \ldots, s_q\}$, there exists a path from s_θ to s_q without using the bridge $\overleftarrow{s_{a'} s_{b'}}$. Again, a radio station, which is covered by the bridge $\overleftarrow{s_{a'} s_{b'}}$, is also covered by $\overleftarrow{s_p s_q}$. Thus, the right-bridges $\overleftarrow{s_{a'} s_{b'}}$ and $\overleftarrow{s_p s_q}$ can not be simultaneously functional. Thus, we have a contradiction. □

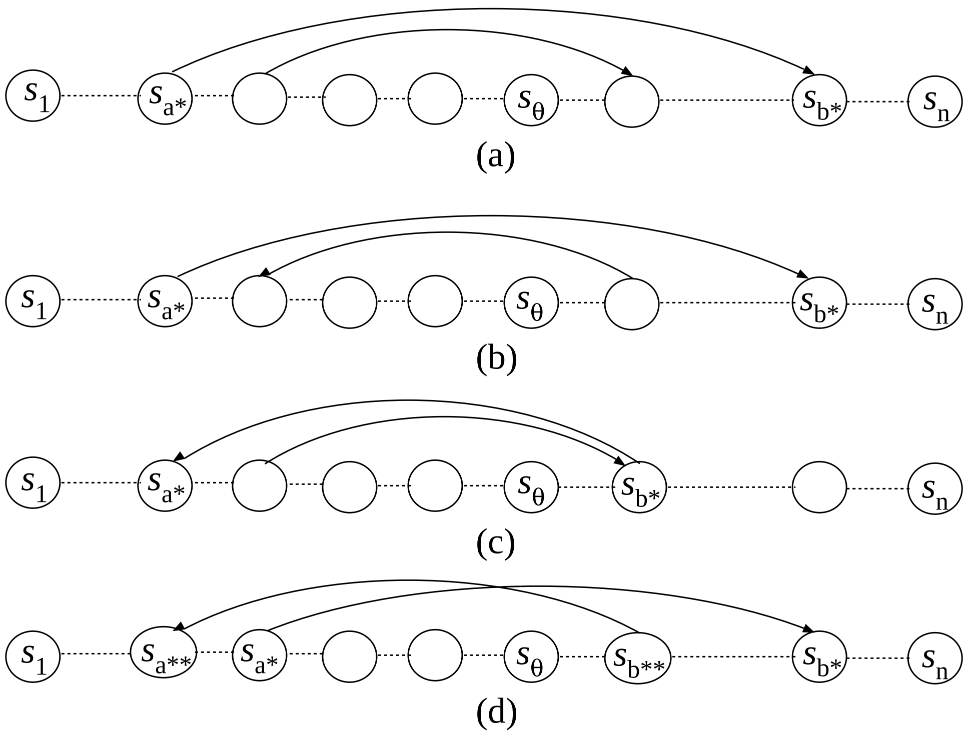


Figure 3.5: Illustration of left-most and right-most functional bridges

The optimum h-hop weighted broadcast range assignment may consist of many left-bridges and/or many right-bridges (see Figure 3.1). Feasible configurations of overlapping bridges are shown in Figure 3.5. We now introduce the concept of *right-most functional right-bridge* and *left-most functional left-bridge* as follows. These help us in designing our algorithms.

Definition 3.2 *A functional right-bridge* $\overleftarrow{s_{a^*}s_{b^*}}$ *in a range assignment is said to be* right-most functional right-bridge *if (i) there exists no other functional right-bridges* $\overleftarrow{s_a s_b}$ *in that range assignment satisfying* $b \geq b^*$ *and (ii) there exists no functional left-bridge* $\overrightarrow{s_{a'}s_{b'}}$ *satisfying* $a' \leq a^*$.

Lemma 3.2 says that, if more than one right bridges exist in a range assignment, and $\overleftarrow{s_{a^*}s_{b^*}}$ is the right-most functional right-bridge, then for all other functional right-bridge $\overleftarrow{s_a s_b}$, $a > a^*$.

Similarly, one can define a *left-most functional left-bridge*. In Figure 3.5(a) and 3.5(b), $\overrightarrow{s_{a^*}s_{b^*}}$ is the left-most functional left-bridge; in Figure 3.5(c), $\overleftarrow{s_{a^*}s_{b^*}}$ is the right-most functional right-bridge; in Figure 3.5(d), $\overrightarrow{s_{a^*}s_{b^*}}$ and $\overleftarrow{s_{a^{**}}s_{b^{**}}}$ are left-most and right-most functional bridges respectively.

3.3 Unbounded weighted broadcast problem

In this section, we consider unbounded version ($h = n - 1$) of the weighted broadcast problem. The following lemma says that the optimal solution for the unbounded broadcast range assignment problem may be one of the following three types: (i) with no functional bridge, (ii) with a right-most functional right-bridge, and (iii) with a left-most functional left-bridge.

Lemma 3.5 *In a unbounded broadcast range assignment, both left-most functional left-bridge and right-most functional right-bridge can not exist simultaneously.*

Proof: On the contrary, let the optimum range assignment $\mathcal{R}$ contains both right-most functional right-bridge ($\overleftarrow{s_a s_b}$) and left-most functional left-bridge ($\overrightarrow{s_{a'}s_{b'}}$). Let s_ℓ be the right-most radio station satisfying $d(s_b, s_\ell) \leq d(s_b, s_a)$, and $s_{\ell'}$ be the left-most radio station satisfying $d(s_{a'}, s_{\ell'}) \leq d(s_{a'}, s_{b'})$. Therefore from the Definition 3.2, we have $a < a' < \theta < b < b'$. This implies, either s_b is reachable from s_θ using the bridge $\overrightarrow{s_{a'}s_{b'}}$ or $s_{a'}$ is reachable from s_θ using the bridge $\overleftarrow{s_a s_b}$. Without loss of generality assume that s_b is reachable from s_θ using the bridge $\overrightarrow{s_{a'}s_{b'}}$. Here we need to consider the following two situations: (i) $b < b' \leq \ell$ (see Figure 3.6(a)) and (ii) $b' > \ell$ (see Figure 3.6(b)).

If (i) is true, then left-bridge $\overrightarrow{s_{a'}s_{b'}}$ itself is not functional. If (ii) is true, then the right-bridge $\overleftarrow{s_a s_b}$ will not remain functional since $d(s_{a'}, s_{b'}) > d(s_a, s_b)$, and hence the radio stations covered by s_b will also be covered by $s_{a'}$. $\square$

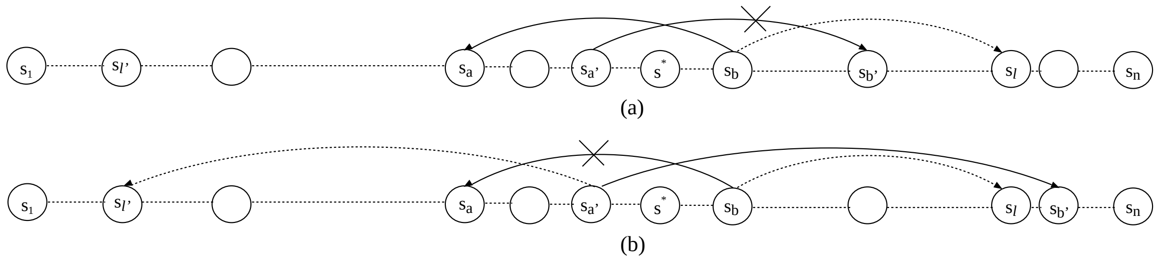


Figure 3.6: Proof of Lemma 3.5

We first execute a preprocessing phase, and then compute the optimum solution for the above three cases separately. Finally, the one having minimum cost is reported.

3.3.1 Preprocessing

In this step, we use dynamic programming to create three arrays M, N and P, each of size n. Each entry of these arrays is a tuple (χ, γ) as described below.

$M[i].\chi$ stores the cost of the optimum broadcast from s_i to all the radio stations in the set $S_i^+ = \{s_i, s_{i+1}, \ldots, s_n\}$, and $M[i].\gamma$ stores the index of the farthest radio station to the right of s_i which can be reached from s_i in 1 hop due to the assigned range of s_i.

$N[i].\chi$ contains the cost of optimum broadcast range assignment from s_i to all the nodes in $S_i^- = \{s_1, s_2, \ldots, s_i\}$, and $N[i].\gamma$ contains the index of the farthest radio station to the left of s_i which can be reached from s_i in 1 hop due to the assigned range of s_i.

The fields attached to the elements of the array P are defined using a complete weighted digraph G with the vertices corresponding to the radio stations in S. The weight attached with a directed edge (s_a, s_b) is $w_a \times (d(s_a, s_b))^2$. $P[i].\chi$ contains the cost of the shortest path from s_θ to s_i in the graph G, and $P[i].\gamma$ contains the index of the predecessor of s_i in the shortest path from s_θ to s_i.

We now describe an algorithm for computing the arrays M and P. The array N can be computed using the technique for computing M.

Lemma 3.6 *If $i < n$ then $M[i].\chi = \min_{k=i}^{n}(w_i \times (d(s_i, s_k))^2 + M[k].\chi)$, and if minimum is obtained for $k = k^*$, then $M[i].\gamma = k^*$. If $i = n$ then $M[i].\chi = 0$ and $M[i].\gamma = n$.*

Proof: The case for $i = n$ is trivial. So, we consider the case where $i < n$. It is clear that, if there is a path from s_i to s_n in the communication graph corresponding to a range assignment, then that range assignment is feasible for the broadcast from s_i to all the members in S_i^+. By Lemma 3.1, there is no functional back edge in the optimum range assignment for broadcasting from s_i to the members in S_i^+. Thus, there exists some $s_k \in S_i^+$, where s_i first reaches s_k in 1 hop, and then reaches s_n from s_k in a minimum cost path. This proves the lemma. □

Thus, Lemma 3.6 gives a dynamic programming based algorithm for computing M and N that runs in $O(n^2)$ time.

In order to compute the entries of the array P, we may use Dijkstra's single source shortest path algorithm to compute the cost of the shortest path from s_θ to each radio station $s_i \in S$. This needs $O(n^2)$ time. Note that, the weight of an edge in G can be computed from the positional information and weight attached to its end-vertices. Thus, P can be computed without explicitly constructing the graph G.

By Lemma 3.1, $M[1].\chi$ and $N[1].\chi$ give the cost of range assignment for the weighted unbounded broadcast when $\theta = 1$ and $\theta = n$ respectively. We now consider the case where $s^* = s_\theta$, and $\theta \neq 1$ or n.

3.3.2 Bridge-free solution

The algorithm for the optimal solution with no functional bridge uses two variables *opt_cost* and *opt_range*, where *opt_cost* stores the cost of optimum range assignment, and *opt_range* stores the index α such that $\rho(s_\theta) = d(s_\theta, s_\alpha)$ in the range assignment corresponding to *opt_cost*. The stepwise description of the algorithm is as follows:

Step 1: $opt\_cost = \infty$.

Step 2: Consider each element $s_i \in S \setminus \{s_\theta\}$ in order of their distances from s_θ.

Compute the optimum range assignment with $\rho(s_\theta) = d(s_\theta, s_i)$ as follows:

Let s_α and s_β be respectively the left-most and right-most radio stations satisfying $d(s_\theta, s_\alpha) \leq \rho(s_\theta)$ and $d(s_\theta, s_\beta) \leq \rho(s_\theta)$ respectively.

If $i = \alpha$, then $C = N[\alpha].\chi + w_\theta \times (\rho(s_\theta))^2 + \min_{j=\theta+1}^{\beta} M[j].\chi$, and

if $i = \beta$, then $C = \min_{j=\alpha}^{\theta-1} N[j].\chi + w_\theta \times (\rho(s_\theta))^2 + M[\beta].\chi$.

If the value of C is less than opt_cost, then update opt_cost and opt_range.

Step 3: (* Range assignment *)
Set $\rho(s_\theta) = d(s_\theta, s_{\alpha^*})$, where α^* is stored in opt_range. The range of the other radio stations are computed as follows:

Let $\alpha^* < \theta$. We assign $\alpha = \alpha^*$. Let s_β ($\beta > \theta$) be the right-most radio station such that s_θ can reach s_β in 1 hop.

We identify an index k such that $M[k].\chi = \min_{j=\theta+1}^{\beta} M[j].\chi$. Next, we assign $\rho(s_\alpha) = d(s_\alpha, s_{N[\alpha].\gamma})$ and $\rho(s_k) = d(s_k, s_{M[k].\gamma})$.

We proceed further in both left and right directions separately. At each move towards left (resp. right) we update $\alpha = N[\alpha].\gamma$ (resp. $k = M[k].\gamma$), and assign the range of s_α (resp. s_k) as mentioned above, until $\alpha = 1$ (resp. $k = n$) is reached. Range of the other radio stations are assigned to zero.

The case, where $\alpha^*(= opt\_range) > \theta$ is similarly handled.

3.3.3 Solution having right-most functional right-bridge

In this subsection, we describe the algorithm for computing the optimal solution having right-most functional right-bridge. It considers each possible right-bridge as a right-most functional right-bridge and computes the cost of the corresponding range assignment. Finally, we choose the one having minimum cost.

Let $\overleftarrow{s_p s_q}$ denote the right-most functional right-bridge that produces optimum cost. We compute the optimum cost range assignment for reaching s_q from s_θ using the array P. Let s_r be the right-most radio station which is reachable from s_q in 1 hop due to its assigned range. Thus, all the radio stations $S_{pq} = \{s_p, s_{p+1}, \ldots, s_q, s_{q+1}, \ldots, s_r\}$ are reached from s_θ. Now, we need to consider the path for reaching s_1 to the left and s_n to the right.

For reaching s_1, we will only consider s_p, and compute the minimum cost path. The reasons are stated below.

- If we choose some element $s_j \in S_{pq}$, $j < \theta$ then the cost of reaching s_1 can further be reduced by choosing the bridge $\overleftarrow{s_j s_q}$, which we have separately considered.
- If we choose some element $s_j \in S_{pq}$, $q \geq j > \theta$ then such a pair of overlapping functional right-bridges can not exist in the optimum solution (see Lemma 3.3).
- If we choose some element s_j, $j > q$, for reaching s_1 and it hops at $s_{p'}$, then $p' < p$ (by Lemma 3.3). In this situation, the right-bridge $\overleftarrow{s_{p'} s_j}$ is another right-bridge, which will be separately considered as the right-most functional right-bridge.

Note that, one may need to consider the cost of range assignments for the paths from several radio stations to reach s_n due to the weight constraint. Below we justify that we have to choose a radio station s_t, $q < t \leq r$, for reaching s_n.

- If we choose a radio station s_t, $t \leq p$ for reaching s_n, then the optimum path from s_t to s_n will take its first hop at a node $s_{r'}$, where $r' > q$ (by Lemma 3.3). But, this situation results the same cost while considering $\overrightarrow{s_t s_{r'}}$ as the left-most functional left-bridge. Thus, such a choice of s_t ($t \leq p$) is not required.

- If we choose a radio station s_t, $p < t < \theta$, then by Lemma 3.5, the assigned range of s_t is such that it does not form the left-most functional left-bridge. Thus, the path from s_t to s_n passes through a node s_k, where $\theta < k < q$. The cost of this path is strictly less than that from s_k to s_n. Thus, such a choice of s_t, $p < t < \theta$ is also not required.

- If $s_t = s_\theta$, then, following the same reason as mentioned in the above item, the minimum cost path from s_θ to s_n can not pass through a node s_k, where $\theta < k \leq r$. Thus, the first hop from s_θ is at a node s_k, where $k > r$. This implies, that the right-bridge $\overleftarrow{s_p s_q}$ is not functional.

- Following a similar argument as stated in the above item, it can be shown that a choice of s_t, $\theta < t \leq q$, will also not produce a minimum cost path with $\overleftarrow{s_p s_q}$ as the right-most functional right-bridge.

We maintain *opt_cost* to store the optimum cost. We also maintain a tuple of 5 integer variables (p, q, r, t, f); p, q, r and t are as explained earlier, and f is a flag bit. The arrays M and N are used for computing the range assignments for reaching s_n from s_t, and s_1 from s_p respectively. The stepwise algorithm is stated below. It maintains three variables ℓ, ℓ', and k. While considering a bridge $\overleftarrow{s_a s_b}$, s_ℓ denotes the right-most radio station such that $d(s_b, s_\ell) \leq d(s_b, s_a)$, ℓ' is a temporary variable, and the index k is such that $M[k].\chi = \min_{j=b+1}^{\ell} M[j].\chi$.

Step 1: Initialize *opt_cost* $= \infty$.

Step 2: (* Compute the optimum solution with a right-most functional right-bridge*)

Consider each radio-station s_b, $\theta < b \leq n$.

Step 2.0: Initialize $\ell = b$.

Consider each right-bridge $\overleftarrow{s_a s_b}$, $a = \theta - 1, \theta - 2, \ldots, 1$ in order, and execute the steps 2.1-2.5.

Step 2.1: Set $\rho(s_b) = d(s_a, s_b)$, and $\ell' = \ell$

Repeatedly increment ℓ by 1 until $d(s_b, s_\ell) > d(s_a, s_b)$. Finally decrement ℓ by 1.

Step 2.2: Compute k such that $M[k].\chi = \min_{j=\ell'+1}^{\ell} M[j].\chi$ (see Lemma 3.5).

Step 2.3: Thus the optimum cost for considering the right-bridge $\overleftarrow{s_a s_b}$ as the right-most functional right-bridge is

$$C = P[b].\chi + w_b \times (d(s_a, s_b))^2 + N[a].\chi + M[k].\chi.$$

Step 2.4: If $C < opt\_cost$, then set $opt\_cost = C$, and set $(p, q, r, t, f) = (a, b, \ell, k, 0)$.

Step 2.5: If $d(s_b, s_{\ell+1}) < d(s_{a-1}, s_b)$ then $\rho(s_b) = d(s_b, s_{\ell+1})$ also serves the role of right-bridge $\overleftarrow{s_a s_b}$, and we compute the optimum cost for this assignment of $\rho(s_b)$. If the corresponding cost is less than *opt_cost*, then store it in *opt_cost*, set $f = 1$, and set the other fields of the 5-tuple appropriately.

Step 3: Finally, the range assignment corresponding to the optimum solution, is done as follows:

Let the 5-tuple (p, q, r, t, f) corresponds to the optimum solution.

Assign $\rho(s_q) = d(s_p, s_q)$ or $d(s_q, s_r)$ depending on whether f bit is 0 or 1. This establishes the right-bridge $\overleftarrow{s_p s_q}$. The range assignment of all the radio stations on the path from s_θ to s_q are obtained from the array P. The range assignment from s_p to s_1 are obtained from the array N and the range of the radio stations from s_t to s_n are obtained from the array M as described in Subsection 3.3.2.

3.3.4 Correctness and complexity

The correctness of the algorithm for computing optimum solution without any functional bridge follows from the fact that we have considered all possible range of the source s_θ. For each choice of the range of s_θ, we have computed the optimum range assignment of the other radio stations for reaching s_1 and s_n. The correctness of the algorithm for computing the optimum range assignment with a right-most functional right-bridge $\overleftarrow{s_p s_q}$ follows from Lemmas 3.2, 3.3, 3.5, and the discussions in Subsection 3.3.3 on the choice of radio stations from which s_1 and s_n are reached. Exactly same method as in Subsection 3.3.3 works for computing the optimum range assignment with a left-most functional left-bridge.

Theorem 3.1 *The worst case time and space complexities of the weighted unbounded broadcast range assignment problem are $O(n^2)$ and $O(n)$ respectively.*

Proof: The space complexity follows from the size of the arrays M, N and P. The time required for computing these arrays is $O(n^2)$. After computing M and N, the time required for computing the optimum solution with no functional bridge is $O(n)$ time. In Step 2 of the algorithm, we fix a radio station b and consider the right-bridges $\overleftarrow{s_a s_b}$ for all $a = \theta - 1, \theta - 2, \ldots, 1$ in order. Note that, the computation of ℓ and k in each iteration of Steps 2.1-2.5 (i.e., for a fixed index b) is incremental. Step 2.5 may also need some incremental time, and this reduces the time requirement of the next iteration. Thus, for a particular radio station s_b, $b > \theta$, the total number of distance computations in Step 2.1 and finding minimum value in the M array in Step 2.2 is $O(n)$. This indicates that the total time complexity of Step 2 is $O(n^2)$. The range assignment in Step 3 needs $O(n)$ time. □

3.4 Weighted h-hop broadcast problem

If the number of hops is restricted to a specified integer h $(1 \leq h \leq n-1)$, the graph-theoretic approach, described above, does not work. We apply dynamic programming based approach for solving this problem. We first compute three $n \times h$ matrices, namely A, B and C, whose each entry is a tuple (χ, γ) as mentioned below.

(a) $A[i,j].\chi$ = minimum cost for sending message from s_i to the radio stations in $S_{i+1}^{+} = \{s_{i+1}, s_{i+2}, \ldots, s_n\}$ using at most j hops; $A[i,j].\gamma$ = index k, such that in the minimum cost j-hop path from s_i to s_n, the first hop takes place at s_k.

(b) $B[i,j].\chi$ = minimum cost for sending message from s_i to the radio stations in $S_{i-1}^{-} = \{s_1, s_2, \ldots, s_{i-1}\}$ using at most j hops; $B[i,j].\gamma$ = index k, such that in the minimum cost j-hop path from s_i to s_1, the first hop takes place at s_k.

(c) $C[i,j].\chi$ = minimum cost of sending message from s_θ (source) to s_i using at most j hops; $C[i,j].\gamma$ = index k, such that in the minimum cost j-hop path from s_θ to s_i, the last hop takes place from s_k to s_i.

We explain the computation of matrices A and C. The computation of the matrix B is similar to that of A.

The elements of the first column of matrix A are $A[i,1].\chi = w_i \times (d(s_i, s_n))^2$ and $A[i,1].\gamma = n$ for $i = 1, 2, \ldots, n$. After computing the $(j-1)$-th column, the computation of the j-th column is as follows:

$$A[i,j].\chi = \min_{k=i}^{n}(w_i \times (d(s_i, s_k))^2 + A[k, j-1].\chi)$$

If the minimum is achieved for $k = k'$, then we set $A[i,j].\gamma = k'$.

The elements of the first column of matrix C are $C[i,1].\chi = w_\theta \times (d(s_\theta, s_i))^2$ and $C[i,1].\gamma = \theta$, for $i = 1, 2, \ldots, n$. After computing the $(j-1)$-th column, the computation of the j-th column is as follows:

$$C[i,j].\chi = \min_{k=1}^{n}(C[k, j-1].\chi + w_k \times (d(s_k, s_i))^2)$$

If the minimum is achieved for $k = k'$, then we set $C[i,j].\gamma = k'$.

It is clear from the above discussion that the time required for computing the matrices A, B and C is $O(hn^2)$. In the following two subsections we describe the method of computing the optimum range assignment for the h-hop broadcast (i) with no functional bridge and (ii) with right-most functional right-bridge. The optimum solution with left-most functional left-bridge is similarly computed.

3.4.1 Bridge-free solution

The algorithm for the weighted h-hop broadcast range assignment problem having no functional bridge is similar to the algorithm for the unbounded version of the same problem described in Subsection 3.3.2. The only change is that, here we need to replace $M[i]$ and $N[i]$ by $A[i, h-1]$ and $B[i, h-1]$ respectively.

3.4.2 Solution with right-most functional right bridge

We consider each $\overleftarrow{s_a s_b}$ ($a < \theta < b$), and compute the minimum cost of range assignment with $\overleftarrow{s_a s_b}$ as the right-most functional right-bridge. Finally, the one having the overall minimum cost, is identified.

Here also, we will initialize *opt_cost* by ∞, and will use the 5-tuple (p, q, r, t, f) as described in the algorithm of Subsection 3.3.3.

We shall consider each right-bridge $\overleftarrow{s_a s_b}$ ($1 \leq a < \theta < b \leq n$) and each integer k ($1 \leq k \leq h$). For a particular choice of $\overleftarrow{s_a s_b}$ and k, we execute the following steps (assuming $\overleftarrow{s_a s_b}$ is the right-most functional right-bridge, and s_b is reached from s_θ (source) in k hops).

Step 1: We assign $\rho(s_b) = d(s_a, s_b)$ to implement the right-bridge $\overleftarrow{s_a s_b}$. Let ℓ be the maximum index such that $d(s_b, s_\ell) \leq d(s_a, s_b)$.

Step 2: From the C matrix, we compute the minimum cost range assignment for reaching s_θ to s_b using k hops. Now, the following three sets of nodes are already reached from s_θ.

(i) In the k-hop path from s_θ to s_b, $S^1_{ab} = \{s_i, s_{i+1}, \ldots, s_b\}$ are reached from s_θ in $h_i, h_{i+1}, \ldots, h_b$ hops respectively, where each $h_j \leq k$,

(ii) the bridge $\overleftarrow{s_a s_b}$ enables $S^2_{ab} = \{s_a, s_{a+1}, \ldots, s_{i-1}\}$ to be reached from s_θ in $k+1$ hops (where $a < i$), and

(iii) due to the assigned range of s_b, the radio stations in $S^3_{ab} = \{s_{b+1}, s_{b+2}, \ \ldots \ s_\ell\}$ are all reached from s_θ in $k+1$ hops (where $\ell \geq b$). In particular, if $\ell = b$ then $S^3_{ab} = \emptyset$.

Step 3: As described in Subsection 3.3.3, here also we compute the cost of reaching s_1 from s_a only, which is equal to $B[a, h-k-1]$.

Step 4: For reaching s_n, we need to choose an appropriate radio stations in $S_{ab} = S^1_{ab} \cup S^2_{ab} \cup S^3_{ab}$ for which the cost is minimum.

We compute $A^* = \min(A^*_1, A^*_2, A^*_3)$, where
$A^*_1 = \min_{j=i}^{b} A[j, h-h_j]$ (* cost of reaching s_n from any one of the nodes in S^1_{ab} *),
$A^*_2 = \min_{j=a}^{i-1} A[j, h-k-1]$ (* cost of reaching s_n from any one of the nodes in S^2_{ab} *), and
$A^*_3 = \min_{j=b+1}^{\ell} A[j, h-k-1]$ (* cost of reaching s_n from any one of the nodes in S^3_{ab} *).

Step 5: Let A^* corresponds to s_j $(a < j \leq \ell)$. We need to consider the following three cases for computing the cost of range assignment.

$j = b$**:** Here we have three parts in the cost: (i) from s_θ to s_b (in k hops), (ii) from s_b to s_n, and (iii) from s_a to s_1. Thus, the cost of range assignment is

$$cost = C[b, k] + A^* + B[a, h-k-1].$$

Note that, here the contribution of the bridge $\overleftarrow{s_a s_b}$ is not added in the cost due to the fact that, in A_1^* the range of s_b is greater than $d(s_a, s_b)$. Otherwise, in the path from s_b to s_n, it first hops at a radio station, say $s_{\ell'}$ where $\ell' < \ell$, and from $s_{\ell'}$ it reaches s_n using a $(h-k-1)$-hop path. Thus the cost of reaching s_n from $s_{\ell'}$ using a $(h-k-1)$-hop path is less than the cost of reaching s_n from s_b using $(h-k)$-hop path. Thus we have a contradiction that A^* corresponds to $j = b$.

$i \leq j < b$: Here we have four parts in the cost: (i) from s_θ to s_j (in h_j hops), (ii) from s_j to s_n (this enables s_b to get the message from s_θ), (iii) from s_b to s_a (using the bridge), and (iv) from s_a to s_1. Thus, the cost of range assignment is

$$cost = C[j, h_j] + w_b \times (d(s_a, s_b))^2 + A^* + B[a, h-k-1],$$

$a + 1 \leq j < i$ or $b < j \leq \ell$: Here also we have four parts in the cost: (i) from s_θ to s_b (in k hops), (ii) from s_j to s_n, (iii) from s_b to s_a (using the bridge), and (iv) from s_a to s_1. Thus, the cost of range assignment is

$$cost = C[b, k] + w_b \times (d(s_a, s_b))^2 + A^* + B[a, h-k-1].$$

The justifications of the above expression are as follows: If $j \geq b$, then the reason is trivial. Otherwise (i.e., if $j < b$) consider the path from s_θ to s_b. Let s_b is reached from s_j in k' hops. The actual desired situation is $h_j + k' = k$. If $h_j + k' < k$, then the produced cost may be dominated by another value of $k = h_j + k'$. But this does not create any problem due to the fact we are considering all possible values of $k = 1, 2, \ldots, h$. Below, we argue that $h_j + k' > k$ is not possible.

If $h_j + k' > k$, a node $s_{\ell'}$ exists on the path from s_j to s_b which is reached from s_θ in exactly k hops. The path from $s_{\ell'}$ to s_n is subsumed in the path from s_j to s_n, and hence the cost of the former one is less than that of the latter. Thus the assumption that A^* corresponds to s_j, is violated.

Step 6: If *cost* < *opt_cost* then

set *opt_cost* = *cost*, and the 5-tuple $(p, q, r, t, f) = (a, b, \ell, j^*, 0)$, where j^* is the index of the radio station for which the minimum occurs in the expression of A^*.

Step 7: If $d(s_b, s_{\ell+1}) < d(s_{a-1}, s_b)$ then

Assign $\rho(s_b) = d(s_b, s_{\ell+1})$ (* this also serves the role of right bridge $\overleftarrow{s_a s_b}$ *), and repeat Steps 2 to 5 for computing the optimum cost for assigning this range of $\rho(s_b)$. If the observed cost is less than *opt_cost*, then update *opt_cost* and the 5-tuple appropriately with the flag bit f set to 1.

The optimal range assignments of the radio stations can be done using the 5-tuple, and using the same technique as stated in Subsection 3.3.2.

3.4.3 Correctness and complexity

The correctness of the algorithm for computing optimum solution without any functional bridge is same as that of unbounded version of the problem (see Subsection 3.3.4). For the case where the solution contains functional bridge(s), the correctness follows from the Lemmas 3.2, 3.3, 3.4, and the justifications given in Step 5.

In order to analyze the time complexity of our algorithm, let us consider a right-bridge $\overleftarrow{s_a s_b}$, and an integer k ($1 \leq k < h$), where k is the number of hops needed to reach s_b from s_θ. Step 1 needs $O(n)$ time to compute ℓ. In Step 2, $C[b,k].\chi$ is used to compute the minimum cost of reaching from s_θ to s_b using k hops. Similarly, in Step 3, $B[a, h-k-1].\chi$ is used to compute the minimum cost of reaching from s_a to s_1 using $(h-k-1)$. Thus, Steps 2 and 3 need $O(1)$ time. Steps 4 needs $O(n)$ time to compute A^*. Steps 5 and 6 need $O(1)$ time. Step 7 may need an additional $O(n)$ time if $\rho(s_b)$ needs a minor increment maintaining the right bridge $\overleftarrow{s_a s_b}$. Thus, we have the following theorem.

Theorem 3.2 *The worst case time and space complexities of the weighted h-hop broadcast problem is $O(hn^3)$ and $O(nh)$ respectively.*

Proof: The minimum cost range assignment without any functional bridge can be computed in $O(n)$ time using the matrices A and B. In order to compute the solution with a functional bridge, we may need to consider $O(n^2)$ pairs of nodes in S as the possible right-most functional right-bridge (resp. left-most functional left-bridge), and for each such pair, the possible choices of k is at most $h-1$. As mentioned earlier, the time required for considering each triple (s_a, s_b, k) is $O(n)$. Thus, the time complexity result follows. It needs to be mentioned that, another $O(n)$ time pass is required to compute the range assigned to each node in S. The space complexity result follows from the storage requirement of the matrices A, B and C. $\square$

3.4.4 Further refinement

The time complexity of the Step 4 of the algorithm can be reduced if we can get the three quantities A_1^*, A_2^* and A_3^* on demand from a preprocessed data structure. In order to achieve this, we store each column (k) of the matrix A in a height balanced binary tree T_k whose each node is a tuple (s, c). The s fields of the leaf nodes contain $\{1, 2, \ldots, n\}$ in left to right order, and their c fields contain the corresponding elements of the k-th column of matrix A. The discriminant value stored at each internal node (say v) of T_k is the average of the s values stored in its inorder predecessor and successor nodes. The corresponding c field stores the minimum value of the c fields attached to all the nodes rooted at v in T_k.

Let us now fix a node s_b, $b > \theta$, and consider the minimum cost of range assignment with all possible right bridges $\overleftarrow{s_a s_b}$, where $a < \theta$. Let s_b be reached from s_θ in k hops. While computing the k-hop path from s_θ to s_b, we can get the set S_{ab}^1, and the number of hops needed for each node to reach from s_θ. Thus, we can also compute A_1^* in $O(|S_{ab}^1|)$ time using the matrix A as mentioned in Step 3 of the algorithm.

Now, assign different ranges to s_b for generating right bridges $\overleftarrow{s_a s_b}$ for different $a < \theta$. This determines the set S_{ab}^2 and S_{ab}^3. Now we can get A_2^* and A_3^* on demand, by searching in the tree T_k in $O(\log n)$ time. Note that, for a given k and b, A_1^* does not change. So, we can compute the cost of the range assignment for all the bridges initiated from b in $O(n \log n)$ time. Thus, we have the following theorem:

Theorem 3.3 *The worst case time and space complexities of the weighted h-hop broadcast problem is $O(hn^2 \log n)$ and $O(nh)$ respectively.*

3.5 Weighted Accumulation Problem

In this section, we propose an $O(n^2)$ time algorithm for solving the unbounded ($h = n-1$) version of the accumulation range assignment problem for linear radio network. This algorithm can be used to solve the same problem when radio stations are located in any arbitrary dimension. In this context, it needs to be mentioned that the existing $O(hn^3)$ time algorithm for the h-hop accumulation range assignment problem [39] works for the weighted version of the problem also.

Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of n radio stations located on a straight line and $W = \{w_1, w_2, \ldots, w_n\}$ be the weights of the radio stations in S. Without loss of generality, assume that $s^* = s_n$ is the target radio station. For solving the accumulation range assignment problem, we use the standard algorithm for arborescence problem on a digraph G as stated below.

Arborescence Problem : Given a weighted directed graph G and a root node s^*, the minimum cost arborescence problem is to find the minimum-cost spanning tree in G directed out from s^* (called arborescence tree).

In the accumulation range assignment problem, the ranges are assigned in such a way that s^* can be reached from each node in S, and the total cost of the entire network is minimum. Thus, in order to map this problem with the standard arborescence problem, the weight of each edge (s_i, s_j) must depend on the weight of node s_j (not on s_i). The formal algorithm is stated below.

Step 1: Let $G = (V, E)$ be a complete digraph, whose vertex set V corresponds to the set of radio stations S. The weight of a directed edge $(s_i, s_j) \in E$ is $w_j \times (d(s_i, s_j))^2$.

Step 2: Compute the arborescence tree T of G directed out from s^* using the $O(n^2)$ time algorithm proposed by Gabow et al. [59].

Step 3: Change the direction of each edge in T.

Step 4: For each edge $(s_i, s_j) \in T$ do $\rho(s_i) = d(s_i, s_j)$.

The correctness of the above algorithm follows from the definition of arborescence problem. Clearly the time complexity of the proposed algorithm is $O(n^2)$.

3.6 Summary

The weighted version of the range assignment problem in linear radio network is studied in the context of information *broadcast* and *accumulation*. The time complexity of our proposed algorithms for the unbounded and bounded (h-hop) versions of the broadcast problem are $O(n^2)$ and $O(hn^2 \log n)$ respectively. This improves time complexity of the existing results for the same two problems by a factor of n and $\frac{n^2}{\log n}$ respectively [5]. An $O(hn^3)$ time algorithm for the h-hop accumulation problem for linear radio network is available in [39]. We show that the unbounded version of the accumulation problem can be solved in $O(n^2)$ time. This algorithm works even when the radio stations are placed in $I\!R^d$. It needs to be mentioned that the status of the h-hop accumulation problem in $I\!R^d$ is not known for $d \geq 2$.

Chapter 4

Homogeneous 2-hop Broadcast in 2D

4.1 Introduction

In this chapter, we study the 2-hop broadcast range assignment problem in $I\!R^2$. Given a set $S = \{s_1, s_2, \ldots, s_n\}$ of n pre-placed radio stations and a source radio station $s^* \in S$, we consider the following two variations of minimum cost homogeneous (equal) range assignment problem for the 2-hop broadcast from s^* to all the members in S: (i) find the value of ρ such that 2-hop homogeneous broadcast from s^* is possible with minimum cost, and (ii) given a real number ρ, check whether homogeneous 2-hop broadcast from s^* to the members in S is possible with range ρ, and if so, then identify the smallest subset of S whom range ρ is to be assigned to accomplish the 2-hop broadcast from s^*. The first problem is optimally solved in $O(n^{2.376} \log n)$ time and $O(n^2)$ space. The second problem seems to be computationally hard. We present a 2-factor approximation algorithm for this problem, which runs in $O(n^2)$ time.

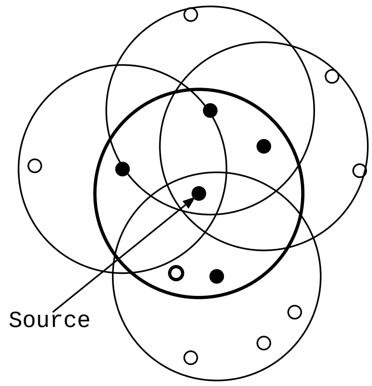


Figure 4.1: An example of 2-hop broadcast

4.2 Preliminaries

In a homogeneous radio network, each member $s_i \in S$ is either assigned a fixed range ρ or is not assigned any range. In the later case, s_i can not send message to other radio stations, but can receive message from other radio stations having non-zero range. Thus, the cost of a homogeneous radio-network is $k \times \rho^2$, where k is the number of radio stations having range ρ.

In Figure 4.1, an example is demonstrated for a given range value ρ. The black sites or the sites having thick boundary (except source s^*) indicate the subset of radio stations (called S_1) which are reachable from s^* in 1 hop, and the sites having thin boundary indicate the subset (called S_2) which are reachable from s^* in 2 hop. It is easy to understand that s^* must be assigned range ρ, and the members in S_1 lie inside the circle C^* having radius ρ and centered at s^*. Among the members in S_1, the black sites (denoted by S^*) are assigned range ρ for the optimum 2-hop broadcast from s^* to all the members in S, but those having thick boundary only, need not be assigned any range i.e., their range is zero. Thus, $S^* \subseteq S_1 \subset S$. All the members in S_2 lie outside C^*, and so, the range assigned to each members in S_2 is zero.

We consider the following two variations of the homogeneous 2-hop broadcast range assignment problem:

P1: Find the range value ρ that supports 2-hop broadcast from s^* to all the members in S, and the total cost of the entire network is minimum, and

P2: Given a range value ρ, check whether 2-hop broadcast is possible from s^* to all the members in S with range ρ, and if possible then identify the minimum cardinality subset $S^* \subset S$, whom range $(= \rho)$ is to be assigned to accomplish the 2-hop broadcast from s^* to all the members in S.

Problem P2 is a special case of the *minimum geometric disk cover problem* [33] as stated below.

Given two sets of points S_1 and S_2, the objective is to cover the points in S_2 by minimum number of disks of a given radius r and centered at the points in S_1.

It [68], it is shown that the minimum geometric disk cover problem admits a PTAS. Given a positive integer ℓ, the algorithm produces a solution with approximation factor $(1 + \frac{1}{\ell})^2$ in time $O(\ell^4 (2n)^{4\ell^2+1})$. Substituting $\ell = 2$, we have a $\frac{9}{4}$-factor approximation algorithm with worst case time complexity $O(n^{17})$. In [38], two algorithms for the problem P2 are proposed; the first one produces 6-factor approximation result in $O(n \log n)$ time and the second one produces 3-factor approximation result in $O(n \log^2 n)$ time. It can be shown that the time complexity of their second algorithm can easily be improved to $O(n \log n)$ using fractional cascading [27].

We give a very simple algorithm for problem P1 which runs in $O(n^{2.376} \text{log} n)$ time and $O(n^2)$ space. For problem P2, we propose a 2-factor approximation algorithm, which runs in $O(n^2)$ time and $O(n)$ space.

4.3 Problem P1

Throughout this chapter, we assume that the radio stations in S are numbered as $\{s_1, s_2, \ldots, s_n\}$ in increasing order of their distances from $s^*(= s_1)$. A radio station s_j is said to be *covered* by another radio station s_i with its assigned range ρ if $d(s_i, s_j) \leq \rho$. The 1-hop broadcast problem is trivial; here only one radio station s^* is to be assigned a non-zero range, and its value is $d(s^*, s_n)$.

Definition 4.1 *A range value ρ is said to be* feasible *if 2-hop broadcast from s^* to all the members in S is possible by assigning range ρ to s^* and some radio stations in S_1.*

Lemma 4.1 *For a range value ρ, its feasibility can be tested in $O(n \log n)$ time.*

Proof: We compute the sets S_1 and S_2 $(= S \setminus (S_1 \cup \{s^*\}))$ in $O(n)$ time. Next, we compute the Voronoi diagram of the points in S_1 in $O(n \log n)$ time [19]. For each point $p \in S_2$, we find its nearest point $q_p \in S_1$ by performing the planar point location [19] in the aforesaid Voronoi diagram. If $d(p, q_p) \leq \rho$ for all $p \in S_2$, then ρ is a feasible range for the 2-hop broadcast. The time complexity result follows from the fact that each planar point location query needs $O(\log n)$ time. $\square$

Fact 4.1 *If ρ is a feasible range for problem P1, then any range value ρ' greater than ρ is feasible.*

In the 2-hop broadcast range assignment problem from s^*, if the minimum cost is achieved for a range value ρ, then ρ must be the distance between a pair of members in S (see Lemma 2.1).

Lemma 4.2 *In the minimum cost range assignment, $|S^*| \leq 2$.*

Proof: Let ρ_{min} be the minimum feasible range for the 2-hop broadcast from s^*. It is obvious that if a range value $2\rho_{min}$ is assigned to s^*, then all the nodes in S can be reached from s^* in 1 hop, and the cost of this range assignment would be $4\rho_{min}^2$. If a lesser cost of 2-hop broadcast range assignment with range value ρ $(> \rho_{min})$ is possible, then $k\rho^2 < 4\rho_{min}^2$, where $k = |S^*| + 1$ (this includes s^* and the members in S^*). As $\rho > \rho_{min}$, we have $k \leq 3$, and hence $|S^*| \leq 2$. $\square$

Lemma 4.2 says that, for finding the optimum solution of problem P1, we need to compute the optimum costs of broadcast with $|S^*| = 0$, $|S^*| = 1$, and $|S^*| = 2$. The optimum cost with $|S^*| = 0$ (the 1-hop broadcast) is $(d(s^*, s_n))^2$. We separately compute the optimum cost with $|S^*| = 1$ and with $|S^*| = 2$. Finally, we choose the one having minimum cost. As a preprocessing, we execute the following steps:

- Compute an array $\mathcal{D}$ containing the distance of each pair of radio stations in S.
- Sort the array $\mathcal{D}$ in increasing order.
- Perform binary search to find the minimum feasible range ρ_{min} for 2-hop broadcast.
- Delete all the elements from the array $\mathcal{D}$ which are less than ρ_{min}. Thus, the array $\mathcal{D}$ contains all the feasible ranges for the 2-hop broadcast from s^*.

We consider $|S^*| = 1$, and $|S^*| = 2$ separately and compute the corresponding minimum value of the range for the 2-hop broadcast as follows:

Perform binary search in the array $\mathcal{D}$.
For each chosen ρ do
    compute S_1 and S_2 in $O(n)$ time, and
    execute the decision procedures for $|S^*| = \theta$, for $\theta = 1, 2$
    as described in the following two subsections.

4.3.1 Decision procedure for $|S^*| = 1$

For each member $s \in S_1$, compute $\delta_s = \max_{t \in S_2} d(s,t)$. Let $\Delta = \min_{s \in S_1} \delta_s$, and Δ corresponds to the radio station s' ($\in S_1$). For the given parameter (range) ρ, if $\Delta \leq \rho$ then decision procedure returns *true* with the radio station s'; otherwise it returns *false*.

4.3.2 Decision procedure for $|S^*| = 2$

For the given range ρ, let $|S_1| = k$ and $|S_2| = \ell$, where $s^* \notin S_1$ and $k + \ell = n - 1$. Let us name the members in S_1 as $\{s_{11}, s_{12}, \ldots, s_{1k}\}$ and the members in S_2 as $\{s_{21}, s_{22}, \ldots, s_{2\ell}\}$. We allocate a matrix M of size $k \times \ell$. Its (i,j)-th cell contains 1 if $d(s_{1i}, s_{2j}) \leq \rho$, otherwise it contains 0. Thus, $M[i,j] = 1$ implies s_{1i} can communicate with s_{2j} with range ρ.

The event $|S^*| = 2$ is *true* if and only if there exists a pair of rows, say a and b, of the matrix M such that $M[a,j] = 1$ and/or $M[b,j] = 1$ for every $j = 1, 2, \ldots, \ell$. This checking can easily be done in $O(n^3)$ time. We describe a faster algorithm for this decision procedure using matrix multiplication.

Let MC be the complement of matrix M, where $MC[i,j] = 1 - M[i,j]$, for each $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, \ell$. Now, compute the matrix product $P = MC \times MC'$, where MC' is the transpose of matrix MC. If $P[a,b] = \nu$ ($\neq 0$), then there exists ν many columns in the original matrix M where '0' is stored in both a-th and b-th row. In other words, if we assign range ρ to the radio stations s_{1a} and s_{1b}, then we can not cover ν many members in S_2. If $P[a,b] = 0$, then s_{1a} and s_{1b} (with range assignment ρ) can cover all the members in S_2. Thus, if there exists at least one '0' entry in the product matrix P then the procedure returns *true* with the corresponding radio stations. Otherwise, the procedure returns *false*. Here the following two important notes need to be considered

- If there exists a '0' entry in a diagonal element of the product matrix P, then $|S^*| = 1$. But, the decision procedure for the case $|S^*| = 1$ using matrix multiplication is expensive with respect to the running time. It also needs to be mentioned that, we need to compute the optimum cost for $|S^*| = 1$ and $|S^*| = 2$ separately. The reason is that, if we run the matrix multiplication based decision procedure for testing the feasibility of 2-hop broadcast for $|S^*| = 1$ and $|S^*| = 2$ together, it returns the minimum radius without considering whether $|S^*| = 1$ or $|S^*| = 2$. Suppose the reported radius corresponds to $|S^*| = 2$. There may exist situation where a minor increase in the radius may achieve the 2-hop broadcast with $|S^*| = 1$ with a lesser cost.

- While executing the decision procedure for $|S^*| = 2$, if the minimum element in the product matrix P is ν ($\neq 0$), this implies that the best choice of 2 radio stations in S_1 with range ρ can cover $(n - \nu)$ many elements in S.

Theorem 4.1 *The worst case time and space complexities of our proposed algorithm for problem P1 are $O(n^{2.376} \log n)$ and $O(n^2)$ respectively.*

Proof: The preprocessing step needs $O(n^2 \log n)$ time. The decision procedure for $|S^*| = 1$ needs $O(n^2)$ time. The time complexity for the decision procedure for $|S^*| = 2$ is dominated by that of matrix multiplication. The best known time complexity result for the matrix multiplication is $O(n^{2.376})$ [45]. We may need to call both the decision procedures at most $O(\log n)$ time. Thus, the time complexity result follows. The space complexity follows from the size of the array $\mathcal{D}$ and the matrices M, MC and MC'. $\square$

4.4 Problem P2

Now we consider a more generic problem where a range value ρ is given, and the problem is (i) to test whether the 2-hop broadcast from s^* to all other radio stations in

S is possible, and (ii) if possible then identify the minimum number of members in S_1 whom the range ρ need to be assigned for the 2-hop broadcast. The feasibility testing of range ρ in part (i) can be done in $O(n \log n)$ time (see Lemma 4.1).

In part (ii), let ρ be a feasible range for the 2-hop broadcast. Our objective is to identify the minimum number of elements in S_1 for the range assignment. It is very similar to the set-cover problem, and hence it seems to be computationally hard. We present a 2-approximation algorithm for this problem.

We use C_i to denote the circle of radius ρ, and centered at $s_i \in S$. If a pair of circles C_i and C_j intersect, then let $\alpha(C_i, C_j)$ and $\beta(C_i, C_j)$ denote the two intersection points among the boundaries of C_i and C_j, where $\alpha(C_i, C_j)$ lies on the left side of the directed line $\overrightarrow{s_i s_j}$, and $\beta(C_i, C_j)$ lies on the right side of $\overrightarrow{s_i s_j}$. We use $I(C_i, C_j)$ to denote the subset of S which are inside the intersection region of C_i and C_j.

Since 2-hop broadcast from s^* is possible with range ρ, for each member $s_i \in S_2$, $I(C_i, C^*)$ contains at least one member $s \in S_1$. We first apply the following steps for pruning S_2.

Pruning-Step-1: Identify the circles C_i ($s_i \in S_2$) which does not intersect any other circle C_j inside C^*. In other words, for each such s_i, $C_i \cap C_j \cap C^* = \phi$ for all $s_j \in S_2$, $j \neq i$. We assign range ρ to a radio station inside $C_i \cap C^*$ for covering s_i, and delete the radio station s_i from S_2.

Pruning-Step-2: Identify all the circles C_i ($s_i \in S_2$) such that $I(C_i, C^*) \supset I(C_j, C^*)$ for some C_j, $s_j \in S_2$. Here in order to cover s_j, we need to assign range ρ to a radio station in $I(C_j, C^*)$. This also covers s_i. Thus, we can delete the radio station s_i from S_2.

After execution of these pruning steps, let us consider the set of circles $\mathcal{C} = \{C_i | s_i \in S_2\}$. Here each circle C_i ($s_i \in S_2$) intersects with some other circle C_j ($s_j \in S_2$), $s_j \neq s_i$. For a pair of circles (C_i, C_j), if both $\alpha(C_i, C_j)$ and $\beta(C_i, C_j)$ lie inside C^*, then the pair (C_i, C_j) is said to be a *critical pair*; otherwise, the pair is said to be *non-critical*. Example of a critical pair is demonstrated in Figure 4.4(b).

4.4.1 Algorithm

We now describe a simple algorithm for problem P2, and show that if there exists no pair of radio stations $s_i, s_j \in S_2$ such that their corresponding circles C_i, C_j form a critical pair, then it produces optimum solution. We use this algorithm for designing the approximation algorithm for the general case.

Step 1: Let $\mathcal{L}$ be a list of circularly sorted points $\{\alpha(C_i, C^*) | s_i \in S_2\}$ in anticlockwise order. We use α_i to denote $\alpha(C_i, C^*)$.

Step 2: For each element $\alpha_i \in \mathcal{L}$, we execute the following steps.

Step 2.1: Assume that α_i is the starting position of the circular sorted list. Rename the circles corresponding to the members in S_2 such that the circle corresponding to $\alpha_{((i+k)\ mod\ |S_2|)}$ is assigned the name C_{k+1}, for all $k = 1, 2, \ldots, |S_2|$. Set $j = 1$.

Step 2.2: While $j \leq n$, execute the following steps:

(* Choose a member in S^* for range assignment *)

- Find an index k such that $\bigcap_{\ell=j}^{k} I(C_\ell, C^*) \neq \emptyset$ but $\bigcap_{\ell=j}^{k+1} I(C_\ell, C^*) = \emptyset$.
  (* We say, $C_j, C_{j+1}, \ldots, C_k$ satisfy *consecutive property* *)
  Let $s_p \in \bigcap_{\ell=j}^{k} I(C_\ell, C^*)$.
- Assign range ρ to the radio station s_p.
- set $j = k + 1$.

4.4.2 Correctness and complexity analysis for the restricted case

Theorem 4.2 *If for every pair of points $s_a, s_b \in S_2$, the corresponding circles C_a, C_b form a non-critical pair, then the algorithm proposed in Subsection 4.4.1 produces optimum result in the restricted case of problem P2.*

Proof: Let $S^* = \{s_1^o, s_2^o, \ldots, s_{opt}^o\}$ denote the set of radio stations having range ρ in the optimum solution, where $opt = |S^*|$. As mentioned earlier, s^* is not included in S^*. Let us consider a radio station $s_\theta^o \in S^*$. Let p be the point of intersection of the ray $\overrightarrow{s_\theta^o s^*}$ and the boundary of the circle C^*, and $\mathcal{C} = \{C_\mu | s_\mu \in S_2\}$ be the ordered set of the circles such that $\{\alpha(C_\mu, C^*) | s_\mu \in S_2\}$ are in the anticlockwise sorted order along the boundary of C^* starting from the point p. Assume that $s_\theta^o \in C_{j_1} \cap C_{j_2} \cap \ldots \cap C_{j_\nu} \cap C^*$, where $j_1 < j_2 < \ldots < j_\nu$. Among these, $C_{j_1}, C_{j_2}, \ldots, C_{j_\ell}$ are consecutive in the sense that $j_2 = j_1 + 1$, $j_3 = j_1 + 2, \ldots, j_\ell = j_1 + \ell - 1$, but $j_{\ell+1} \neq j_1 + \ell$, where $\ell < \nu$. In other words,

$$
\begin{aligned}
s_\theta^o &\in C_{j_1} \cap C_{j_1+1} \cap \ldots \cap C_{j_1+\ell-1} \cap C^* = A \text{ (say), but} \\
s_\theta^o &\notin C_{j_1} \cap C_{j_1+1} \cap \ldots \cap C_{j_1+\ell-1} \cap C_{j_1+\ell} \cap C^* = B \text{ (say).}
\end{aligned}
$$

In Figure 4.2, the regions A is shown using light shade and region B is shown using dark shade.

In the optimum solution, let $s_\gamma^o \in S^*$ cover the circle $C_{j_1+\ell}$. It may either lie in the region B or in region $C = (C_{j_1+\ell} \cap C^*) \setminus B$ (marked as a dotted region in Figure 4.2). In the former case, our algorithm chooses s_γ^o for assigning range ρ (instead of s_θ^o) to cover $C_{j_1}, C_{j_2}, \ldots, C_{j_\ell}, C_{j_\ell+1}$, and we are free to choose a radio station for covering $C_{j_{\ell+1}}, C_{j_{\ell+2}}, \ldots, C_{j_\nu}$. In the latter case, each of the circles $C_{j_{\ell+1}}, C_{j_{\ell+2}}, \ldots, C_{j_\nu}$ contains a part of A, and in turn contains region C also (see in Figure 4.2). Thus s_γ^o can cover all the circles $C_{j_{\ell+1}}, C_{j_{\ell+2}}, \ldots, C_{j_\nu}$. Thus, when an iteration (of Step 2.2) of our algorithm

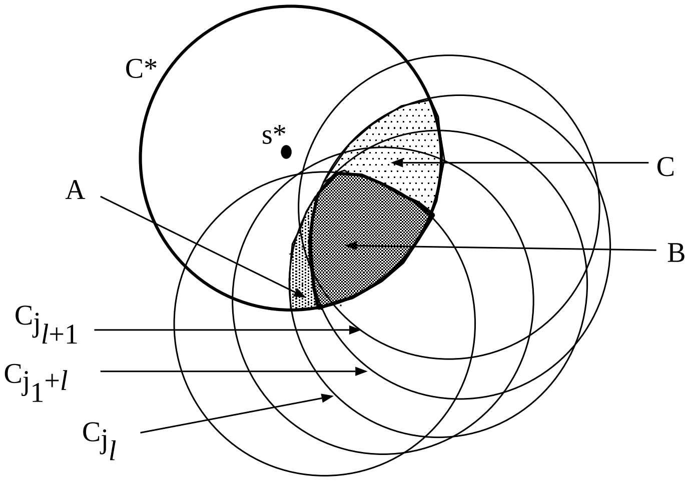


Figure 4.2: Proof of Theorem 4.2

starts from $\alpha(C_{j_1}, C^*)$, then it chooses $s^o_\theta \in A$ or some other radio station in the region A (satisfying consecutive property) for assigning range ρ. The correctness of our algorithm follows from the fact that, in both the cases the number of radio stations with range ρ obtained by our algorithm is less than or equal to $|S^*|$. Since S^* is the optimum solution, the solution obtained by our algorithm is also optimum. □

Theorem 4.3 *The time and space complexities of the proposed algorithm for problem P2 are $O(n^3)$ and $O(n)$ respectively.*

Proof: Step 1 needs $O(n \log n)$ time for the sorting. The number of iterations in Step 2 is $O(n)$. Since the time for computing the intersection of two ordered sets is $O(n)$, and in Step 2.2, $O(n)$ such intersections are computed in the worst case, the time complexity of Step 2 is $O(n^3)$ time. The space complexity follows from the fact that we need to store $\{\alpha(C_i, C^*) | s_i \in S_2\}$ in an array, and the circular scan needs only a constant number of extra space. □

Corollary 4.3.1 *If there exists a point p on the boundary of C^* which is not covered by any one of the circles $\{C_i | s_i \in S_2\}$, then in the restricted case, the optimum solution for problem P2 can be obtained in $O(n^2)$ time.*

Proof: Arrange the $\mathcal{L}$ list such that $\alpha(C_1, C^*)$ appears first in the list if we walk along the boundary of C^* in anticlockwise direction starting from p. Note that, here one iteration of Step 2.2 always starts from $\alpha(C_1, C^*)$ irrespective of where from the first iteration of Step 2.2 starts. Thus, only one iteration (Step 2) suffices to get the optimum solution, and it starts from $\alpha(C_1, C^*)$. Thus, the result follows. $\square$

Theorem 4.4 *In the restricted case, a solution of size at most* $|S^*| + 1$ *can be obtained in* $O(n^2)$ *time.*

Proof: Let us choose a radio station $s \in S_1$ which is farthest from s^*, and draw a ray $\overrightarrow{s^*s}$ which hits the boundary of C^* at p. If p is not covered by any circle C_i, $s_i \in S_2$, then from the Corollary 4.3.1, we get a solution of size $|S^*|$ in $O(n^2)$ time by running our algorithm.

If p is covered by some circle(s) corresponding to the members in S_2, then we assign range ρ to the radio station s ($\in S_1$). If the circle corresponding to s covers some member(s) in S_2, then delete those radio stations from S_2. Let the updated set be S_2'. Now, the following three situations may happen:

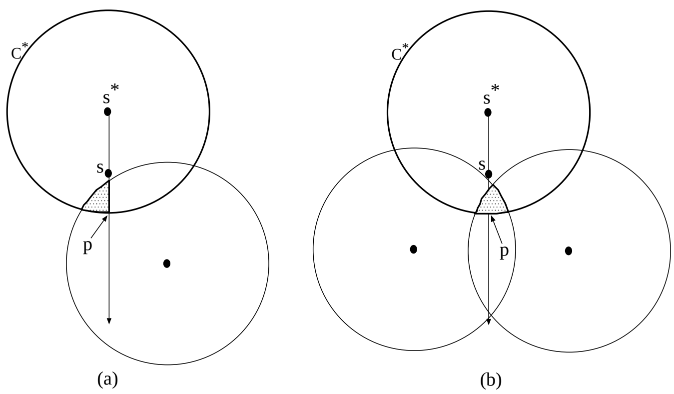


Figure 4.3: Proof of Theorem 4.4

(i) the point p is not covered by any circle corresponding to the members in S_2',

(ii) p is covered by circles corresponding to a member in S_2' such that these members in S_2' lie in the left (resp. right) side of the directed ray $\overrightarrow{s^*s}$. Note that, here the region of each of these circles inside C^* to the right (resp. left) side of the ray $\overrightarrow{s^*s}$ does not contain any point of S_1 (see the shaded region in Figure 4.3(a)). The reason is that we have chosen s to be the nearest from the boundary of C^*, and we have already applied **Pruning-Step-2** (see Section 4.4) on S_2.

(iii) p is covered by circles corresponding to two sets of members in S_2', such that these two sets lie in two different sides of the directed ray $\overrightarrow{s^*s}$. Using the same reason of case (ii), here also the intersection region of two circles of different set inside C^* does not contain any member of S_1 (see the shaded region in Figure 4.3(b)).

In Case (i), the point p is not covered by any member of S_2'. In the other two cases also, after assigning range ρ to the radio station s, we may consider that the point p is not covered by any circle corresponding to the members in S_2', and hence the optimum solution can be obtained in $O(n^2)$ time. Since, we have assigned range ρ to s, the number of radio stations with range ρ (not including s^*) is at most $|S^*| + 1$. $\square$

4.4.3 Approximation algorithm for the general case

The proposed algorithm in Section 4.4.1 may not produce optimum solution for a feasible range ρ if the radio stations are arbitrarily positioned. In Figure 4.4(a), the situation is explained, where $\{s^*, s_1, s_3\}$ indicates the optimum solutions, and $\{s^*, s_2, s_3, s_4\}$ is the output of our algorithm. We show that, the above method can be used for producing a 2-approximation result.

Lemma 4.3 *For a pair of points $s_i, s_j \in S_2$, if both $\alpha(C_i, C_j)$ and $\beta(C_i, C_j)$ are inside C^*, then $\angle s_i s^* s_j > \frac{\pi}{2}$.*

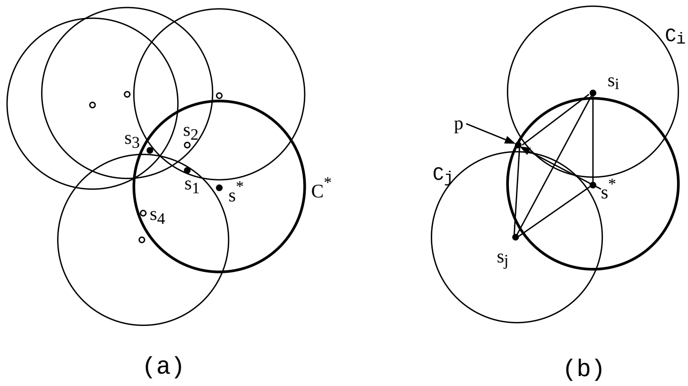


Figure 4.4: (a) Intuitive idea of the approximation algorithm, (b) proof of Lemma 4.3

Proof: Let p be the point on the boundary of C^* such that $p \notin C_i \cup C_j$ (see Figure 4.4(b)). Now, consider the triangle $\Delta s_i p s^*$, where $\overline{s_i p} > \rho$ (as p is outside C_i) and $\overline{ps^*} = \rho$. Thus, $\angle p s_i s^* < \angle s_i s^* p$. Similarly in the triangle $\Delta s_j p s^*$, $\angle p s_j s^* < \angle s_j s^* p$. Thus, in the triangle $\Delta s_i s^* s_j$, $\angle s_i s_j s^* + \angle s_j s_i s^* < \angle s_i s^* s_j$. This proves the lemma. $\square$

Based on Lemma 4.3, we modify the algorithm proposed in Section 4.4.1, so that it produces a 3-factor approximation result in $O(n^2)$ time.

We draw two mutually orthogonal lines L_1 and L_2 passing through s^*. This partitions the plane into four quadrants. Let S_1^k and S_2^k be respectively the radio stations of S_1 and S_2 in the k-th quadrant, $k = 1, 2, 3, 4$. By Lemma 4.3, C_i and C_j (corresponding to a pair of points $s_i, s_j \in S_2^k$) intersect, then at least one point of $\alpha(C_i, C_j)$ and $\beta(C_i, C_j)$ will lie outside C^*. Thus, if the algorithm presented in Section 4.4.1 is executed for S_2^k, it assigns range ρ to a minimum size subset of S_1 (of size χ_k say) for the 2-hop broadcast from s^* to the radio stations in S_2^k (see Theorem 4.2). Obviously, $\sum_{k=1}^{4} \chi_k \leq 4|S^*|$. Following theorem says that it is indeed a 3-approximation algorithm.

Theorem 4.5 $\sum_{k=1}^{4} \chi_k \leq 3|S^*|$.

Proof: Let S^{k*} denote the radio stations in the optimum solution S^* which lie inside the k-th quadrant, $k = 1, 2, 3, 4$. In our proposed algorithm, χ_k denotes the number of members chosen from S_1 for assigning range ρ to broadcast among the members in S_2^k. Consider the members in S_2^1. For any radio station $s_j \in S_2^1$, the circle C_j does not span inside the portion of C^* in 3-rd quadrant. Thus, the optimum solution for S_2^1 does not contain any member of S_1^3. This implies, $\chi_1 \leq |S^{1*}| + |S^{2*}| + |S^{4*}|$. Similarly, it can be proved that $\chi_2 \leq |S^{1*}| + |S^{2*}| + |S^{3*}|$, $\chi_3 \leq |S^{2*}| + |S^{3*}| + |S^{4*}|$, and $\chi_4 \leq |S^{1*}| + |S^{3*}| + |S^{4*}|$. Thus, $\sum_{k=1}^{4} \chi_k \leq 3(|S^{1*}| + |S^{2*}| + |S^{3*}| + |S^{4*}|) = 3|S^*|$. □

Theorem 4.6 *The time complexity of the proposed 3-approximation algorithm in the general case is $O(n^2)$.*

Proof: We run this algorithm for the subset $S_2^k \in S_2$ separately for $k = 1, 2, 3, 4$. While considering S_2^k, none of the arcs span to its diagonally opposite quadrant. By Corollary 4.3.1, here we need to execute only one pass instead of $|S_2^k|$ passes of Step 2. □

4.4.4 Improved analysis of the approximation factor

We now show that the algorithm proposed in Subsection 4.4.1 produces a 2-factor approximation result in $O(n^2)$ time.

Lemma 4.4 *For a triple of points $s_i, s_j, s_k \in S_2$, if all the points $\alpha(C_i, C_j)$, $\beta(C_i, C_j)$, $\alpha(C_j, C_k)$, $\beta(C_j, C_k)$, $\alpha(C_i, C_k)$ and $\beta(C_i, C_k)$ are inside C^*, then $C_i \cap C_j \cap C_k = \phi$.*

Proof: Since $\alpha(C_i, C_j)$, $\beta(C_i, C_j)$ are inside C^*, then $\angle s_i s^* s_j > \frac{\pi}{2}$ (by Lemma 4.3). Thus, $C_i \cap C_j$ lies entirely in the region A_1 as shown in Figure 4.5. Using the same argument, $C_j \cap C_k$ and $C_k \cap C_i$ lie in the regions A_2 and A_3 respectively. Since the regions A_1, A_2 and A_3 have only one common point, namely s^*, we may have $s^* \in C_i \cap C_j \cap C_k$. Again, as $s_i, s_j, s_k \in S_2$, $s^* \notin C_i, C_j, C_k$. Thus we have a contradiction. □

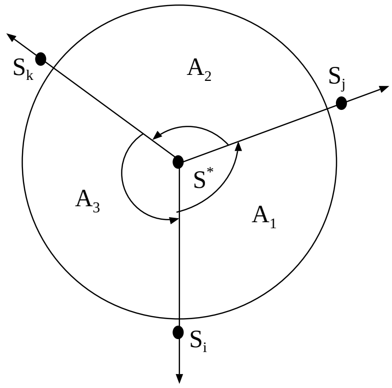


Figure 4.5: Proof of Lemma 4.4

Remark 4.1 *In Step 2.2 of the algorithm, we computed the maximum index k such that $\cap_{\ell=j}^{k} I(C_\ell, C^*) \neq \emptyset$, and then assigned range ρ to a radio station inside $\cap_{\ell=j}^{k} I(C_\ell, C^*)$, where C_j is the starting point of an iteration. Next, we started another iteration from C_{k+1}, and so on. Let the size of the solution obtained by our algorithm be Δ.*

Instead of that, if we choose a radio station inside $\cap_{\ell=j}^{k'} I(C_\ell, C^*)$, where $k' < k$, then the size of the solution obtained is greater than or equal to Δ.

Lemma 4.5 *If there exists a point p on the boundary of C^* which is not covered by any one of the circles $\{C_i | s_i \in S_2\}$, then the algorithm in Subsection 4.4.1 produces a 2-factor approximation result for the general case in $O(n^2)$ time.*

Proof: Sort the points $\{\alpha(C_i, C^*), s_i \in S_2\}$ in anticlockwise order starting from the point p. Let the corresponding order of the circles be $\{C_1, C_2, \ldots, C_{|S_2|}\}$. In the optimum solution, let C_1 be covered by a radio station $s_\theta \in S_1$. Consider the set of circles $\mathcal{C} = \{C_{i_1}, C_{i_2}, \ldots C_{i_k}, C_{j_1}, C_{j_2}, \ldots, C_{j_\ell}\}$ covered by s_θ, where $i_1(=1) < i_2 < \ldots < i_k < j_1 < j_2 < \ldots < j_\ell$. By Lemma 4.4, we may split these circles in at most two sets $\mathcal{C}_1 = \{C_{i_1}, C_{i_2}, \ldots C_{i_k}\}$ and $\mathcal{C}_2 = \{C_{j_1}, C_{j_2}, \ldots, C_{j_\ell}\}$ such that there is no critical pair inside $\mathcal{C}_1$ and $\mathcal{C}_2$. We show that our algorithm uses at most two radio stations to cover the circles in $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$. Let us now consider the circles in $\mathcal{C}_1$. If these are contiguous in the ordering of the circles then these are covered by a single radio station s_θ or some other radio station in their intersection zone. Otherwise, let μ be the minimum index such that $C_\mu \notin \mathcal{C}_1$, and C_μ lies between $C_{i_\delta}, C_{i_{\delta+1}} \in \mathcal{C}_1$ in the ordering of circles. As shown in the proof of Theorem 4.2, it is enough to consider the circles $C_{i_1}, C_{i_2}, \ldots, C_{i_\delta}$ which can be covered by a single radio station in our algorithm. The other circles $C_{i_{\delta+1}}, \ldots, C_{i_k} \in \mathcal{C}_1$ will be covered by the radio station used for covering C_μ in the optimum solution. Note that, we have not yet considered the covering of the members in $\mathcal{C}_2$. For this, we may assume that a new iteration of Step 2.2 of the algorithm starts from C_{j_1}, and it uses one radio station for covering the members in $\mathcal{C}_2$ or a part of it. By Remark 4.1, the size of the solution obtained here is greater than the size of the solution obtained by our algorithm.

We now eliminate $C_{i_1}, C_{i_2}, \ldots, C_{i_\delta}$ and proceed with the algorithm from C_μ. The same situation may arise with the radio station in the optimum solution that covers C_μ. Thus, for each radio station in the optimum solution, our algorithm chooses at most two radio stations for assigning range ρ in the worst case. Thus the approximation result follows.

The time complexity follows from the fact that only one iteration of Step 2 of the algorithm is executed here. $\square$

Theorem 4.7 *The algorithm in Section 4.4.1 produces a 2-factor approximation result for the general 2-hop broadcast problem.*

Proof: If there exists a point p on the boundary of C^* which is not covered by any circle corresponding to the members in S_2, then we can generate a solution of size $2|S^*|$ (excluding s^*) for the general problem (see Lemma 4.5). If no such point p is found, we can create such a point by assigning range ρ to a member in S_1 which is farthest from s^*. Since range ρ is assigned to s^* also, the size of the solution produced by our algorithm is at most $2|S^*| + 2$. The theorem follows from the fact that the size of the optimum solution is $|S^*| + 1$. $\square$

4.4.5 An efficient heuristic

In this section, we present an efficient heuristic algorithm for problem P2. We assume that a count field $COUNT(s)$ is attached with every member $s \in S_2$.

Step 1: For each member $s_i \in S_1$ do

If $d(s_i, s_j) \leq \rho$ for an element $s_j \in S_2$, then increase $COUNT(s_j)$

Step 2: For each member $s_j \in S_2$ having $COUNT(s_j) = 1$ do

Select the corresponding radio station (say s_i) in S_1, and assign range ρ to s_i; remove s_i from S_1 and all the radio stations of S_2 that are covered by the circle C_i (with radius ρ and centered at s_i).

Step 3: For each element $s_i \in S_1$, compute $A(s_i)$ = subset of S_2 whose elements are inside C_i.

For a pair of elements s_i and s_k, if $A(s_i) \subseteq A(s_k)$, then s_i can be removed from S_1, and

for each element $s_j \in S_2$, if C_j contains s_i then decrease $COUNT(s_j)$ by 1.

Step 4: Repeat Steps 2 and 3 as long as possible.

Finally, if all the points in S_2 are exhausted, then the optimal solution is achieved. Otherwise, we need to apply Step 5.

Step 5: (* Heuristic step *)

Repeat the following steps until the $COUNT$ field of at least one member in S_2 is reached to 1.

- Arbitrarily choose a member in $s_i \in S_1$, assign its range to '0', and delete s_i from S_1.
- if C_i contains a radio station $s_j \in S_2$, then $COUNT(s_j)$ is decreased by 1.

Again start executing from Step 2.

The time required for computing the $COUNT$ field for all the members in S_2 is $O(n^2)$. It is easy to observe that in each of the Steps 2 to 5, at least one point of either S_1 or S_2 is removed. After selecting a radio station in S_1 for the range assignment, the adjustment of the $COUNT$ field for the members in S_2 needs $O(n)$ time. Thus, the time complexity of the proposed heuristic algorithm is $O(n^2)$.

We have performed a detailed experiment in SUN BLADE 1000 machine with 750 MHz CPU speed and have used LEDA software [91]. We have considered different values on n, and for each n, we have generated 100 instances. We computed minimum feasible ρ value, say $\rho_{\min}$ for 2-hop broadcast, and the ρ value for 1-hop broadcast, say $\rho_{\max}$. Then we have generated a ρ value in $[\rho_{\min}, \rho_{max}]$ randomly. This execution time is not included in the running time of the experiment. We execute both the heuristic and approximation algorithms. Finally, in Tables 4.1 and 4.2 respectively, we report the average size of the solution (cardinality of the subset of S_1 having range ρ), and the running time for both the algorithms.

During the experiment, it is observed that in very few cases we need to apply Step 5 of the heuristic algorithm. This indicates that the optimum solution may not be obtained by this algorithm for all the instances. But, it is also observed that, if Step 5 is executed once, the control never came back to Step 5 again. Thus, we may infer that the solution obtained by this heuristic algorithm is very close to the optimum solution.

Table 4.1: Performance of our heuristic and approximation algorithms

No. of radio stations $\longrightarrow$ Solution produced by $\downarrow$	100	200	300	400	500	600	700	800	900	1000
Heuristic Algorithm	5.5	6.0	6.0	5.6	5.0	5.0	5.6	6.2	5.6	5.7
Approximation Algorithm	6.5	6.0	7.0	6.0	5.4	6.0	6.4	6.8	6.4	6.4

Table 4.2: Execution time of the heuristic and the approximation algorithms

No. of radio stations $\longrightarrow$ Running time (in sec.) of $\downarrow$	100	200	300	400	500	600	700	800	900	1000
Heuristic Algorithm	0.0	0.0	0.1	0.3	0.6	1.0	1.3	1.6	3.1	4.0
Approximation Algorithm	0.0	0.2	0.3	0.6	1.0	1.3	2.0	2.8	3.3	4.4

4.5 Summary

Given a set S of n pre-placed radio stations and a source station s^* in $I\!R^2$, we considered the following two variations of minimum cost homogeneous range assignment problem for the 2-hop broadcast from s^* to all the members in S: (i) find the value of ρ such that 2-hop homogeneous broadcast from s^* is possible with minimum cost, and (ii) given a real number ρ, check whether homogeneous 2-hop broadcast from s^* to the members in S is possible with range ρ, and if so, then identify the smallest subset of S whom range ρ is to be assigned to accomplish the 2-hop broadcast from s^*. The first problem is optimally solved in $O(n^{2.376}\log n)$ time and $O(n^2)$ space. For the second problem we presented a 2-factor approximation algorithm, which runs in $O(n^2)$ time. We have also proposed a heuristic algorithm for problem P2. Experimental evidences demonstrate that our heuristic algorithm runs very fast, and produces a optimum solution in most of the cases. Surely, in some instances it could not produce optimum solution, but the produced solution is very close to the optimum one. The proof of the computational hardness result of problem P2 is still undecided.

Chapter 5

All-to-all Communication in Linear Radio Networks

5.1 Introduction

In this chapter, we will study the unbounded version of the range assignment problem for all-to-all communication in linear radio network. Here a set $S = \{s_1, s_2, \ldots, s_n\}$ of n radio stations are arbitrarily placed on a line. The objective is to assign range $\rho(s_i)$ to each radio station $s_i \in S$ such that each radio station in S can communicate with the other members in S and the total power $(\sum_{s_i \in S}(\rho(s_i))^2)$ consumption is minimum. A simple incremental algorithm for this problem is proposed. It produces optimum solution in $O(n^3)$ time and $O(n^2)$ space. Thus, the earlier time complexity result on this problem is improved by a factor of n [76].

5.2 Preliminaries

As in Chapter 2, the members in $S = \{s_1, s_2, \ldots, s_n\}$ are assumed to be ordered from left to right on a line with s_1 at the origin. Given a range assignment $\mathcal{R} = \{\rho(s_1), \rho(s_2),$

$\ldots, \rho(s_n)\}$ of the radio stations in S, the communication graph is a directed graph $G = (V, E)$, where $V = S$, and $E = \{(s_i, s_j) | d(s - i, s_j) \leq \rho(s_i)\}$, as defined in Section 1.1.

Definition 5.1 *A communication graph G corresponding to a range assignment $\mathcal{R}$ is said to be* h-hop connected *if from each vertex $s_i \in S$ there exists a directed path of length less than or equal to h to every other vertex $s_j \in S$.*

For each radio station s_i, we maintain an array D_i which contains the set of distances $\{d(s_i, s_j), j = 1, \ldots, n\}$ in increasing order. Now we have the following lemma.

Lemma 5.1 *For any given h, if $\mathcal{R} = \{\rho_1, \rho_2, \ldots, \rho_n\}$ denotes the optimum range assignment of $\{s_1, s_2, \ldots, s_n\}$ for h-hop all-to-all communication then $\rho_i \in D_i$ for all $i = 1, 2, \ldots, n$.*

Proof: Same as in Lemma 2.1. $\square$

From now onwards, we restrict ourselves to the unbounded version of the problem, i.e., $h = n - 1$. Here the optimal solution corresponds to a range assignment such that the communication graph G is strongly connected, and the sum of powers $(\sum_{i=1}^{n}(\rho(s_i))^2)$ of all the radio stations in the network is minimum. The following two lemmas indicate two important features of the optimum range assignment.

Lemma 5.2 *Let ρ be the range assigned to a vertex s_i; s_ℓ and s_r be respectively the left-most and right-most radio stations such that $d(s_i, s_\ell) \leq \rho$ and $d(s_i, s_r) \leq \rho$. Now, if we consider the optimum range assignment of the radio stations $\{s_\ell, s_{\ell+1}, \ldots, s_i, \ldots, s_{r-1}, s_r\}$ only subject to the condition that $\rho(s_i) = \rho$, then (i) the range assigned to the radio station s_j is equal to $d(s_j, s_{j+1})$ for all $j = \ell, \ell + 1, \ldots, i - 1$, and (ii) the range assigned to the radio station s_k is equal to $d(s_k, s_{k-1})$ for all $k = i + 1, i + 2, \ldots, r$.*

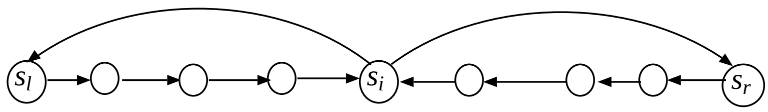


Figure 5.1: Illustration of Lemma 5.2

Proof: The feasibility of the range assignment as mentioned in the lemma follows from the fact that there are directed paths from s_ℓ to s_i and s_r to s_i in the corresponding communication graph. The optimality follows from the inequality $(p+q)^2 > p^2 + q^2$ for any two positive real numbers p and q (See Figure 5.1). □

Lemma 5.3 *In optimum range assignment* $\mathcal{R} = \{\rho_1, \rho_2, \ldots, \rho_n\}$*,* $\rho_1 = d(s_1, s_2)$ *and* $\rho_n = d(s_{n-1}, s_n)$*.*

Proof: On the contrary, let us assume that $\rho_1 = d(s_1, s_i)$, where $i > 2$. Now, we prove the lemma considering the following two cases : (i) $\rho_2 \leq d(s_2, s_i)$, and (ii) $\rho_2 > d(s_2, s_i)$. In Case (i), let us consider a modified assignment $\mathcal{R}' = \{\rho'_1, \rho'_2, \ldots, \rho'_n\}$, where $\rho'_1 = d(s_1, s_2), \rho'_2 = d(s_2, s_i), \rho'_3 = \rho_3, \rho'_4 = \rho_4 \ldots, \rho'_n = \rho_n$. Note that, the communication graph corresponding to the range assignment $\mathcal{R}'$ is still strongly connected, and $cost(\mathcal{R}') = cost(\mathcal{R}) - (d(s_1, s_i))^2 - \rho_2^2 + (d(s_1, s_2))^2 + (d(s_2, s_i))^2 = cost(\mathcal{R}) - 2d(s_1, s_2)d(s_2, s_i) - \rho_2^2 < cost(\mathcal{R})$. In Case (ii) also, let us consider a modified assignment $\mathcal{R}' = \{\rho'_1, \rho'_2, \ldots, \rho'_n\}$, where $\rho'_1 = d(s_1, s_2), \rho'_2 = \rho_2, \rho'_3 = \rho_3, \ldots, \rho'_n = \rho_n$. Note that, the communication graph corresponding to the new range assignment $\mathcal{R}'$ is still strongly connected, and $cost(\mathcal{R}') = cost(\mathcal{R}) - (d(s_1, s_i))^2 + (d(s_1, s_2))^2 = cost(\mathcal{R}) - 2d(s_1, s_2)d(s_2, s_i) - (d(s_2, s_i))^2 < cost(\mathcal{R})$. Therefore, in both the cases, there is another range assignment $\mathcal{R}'$ with $\rho_1 = d(s_1, s_2)$ whose cost is less than that of $\mathcal{R}$. The second part of the lemma can be proved in exactly similar manner. □

Our proposed algorithm is an incremental one. We denote the optimal range assignment of a subset $S_k = \{s_1, s_2, \ldots, s_k\}$ by $\mathcal{R}_k = \{\rho_1^k, \rho_2^k, \ldots, \rho_k^k\}$. Here the problem is: given $\mathcal{R}_j$ for all $j = 2, 3, \ldots, k$, obtain $\mathcal{R}_{k+1}$ by considering the next radio station $s_{k+1} \in S$.

An almost similar dynamic programming approach is used in [76] for solving the same problem in $O(n^4)$ time. Our approach is based on a detailed geometric analysis of the optimum solution, and it solves the problem in $O(n^3)$ time.

5.3 Method

We assume that for each $j = 2, 3, \ldots, k$, the optimal range assignment of $S_j = \{s_1, s_2, \ldots, s_j\}$ is stored in an array $\mathcal{R}_j$ of size j. The elements in $\mathcal{R}_j$ are $\{\rho_1^j, \rho_2^j, \ldots, \rho_j^j\}$, and $cost(\mathcal{R}_j) = \sum_{\alpha=1}^{j} (\rho_\alpha^j)^2$. The radio station s_{k+1} is the next element under consideration. An obvious choice of $\mathcal{R}_{k+1}$ for making the communication graph $G_{\mathcal{R}_{k+1}}$ strongly connected is $\rho_{k+1}^{k+1} = d(s_k, s_{k+1})$ and $\rho_k^{k+1} = \max(d(s_k, s_{k+1}), \rho_k^k)$. Lemma 5.4 says that this may not lead to an optimum result.

Lemma 5.4 $(d(s_k, s_{k+1}))^2 \leq cost(\mathcal{R}_{k+1}) - cost(\mathcal{R}_k) \leq (d(s_k, s_{k+1}))^2 + (\max(d(s_k, s_{k+1}), \rho_k^k))^2 - (\rho_k^k)^2$.

Proof: In $\mathcal{R}_{k+1}$, s_{k+1} will receive range equal to $d(s_k, s_{k+1})$ for connecting it with its closest member $s_k \in S_k$ (see Lemma 5.3). Thus, the left hand side of the inequality follows. The equality takes place when s_{k+1} is reachable from some member in S_k with its existing range assignment in $\mathcal{R}_k$. If this situation does not take place, then one needs to extend the range of some member in S_k to reach s_{k+1}. The inequality in the right hand side follows from the obvious choice s_k for which the range $\rho_k^k < d(s_k, s_{k+1})$, and is extended to $d(s_k, s_{k+1})$. Here, the equality takes place if $\rho_k^k \geq d(s_k, s_{k+1})$. $\square$

Illustrative examples are demonstrated in Figure 5.2, where the distance between each two consecutive nodes is shown along that edge; the range assignment for each node before and after inserting radio station s_5 are shown in parenthesis and square bracket respectively. From the left hand inequality of Lemma 5.4, the range of s_{k+1} (i.e., ρ_{k+1}^{k+1}) needs to be assigned to $d(s_k, s_{k+1})$ (see the range assigned to s_5 in both the figures). Now we analyze the different cases that may be observed in $\mathcal{R}_k$, and the actions necessary

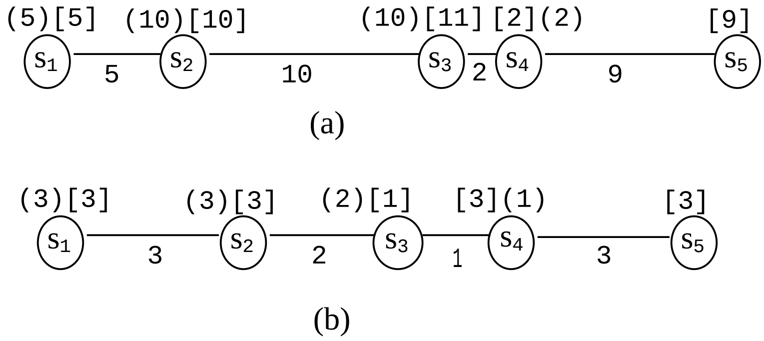


Figure 5.2: Proof of Lemma 5.4

for all those cases such that at least one member of S_k can communicate with s_{k+1} in 1-hop, and the total cost becomes minimum.

The simplest situation occurs if $d(s_i, s_{k+1}) \leq \rho_i^k$ for at least one $i = 1, 2, \ldots, k$. In this case, $\rho_i^{k+1} = \rho_i^k$ for all $i = 1, \ldots, k$. If $d(s_i, s_{k+1}) > \rho_i^k$ for all $i = 1, \ldots, k$, then we need to increase the range of some member in S_k for the communication from S_k to s_{k+1}. This may sometime need changes in different elements of $\mathcal{R}_k$ to achieve $\mathcal{R}_{k+1}$. We have demonstrated two examples in Figure 5.2, where the optimal range assignment of $\{s_1, s_2, s_3, s_4, s_5\}$ is obtained from that of $\{s_1, s_2, s_3, s_4\}$. The optimal range assignment in $\mathcal{R}_4$ and $\mathcal{R}_5$ are given in parenthesis and square bracket respectively. In Figure 5.2(a) the optimal range assignment is obtained by incrementing the range of s_3 only. But in Figure 5.2(b), in addition to incrementing the range of s_4, the range of s_3 is decremented to get the optimal assignment.

We use $\mathcal{R}_{k+1}^i$ to denote the optimum range assignment of the members in S_{k+1} subject to the condition that $\rho_i^{k+1} = d(s_i, s_{k+1})$. Now, $\mathcal{R}_{k+1}$ can be obtained by computing $\mathcal{R}_{k+1}^i$ for all $i = 1, 2, \ldots, k$, and then identifying an i^* such that $cost(\mathcal{R}_{k+1}^{i^*}) = \min_{i=1}^k cost(\mathcal{R}_{k+1}^i)$. We first describe a preprocessing step. Next, we describe in detail the computation of $\mathcal{R}_{k+1}^i$. Finally, we mention the order of invoking $\mathcal{R}_{k+1}^i$ for different values of i.

5.3.1 Preprocessing

In this step, we create the following two matrices using the given set of radio stations $S = \{s_1, s_2, \ldots, s_n\}$.

$T1$**:** It is an $n \times n$ matrix. Its (i, j)-th entry contains the index α ($i \le \alpha < j$) such that $d(s_\alpha, s_{\alpha+1}) = \max_{\beta=i}^{j-1} d(s_\beta, s_{\beta+1})$. Note that, T1 is a symmetric matrix.

$T2$**:** It is also an $n \times n$ matrix. Its (i, j)-th entry contains an index α such that if s_i is assigned the range $d(s_i, s_j)$ then s_i can communicate with s_α in 1-hop, but s_i can not communicate with $s_{\alpha-1}$ (resp. $s_{\alpha+1}$) in 1-hop depending on whether $i < j$ (resp. $i > j$).

Lemma 5.5 *Both the matrices $T1$ and $T2$ can be computed in $O(n^2)$ time.*

Proof: The results of the lemma follows from the computation order of the elements of $T1$ and $T2$. The computation order of i-th rows of $T1$ and $T2$ are as follows:

T1: $T1[i, i], T1[i, i + 1], \ldots, T1[i, n]$

T2: $T2[i, i], T2[i, i - 1], \ldots, T2[i, 1], T2[i, i + 1], T2[i, i + 2], \ldots, T2[i, n]$

The time complexity for computing i-row of $T1$ is O(n), which follows from the fact that (i) $T1[i, i] = 0$ and for $j > i$, $T1[i, j] = \max(T1[i, j - 1], d(s_{j-1}, s_j))$, and (ii) $T1$ is a symmetric matrix.

The time complexity for computing i-row of $T2$ is O(n), which follows from the fact that $T[i, i] = 0$ and (i) if $T2[i, \ell] = \alpha$ and $T2[i, k] = \beta$ where $k < \ell < i$, then $\beta \ge \alpha$, and (ii) if $T2[i, \ell] = \alpha$ and $T2[i, k] = \beta$ where $k > \ell > i$, then $\beta \le \alpha$. $\square$

5.3.2 Computation of R^i_{k+1}

First, we introduce the notion of *left-cover* and *right-cover* which will be used extensively in designing our algorithm.

Definition 5.2 *The* left-cover *of a radio station s_α for its assigned range ρ is the left-most radio station s_β which is reachable from s_α in 1-hop. Thus, s_β = left-cover(s_α, ρ), where $\beta \leq \alpha$ and $d(s_\alpha, s_\beta) \leq \rho < d(s_\alpha, s_{\beta-1})$. If $\beta = 1$ then the right-hand inequality condition is not required.*

Definition 5.3 *The* right-cover *of a radio station s_α for its assigned range ρ is the right-most radio station s_β which is reachable from s_α in 1-hop. Thus, s_β = right-cover(s_α, ρ), where $\beta \geq \alpha$ and $d(s_\alpha, s_\beta) \leq \rho < d(s_\alpha, s_{\beta+1})$. If $\beta = n$ then the right-hand inequality condition is not required.*

For notational convenience we use ρ_j to denote ρ^{k+1}_j, for $j = 1, 2, \ldots, k+1$. We first assign $\rho_i = d(s_i, s_{k+1})$ and $\rho_{k+1} = d(s_k, s_{k+1})$. Let s_ℓ = left-cover(s_i, ρ_i). This implies, s_i can communicate with all the radio stations $\{s_\ell, s_{\ell+1}, \ldots, s_{i-1}, s_i, s_{i+1}, \ldots, s_{k+1}\} = SS^i$ (say) in 1-hop, but s_i can not communicate with $s_{\ell-1}$ in 1-hop. Let us denote $SS^i_L = \{s_\ell, s_{\ell+1}, \ldots, s_{i-1}, s_i\}$, and $SS^i_R = \{s_{i+1}, s_{i+2}, \ldots, s_{k+1}\}$. Thus, we have $SS^i = SS^i_L \cup SS^i_R$.

By applying Lemma 5.2, we assign $\rho_j = d(s_j, s_{j-1})$ for all $s_j \in SS^i_R$, and $\rho_j = d(s_j, s_{j+1})$ for all $s_j \in SS^i_L \setminus \{s_i\}$. Due to this changed range assignment, none of the nodes in SS^i_R can communicate with a node to the left of s_i in 1-hop, but there may exist some member(s) in the set SS^i_L whose left-cover is in $S_{\ell-1}$. Let s_m be the left-most radio station such that s_m = left-cover(s_α, ρ_α) for some $s_\alpha \in SS^i_L$. We now need to consider the following two cases depending on whether (i) $m < \ell$ and (ii) $m = \ell$.

Case (i) $[m < \ell]$**:** Using the same argument as stated in Lemma 5.2, we further update the range of the radio station s_j to $\rho_j = d(s_j, s_{j+1})$ for all $j = m, m+$

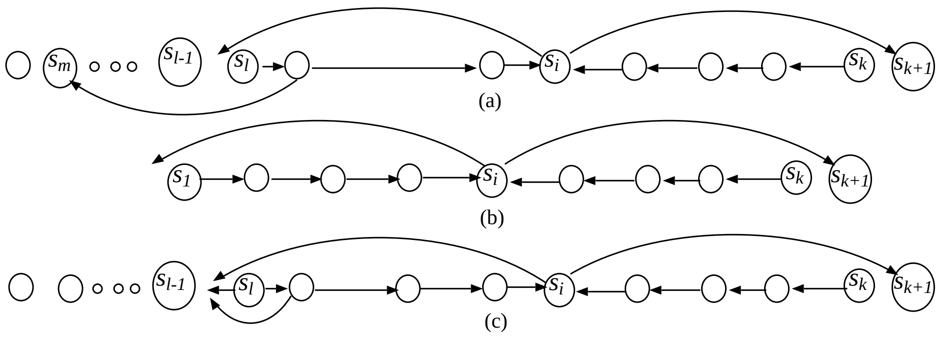


Figure 5.3: Illustration of (a) Case (i), (b) Case (ii) with $m = \ell = 1$, and (c) Case (ii) with $m = \ell > 1$

$1, \ldots, \ell - 1$ (see Figure 5.3(a)). This makes the communication subgraph with radio stations $\{s_m, s_{m+1}, \ldots, s_{\ell-1}, s_\ell, \ldots, s_i, \ldots, s_{k+1}\}$ strongly connected. This new assignment of range may cause some one to the left of s_m to be reachable in 1 hop from $\{s_m, s_{m+1}, \ldots, s_{\ell-1}\}$. We update the set SS^i_L to $SS^i_L \cup \{s_m, s_{m+1}, \ldots, s_{\ell-1}\}$. As a result, SS^i is also being updated accordingly, and m is considered to be as ℓ. Again, we need to consider one among the cases (i) and (ii). Note that, while calculating the left-cover of the updated set of nodes SS^i_L, we need to consider only the newly added nodes in SS^i_L.

Case (ii) $[m = \ell]$**:** Here several nodes in SS^i_L exist whose assigned range enables it to communicate with s_m in 1-hop but not with s_{m-1} (if exists). Thus, Case (i) fails to recur (see Figure 5.3(c)). Here $SS^i = \{s_m, s_{m+1}, \ldots, s_i, \ldots, s_k, s_{k+1}\}$, and m is referred to as the *maximal-left-cover*. The optimum range assignments for the radio stations in SS^i are as follows:

- $\rho_i = d(s_i, s_{k+1})$ (as assumed),
- $\rho_j = d(s_{j-1}, s_j)$ for all $j = i+1, i+2, \ldots, k+1$, and
- $\rho_j = d(s_j, s_{j+1})$ for all $j = m, m+1, \ldots i-1$.

Observation 5.1 *The left-cover of every member in SS^i with respect to the above range assignment lies inside SS^i.*

Now, two cases may arise depending on whether $m = 1$ or $m > 1$. For $m = 1$, the optimum range assignment $\mathcal{R}^i_{k+1}$ is already computed (see Figure 5.3(b)). However, if $m > 1$, we need to compute the range assignments of the members in S_{m-1} and establish communication among SS^i and S_{m-1}.

Let us now consider $\mathcal{R}_m$, and set $\rho_j = \mathcal{R}_m[j]$ for $j = 1, 2, \ldots, m-1$. Since $\mathcal{R}_m$ supports strong connectivity among the members in S_m, at least one member in S_{m-1} directly (in 1 hop) communicates with a member in SS^i with the range assignment $\mathcal{R}_m$. Let s_μ be the right-most member in SS^i which is directly (in 1 hop) reachable from a member $s_\nu \in S_{m-1}$. But, no element in SS^i can communicate with S_{m-1} with its presently assigned range. We now introduce the concept of *critical-gap* and use it to describe two procedures for restoring the strong connectivity in the entire S_{k+1}.

Definition 5.4 *Let $\{s_a, s_{a+1}, \ldots, s_b\}$ be a sequence of radio stations such that $\Delta = \max_{j=a}^{b-1} d(s_j, s_{j+1}) = d(s_\tau, s_{\tau+1})$ (say). Here, Δ is said to be the* critical-gap *of the sequence of radio stations $\{s_a, s_{a+1}, \ldots, s_b\}$.*

Lemma 5.6 *Let $(s_a, s_{a'})$ and $(s_b, s_{b'})$ be two pairs of radio stations such that $a < b' < a' < b$, and the range assigned to s_a and s_b be $\rho_a = d(s_a, s_{a'})$ and $\rho_b = d(s_b, s_{b'})$ respectively (see Figure 5.4(a)). If the critical-gap in $\{s_{b'}, s_{b'+1}, \ldots, s_{a'}\}$ is $d(s_\tau, s_{\tau+1})$, where $b' \leq \tau < a'$, then in the optimum (cost) range assignment of the radio stations $\{s_a, s_{a+1}, \ldots, s_{b'}, \ldots, s_{a'}, \ldots, s_b\}$, (i) $\rho_j = d(s_j, s_{j-1})$ for $j = a+1, a+2, \ldots, \tau$ and (ii) $\rho_j = d(s_j, s_{j+1})$ for $j = \tau+1, \tau+2, \ldots, b-1$.*

Proof: Since $\rho_a = d(s_a, s_{a'})$, $\rho_b = d(s_b, s_{b'})$ and $a < b' < a' < b$, the communication graph among the nodes $\{s_a, s_{a+1}, \ldots, s_b\}$ remains strongly connected if we choose an index $t \in [b', a']$ and assign (i) ρ_j is equal to $d(s_j, s_{j-1})$ for $j = a+1, a+2, \ldots, t$ and (ii) ρ_j is equal to $d(s_j, s_{j+1})$ for $j = t+1, t+2, \ldots, b-1$ (see Figure 5.4(b) for the demonstration). Thus, the total cost becomes $(d(s_a, s_{a'}))^2 + (d(s_b, s_{b'}))^2 + \sum_{j=a+1}^{t} (d(s_j, s_{j-1}))^2 +$

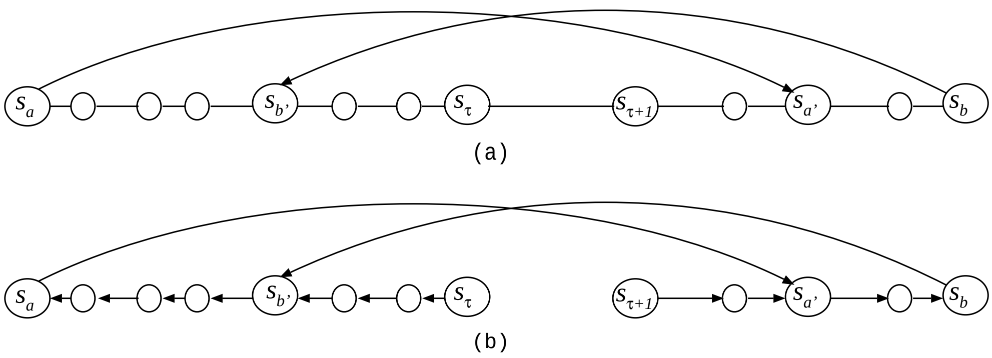


Figure 5.4: Proof of Lemma 5.6

$\sum_{j=t+1}^{b-1}(d(s_j, s_{j+1}))^2 = (d(s_a, s_{a'}))^2 + (d(s_b, s_{b'}))^2 + \sum_{j=a}^{b-1}(d(s_j, s_{j+1}))^2 - (d(s_t, s_{t+1}))^2$. As $d(s_\tau, s_{\tau+1})$ is the critical-gap, the minimum cost is achieved for $t = \tau$. □

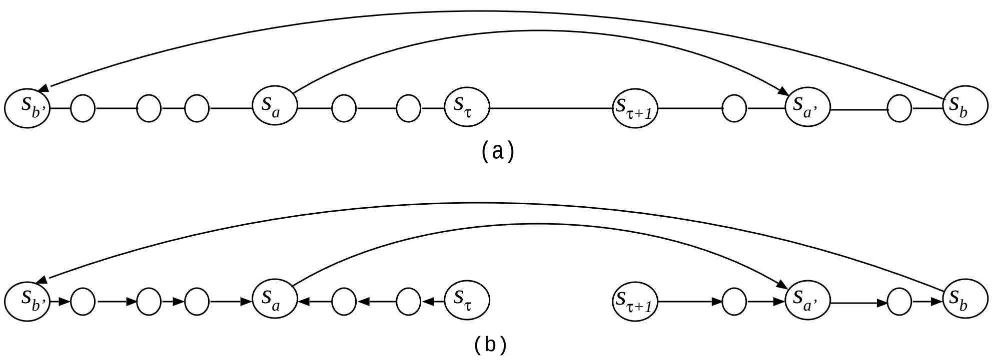


Figure 5.5: Proof of Lemma 5.7

Lemma 5.7 *Let $(s_a, s_{a'})$ and $(s_b, s_{b'})$ be two pairs of radio stations such that $b' < a < a' < b$, and the range assigned to s_a and s_b be $\rho_a = d(s_a, s_{a'})$ and $\rho_b = d(s_b, s_{b'})$ respectively (see Figure 5.5(a)). If the critical-gap in $\{s_a, s_{a+1}, \ldots, s_{a'}\}$ is $d(s_\tau, s_{\tau+1})$, then in the optimum (cost) range assignment of the radio stations $\{s_{b'}, s_{b'+1}, \ldots, s_b\}$, (i) $\rho_j = d(s_j, s_{j+1})$ for $j = b', b'+1, \ldots, a-1$, (ii) $\rho_j = d(s_j, s_{j-1})$ for $j = a+1, a+2, \ldots, \tau$, and (iii) $\rho_j = d(s_j, s_{j+1})$ for $j = \tau+1, \tau+2, \ldots, b-1$ (see Figure 5.5(b)).*

Proof: Proof is similar to Lemma 5.6. □

Recall that, s_μ is the right-most radio station in SS^i which is 1-hop reachable from a member of S_{m-1} with the existing range assignment. We now describe the following two procedures for establishing a connection from SS^i to S_{m-1}, and the necessary adjustments of the existing range assignments for reducing the overall cost. Note that, Procedure-1 is to be run if $\mu > m$. But Procedure-2 needs to be run always (irrespective of the value of μ).

Procedure-1

This procedure is applicable if $\mu > m$. Since the members in SS^i are strongly connected with their existing range assignments and $\mu > m$, there exists some radio station(s) to the right of s_μ whose assigned range enables it to reach a radio station to the left of s_μ. We assume that s_θ is the left-most one among such radio stations, where $m \leq \theta < \mu$. Thus, a situation as in Figure 5.4(a) (ignoring the suffixes of the radio stations) appears here. Let $\Delta = d(s_\tau, s_{\tau+1})$ be the critical-gap in $\{s_\theta, s_{\theta+1}, \ldots, s_\mu\}$. We apply Lemma 5.6 to update the range assignment as $\{\rho_j = d(s_j, s_{j-1}), j = \tau, \tau-1, \ldots, m\}$ (see Figure 5.4(b)). The range assignments of the other radio stations remain unchanged, and the strong connectivity of the entire S_{k+1} is restored. The cost of the updated range assignment is then computed and stored in a variable C^*. We also allocate another variable α^* and initialize it with 0. Here C^* and α^* are used respectively to store the optimum cost of $\mathcal{R}^i_{k+1}$ and the optimum choice of α whose range is to be increased for communication with S_{m-1}.

Note that, if $\mu = m$ then this procedure is not applicable. In that case, we initialize C^* by $\sum_{j=1}^{k+1} \rho_j^2$, where ρ_j is the presently assigned range of s_j; α^* is initialized with 0.

Procedure-2

This procedure is executed irrespective of whether $\mu = m$ or $\mu > m$. Here we restore the strong connectivity by increasing the range of a member $s_\alpha \in SS^i_L$ so that it can

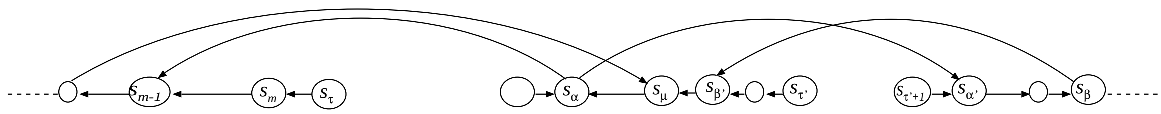


Figure 5.6: Updating range assignment using critical-gap

communicate with a member in S_{m-1}. We consider each member $s_\alpha \in SS_L^i$ separately, and increase its range to $\rho_\alpha = d(s_\alpha, s_{m-1})$. This needs updating the ranges of the radio stations in SS_L^i to achieve the minimum cost. We use $(\mathcal{R}_{k+1}^i \mid \rho_\alpha = d(s_\alpha, s_j))$ to denote the optimum range assignments for maintaining strong connectivity among the members in S_{k+1} with $\rho_i = d(s_i, s_{k+1})$ and $\rho_\alpha = d(s_\alpha, s_j)$.

Consider the computation of *cost*$(\mathcal{R}_{k+1}^i \mid s_\alpha = d(s_\alpha, s_{m-1}))$. Here, the following two instances are created where we need to compute the critical-gap for updating the ranges of the radio stations in SS_L^i.

The range assignments $\rho_\nu = d(s_\nu, s_\mu)$ $(\nu < m)$ and $\rho_\alpha = d(s_\alpha, s_{m-1})$ are such that, both s_ν and s_α can communicate with a non-empty subset of radio stations, namely $\{s_{m-1}, s_m, \ldots, s_\phi\}$, where $\phi = \min(\mu, \alpha)$. We compute the critical-gap $\Delta_1 = \max_{j=m-1}^{\phi-1} d(s_j, s_{j+1}) = d(s_\tau, s_{\tau+1})$ (say).

Let $s_{\alpha'} =$ *right-cover*(s_α, ρ_α), where $\rho_\alpha = d(s_\alpha, s_{m-1})$. As originally SS^i was strong connected, there must exist a radio station s_β $(\beta \geq \alpha')$ which can communicate with a node $s_{\beta'}$ (say) to the left of $s_{\alpha'}$ in 1 hop. In other words, $s_{\beta'} =$ left-cover(s_β, ρ_β). Thus, s_α and s_β can communicate with a non-empty subset of radio stations, namely $\{s_\psi, s_{\psi+1}, \ldots, s_{\alpha'}\}$, where $\psi = \max(\alpha, \beta')$. We compute the critical-gap $\Delta_2 = \max_{j=\psi}^{\alpha'-1} d(s_j, s_{j+1}) = d(s_{\tau'}, s_{\tau'+1})$ (say).

Next, we apply Lemma 5.6 and Lemma 5.7 adequately to revise the range assignments as follows (see Figure 5.6 for an illustration):

- $\rho_\alpha = d(s_\alpha, s_{m-1})$ (as assumed).
- $\rho_j = d(s_j, s_{j-1})$ for $j = \tau, \tau - 1, \ldots, m$.

• $\rho_j = d(s_j, s_{j-1})$ for $j = \tau', \tau' - 1, \ldots, \alpha + 1$.

• The range of other radio stations remain unchanged.

Note that, given $\rho_i = d(s_i, s_{k+1})$ and $\rho_\alpha = d(s_\alpha, s_{m-1})$, we have assigned ranges of $k-1$ radio stations in $(\mathcal{R}^i_{k+1} \mid \rho_\alpha = d(s_\alpha, s_{m-1}))$. It produces the minimum cost because

(i) we have chosen minimum cost range assignments for the $m-1$ radio stations $\{s_1, s_2, \ldots, s_{m-1}\}$ from $\mathcal{R}_m$,

(ii) the range of each of the remaining $(k-m)$ radio stations is equal to its distance from one among its two neighbors, and

(iii) we have $(k-m+2)$ such pairwise distances among the radio stations $\{s_m$, s_{m+1}, $\ldots$, $s_{k+1}\}$, and we have chosen $(k-m)$ such distances leaving the two critical-gaps Δ_1 and Δ_2 for assigning $\rho_\alpha = d(s_\alpha, s_{m-1})$ in the two side of s_α.

Some times the range $\rho_\alpha = d(s_\alpha, s_{m-1})$ is such that a very small further increase of ρ_α enables s_α to communicate with $s_{\alpha'+1}$ directly, and thus a larger critical-gap $d(s_{\alpha'}, s_{\alpha'+1})$ can be reduced from the total cost of range assignment. The following two lemmas indicate that only one more range $\rho_\alpha = d(s_\alpha, s_{\alpha'+1})$ of s_α need to be considered, and the situation where such a choice of ρ_α may produce lower cost.

Lemma 5.8 *If $d(s_\alpha, s_{m-1}) \leq d(s_\alpha, s_{\alpha'}) + C$, where $C = \max\{d(s_j, s_{j+1}) \mid j = m-1, m, \ldots, \alpha' - 1\}$, then $cost(\mathcal{R}^i_{k+1} \mid \rho_\alpha = d(s_\alpha, s_{\alpha'+1})) > cost(\mathcal{R}^i_{k+1} \mid \rho_\alpha = d(s_\alpha, s_{m-1}))$.*

Proof: Let $d(s_\alpha, s_{m-1}) = d(s_\alpha, s_{\alpha'}) + C_1$, $d(s_\alpha, s_{\alpha'+1}) = d(s_\alpha, s_{m-1}) + C_2$, and $D = cost(\mathcal{R}^i_{k+1} \mid \rho_\alpha = d(s_\alpha, s_{\alpha'+1})) - cost(\mathcal{R}^i_{k+1} \mid \rho_\alpha = d(s_\alpha, s_{m-1}))$. To prove $D > 0$ if $C_1 < C$.

Consider the Figure 5.7. Here
$D = [(d(s_\alpha, s_{m-1}) + C_2)^2 - (\delta_1)^2 - (\delta_2)^2] - [(d(s_\alpha, s_{m-1}))^2 - (\Delta_1)^2 - (\Delta_2)^2]$, where δ_1 and δ_2 are the critical-gap for assigning $\rho_\alpha = d(s_\alpha, s_{\alpha'+1})$ in the two sides of s_α.

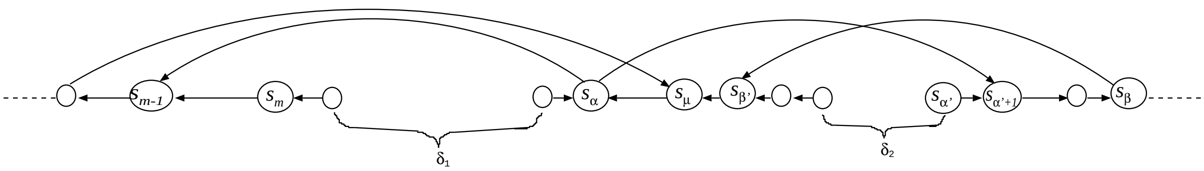


Figure 5.7: Increasing the range of s_α, and the resulting two critical-gaps

Since m is the maximal-left-cover, we have $\delta_2 = C_1 + C_2 < 2d(s_\alpha, s_{m-1}) - C_1$. We also have $\delta_1 < C_2$, since we have increased the range of s_α by an amount C_2.

On simplification of the expression of D, we have
$D = 2C_2 \times d(s_\alpha, s_{m-1}) + (C_2)^2 + (\Delta_1)^2 + (\Delta_2)^2 - (\delta_1)^2 - (\delta_2)^2 \geq (\Delta_1)^2 + (\Delta_2)^2 - (C_1)^2$.

Thus $D > 0$ if $C_1 < \max(\Delta_1, \Delta_2)$. As $C = \max\{d(s_j, s_{j+1}) \mid j = m-1, m, \ldots, \alpha' - 1\}$, we have $C = \max(\Delta_1, \Delta_2)$. Thus the lemma follows. □

Lemma 5.9 *If ρ_α is increased to communicate with S_{m-1}, then the possible values of ρ_α to be considered are $d(s_\alpha, s_{m-1})$ and $d(s_\alpha, s_{\alpha'+1})$.*

Proof: The first choice of the value of ρ_α is obvious. Let us now consider the second choice $\rho_\alpha = d(s_\alpha, s_{\alpha'+1})$. Let s_ψ = left-cover$(s_\alpha, d(s_\alpha, s_{\alpha'+1}))$. Following the same convention as in Lemma 5.8, let $C' = \max(\delta_1, \delta_2)$ and $d(s_\alpha, s_{\alpha'+1}) = d(s_\alpha, s_\psi) + C'_1$. From Lemma 5.8, we have $d(s_{\alpha'}, s_{\alpha'+1}) = C_1 + C_2$. Again since $\psi \leq m - 1$, we have $d(s_\alpha, s_{\alpha'+1}) - d(s_\alpha, s_\psi) \leq C_2$. The lemma follows from the fact that $C'_1 \leq C_2 < d(s_{\alpha'}, s_{\alpha'+1}) < C'$, since $d(s_{\alpha'}, s_{\alpha'+1}) \leq \delta_2$. □

Lemma 5.9 says that, we need to compute $cost(\mathcal{R}^i_{k+1} | s_\alpha = d(s_\alpha, s_{m-1}))$ and $cost(\mathcal{R}^i_{k+1} | s_\alpha = d(s_\alpha, s_{\alpha'+1}))$ for each s_α, $\alpha = m, m+1, \ldots, i$. At each step, if the minimum of these two costs is less than C^*, then C^* is updated accordingly, and α is also stored in α^*. Finally, $cost(\mathcal{R}^i_{k+1}) = C^*$. If $\alpha^* = 0$, we need to run Procedure-1 once again to get the optimum range assignment $\mathcal{R}^i_{k+1}$. Otherwise, we run Procedure-2 with $\alpha = \alpha^*$ to get $\mathcal{R}^i_{k+1}$.

5.3.3 Computation of $\mathcal{R}_{k+1}$

We are now in a position to present the stepwise description of our algorithm. In the preprocessing step, we create two matrices $T1$ and $T2$ of size $n \times n$ each as described in Subsection 5.3.1. Note that, if the (i,j)-th entry of the matrix $T2$ contains α and $i < j$ (resp. $i > j$) then s_α= left-cover($s_i, d(s_i, s_j)$) (resp. s_α = right-cover($s_i, d(s_i, s_j)$)). The input for computing $\mathcal{R}_{k+1}$ is $\{\mathcal{R}_j, j = 2, 3, \ldots, k\}$; these are computed in the previous $(k-1)$ iterations. The following two lemmas say that the computation of $\mathcal{R}_{k+1}$ can be made fast if $\mathcal{R}^i_{k+1}$ are executed for $i = k, k-1, \ldots, 1$ in order.

Lemma 5.10 *Let m and m' be the maximal-left-cover for $\mathcal{R}^i_{k+1}$ and $\mathcal{R}^j_{k+1}$ respectively. Now, if $i < j$ then $m \leq m'$. Furthermore, if s_ℓ = left-cover(s_i, ρ_i) and $\ell \geq m'$, then $m = m'$.*

Proof: The first part of the lemma trivially follows from the fact that if s_i is to the left of s_j and $d(s_i, s_{k+1}) > d(s_j, s_{k+1})$, then $SS^j \subseteq SS^i$. The second part follows from the fact that (a) in $\mathcal{R}^j_{k+1}$ the ranges assigned to each node $s_\beta \in SS^j_L(= \{s_m, s_{m+1}, \ldots, s_{j-1}\})$ is $d(s_\beta, s_{\beta+1})$, and (b) while computing $\mathcal{R}^i_{k+1}$, the range assigned to each node $s_\alpha \in \{s_\ell, s_{\ell+1}, \ldots, s_{i-1}\}$ is equal to $d(s_\alpha, s_{\alpha+1})$. Since $\ell \geq m'$, the repeated computation of left-cover will terminate after observing the maximal-left-cover $m = m'$. □

Lemma 5.11 *Let m' be the* maximal-left-cover *for $\mathcal{R}^j_{k+1}$. While computing $\mathcal{R}^i_{k+1}$ for some $i < j$, if $\ell \geq m'$, then cost($\mathcal{R}^i_{k+1}$) > cost($\mathcal{R}^j_{k+1}$).*

Proof: Let m be the maximal-left-cover for $\mathcal{R}^i_{k+1}$. As $i < j$, $\ell \geq m'$, we have $m = m'$. While increasing the range of s_α to ρ (= $d(s_\alpha, s_{m-1})$ or $d(s_\alpha, s_{\alpha'+1})$ as discussed in Lemma 5.9) to communicate with S_{m-1}, the critical-gap Δ_1 generated for both $\mathcal{R}^j_{k+1}$ and $\mathcal{R}^i_{k+1}$ become same. Let s_β = right-cover(s_α, ρ). If $\beta \leq i$, then Δ_2 value for computing both $\mathcal{R}^j_{k+1}$ and $\mathcal{R}^i_{k+1}$ become same. If $\beta > i$, then Δ_2

value for $\mathcal{R}_{k+1}^j$ is greater than Δ_2 value for $\mathcal{R}_{k+1}^i$ because in the former case Δ_2 is $\max\{d(s_\alpha, s_{\alpha+1}), d(s_{\alpha+1}, s_{\alpha+2}), \ldots, d(s_{\hat{\beta}-1}, s_{\hat{\beta}})\}$ (where $\hat{\beta} = \min(j, \beta)$) and in the latter case Δ_2 is $\max\{d(s_\alpha, s_{\alpha+1}), d(s_{\alpha+1}, s_{\alpha+2}), \ldots, d(s_{i-1}, s_i)\}$.

The lemma follows from the fact that $d(s_j, s_{k+1}) < d(s_i, s_{k+1})$ and Δ_2 value for $\mathcal{R}_{k+1}^j$ is greater than or equal to Δ_2 for $\mathcal{R}_{k+1}^i$. □

Lemmas 5.10 and 5.11 lead to the following conclusion towards accelerating the execution of the algorithm.

While computing $\mathcal{R}_{k+1}^i$ if (i) $cost(\mathcal{R}_{k+1}^{j^*}) = \min_{j=i+1}^{k} cost(\mathcal{R}_{k+1}^j)$ and the maximal-left-cover of s_{j^*} is s_{m^*} in the range assignment $\mathcal{R}_{k+1}^{j^*}$, (ii) the left-cover of s_i is s_ℓ for its range assignment $\rho_i' = d(s_i, s_{k+1})$, and (iii) $\ell > m^*$, then $cost(\mathcal{R}_{k+1}^i) > cost(\mathcal{R}_{k+1}^{j^*})$. So, we need not have to compute $cost(\mathcal{R}_{k+1}^i)$ in such a case.

5.3.4 Algorithm

We now give the stepwise description of the algorithm for computing $\mathcal{R}_{k+1}$. In addition to $\{\mathcal{R}_j, j = 1, 2, \ldots, k\}$, we need four scalar locations, namely *opt*, C^*, i^* and α^*, and two arrays R and LC, each of size n. The array R is used for generating $\mathcal{R}_{k+1}^i$, and the array LC contains the left-cover of some selected radio stations after assigning their ranges. More specifically, each element of the array LC is a tuple (a, b), where $m < a \leq i$ and s_b = left-cover(s_a, ρ_a). The first element of LC contains $a = i$, and the indices (a values) of only those radio stations are to be stored in LC such that the corresponding b values are in strictly decreasing order.

Step 1 Check whether there exists any radio station $s_i \in S_k$ whose range ρ_i ($\in \mathcal{R}_k$) is greater than or equal to $d(s_i, s_{k+1})$. If the check succeeds, then the algorithm terminates by copying the elements in $\mathcal{R}_k$ in first k elements of $\mathcal{R}_{k+1}$, and assigning $d(s_k, s_{k+1})$ to the $(k+1)$-th element of $\mathcal{R}_{k+1}$.

Step 2 If the check in Step 1 fails, then (* run the algorithm for computing $\mathcal{R}_{k+1}$ *)

- Initialize $opt \longleftarrow \infty$, $m \longleftarrow k+1$ and
- For each $i = k, k-1, \ldots, 1$, execute the following sub-steps to compute $\mathcal{R}_{k+1}^{i}$. (* This identifies an i^* such that the cost of $\mathcal{R}_{k+1}^{i^*}$ is minimum. As mentioned above, at each iteration (corresponding to each value of i) the array R will be used to generate $\mathcal{R}_{k+1}^{i}$. For the sake of notational simplicity, we will use ρ_j to denote $R[j]$ *).

Step 2.1 Compute $\ell = \text{left-cover}(s_i, d(s_i, s_{k+1})) = T2[i, k+1]$.

Step 2.2 Let m^* be the maximal-left-cover at the $(k - i^* + 1)$-th iteration, which has produced the optimum solution till the $(k-i)$-th iteration.

Now, if $\ell < m^*$ then execute the following steps (* if $\ell \geq m^*$, we need not have to process s_i (by Lemma 5.11) *).

Step 2.3 Initialize the elements of R as follows. During this process, we also compute the maximal-left-cover m and the array LC.

Step 2.3.1 Assign $\rho_i = d(s_i, s_{k+1})$; $LC\_ptr = 1$; $LC[1] = (i, \ell)$; $m = \ell$ and $\alpha = i$.

Step 2.3.2 Assign $\rho_j = d(s_j, s_{j-1})$ for $j = k+1, k, k-1, \ldots, i+1$.

Step 2.3.3 *for* $j = \alpha - 1, \alpha - 2, \ldots, \ell$ *do*

$\rho_j = d(s_j, s_{j+1})$ and $m = \textit{left-cover}(s_j, \rho_j)$.

if $m < LC[LC\_ptr].b$, *then* $LC\_ptr = LC\_ptr + 1$; $LC[LC\_ptr] = (j, m)$

endfor

Step 2.3.4 *if* $m < \ell$ *then* $\alpha = \ell$; $\ell = m$ and execute Step 2.3.3.

Step 2.3.5 Assign $\rho_j = j$-th element of $\mathcal{R}_m$ for $j = 1, 2, \ldots, m-1$.

Step 2.4 *if* $m = 1$, *then*

Compute $C = cost(R)$.

if $C < opt$, *then* assign $opt = C$, $i^* = i$ and exit from Step 2.

Step 2.5 Set the critical-gap $\Delta_1 = 0$.

Compute $\mu = \max\{\text{right-cover}(s_j, \rho_j)$, j=1, 2, ..., m-1$\}$.

(* S_m is strongly connected with range assignments $\mathcal{R}_m$. So $\mu \geq m$ *)

Step 2.6 (* Procedure-1: If $\mu > m$ then execute this step. *)

- (* Compute s_β, the left-most radio station which is 1-hop reachable from the radio stations to the right of s_μ including itself *)
  $TEMP = LC\_ptr$ (* LC_ptr will again be used in Procedure-2 (Step 2.7) *)
  while $LC[LC\_ptr].a < \mu$ do $LC\_ptr = LC\_ptr - 1$
  $\beta = LC[LC\_ptr].b$
  $LC\_ptr = TEMP$ (* Get back LC_ptr *)
- Assign $\delta = T1[\beta, \mu]$ and compute $\Delta_1 = d(s_\delta, s_{\delta+1})$ = critical-gap in $\{s_\beta, s_{\beta+1}, \ldots, s_\mu\}$.
- Revise the range assignment using the critical-gap Δ_1 as described in Lemma 5.6.
- Compute $C = cost(R) - (\Delta_1)^2 + (d(s_m, s_{m-1}))^2$
- If $C < C^*$ then set $C^* = C$, $\alpha^* = 0$ and $i^* = i$.

Step 2.7 (* Procedure-2 *) Increase the range of each member in $\{s_m, s_{m+1}, \ldots, s_i\}$ one by one for communication with S_{m-1}. Let s_α be under consideration.

Step 2.7a Increase the range of s_α to $\rho'_\alpha = d(s_{m-1}, s_\alpha)$.

(* Compute Δ_1 *)

- Assign $\beta = \min(\mu, \alpha)$
- Assign $\theta = T1[m-1, \beta]$ and compute $\Delta_1 = d(s_\theta, s_{\theta+1})$

(* Compute Δ_2 *)

Let $\alpha' = T2[\alpha, m-1]$, (* $s_{\alpha'} = \textit{right-cover}(s_\alpha, d(s_\alpha, s_{m-1}))$ *)

(* Compute $s_{\beta'}$, the left-most radio station which is 1-hop reachable from a radio station to the right of $s_{\alpha'}$ including itself. *)

- While $LC[LC\_ptr].a < \alpha'$ do $LC\_ptr = LC\_ptr - 1$
- $\beta' = LC[LC\_ptr].b$.
- If $\beta' \geq \alpha$ then set $\theta = T1[\beta', \alpha']$.
  Otherwise set $\theta = T1[\alpha, \alpha']$
- Compute $\Delta_2 = d(s_\theta, s_{\theta+1})$.

- Compute $C = cost(R) - (\rho_\alpha)^2 + (d(s_\alpha, s_{m-1}))^2 - (\Delta_1)^2 - (\Delta_2)^2$.
- If $C < C^*$, then set $C^* = C$, $\alpha^* = \alpha$ and $i^* = i$.

Step 2.7b Increase the range of s_α to $\rho'_\alpha = d(s_\alpha, s_{\alpha'+1})$.

(* Compute Δ'_1: Let s_δ be the left-most radio station which is 1-hop reachable from s_α *)

Compute $\delta' = \max(\nu, \delta)$

Assign $\theta' = T1(\delta', \beta)$ (* β is computed earlier *)

Compute $\Delta'_1 = d(s_{\theta'}, s_{\theta'+1})$

(* Compute Δ'_2: Let $s_{\beta'}$ be the left-most radio station which is 1-hop reachable from a radio station to the right of $s_{\alpha'+1}$ including itself. *)

- Compute $\Delta'_2 = \max(\Delta_2, d(s_{\alpha'}, s_{\alpha'+1}))$.
- Compute $C = cost(R) - (\rho_\alpha)^2 + (d(s_\alpha, s_{\alpha'+1}))^2 - (\Delta'_1)^2 - (\Delta'_2)^2$.
- If $C < C^*$, then set $C^* = C$, $\alpha^* = \alpha$ and $i^* = i$.

Step 3: If $C^* < opt$ then assign $opt = C^*$, and

repeat Step 2.1 to 2.7 with $\alpha = \alpha^*$, and copy the values of R in $\mathcal{R}_{k+1}$.

5.3.5 Correctness of the algorithm

The following lemma is relevant in the context of the proof of correctness of the algorithm.

Lemma 5.12 *While computing the maximal-left-cover for the range assignment $\mathcal{R}^i_{k+1}$, it is enough to consider $\rho(s_i) = d(s_i, s_{k+1})$ as the range of s_i.*

Proof: Consider a typical situation where m is the maximal-left-cover with $\rho(s_i) = d(s_i, s_{k+1})$. Here s_i covers s_ℓ towards its left, but not $s_{\ell-1}$ for a very small (ϵ) shortage of range, i.e., $d(s_i, s_{\ell-1}) - \epsilon < \rho(s_i) < d(s_i, s_{\ell-1})$. We will show that if such a case arises, then also we need not have to consider $d(s_i, s_{\ell-1})$ as a choice for computation of m (the maximal-left-cover). Here two cases need to be considered, namely $m < \ell$ and $m = \ell$.

In the first case, the maximal-left-cover computed using $\rho(s_i) = d(s_i, s_{k+1})$ will be the same as the *maximal-left-cover* with $\rho'(s_i) = d(s_i, s_{\ell-1})$. Thus, the range assignment of radio stations $S_{k+1} \setminus \{s_i\}$ using our algorithm will remain same for both the range assignments $\rho(s_i)$ and $\rho'(s_i)$. Thus, we will loose in terms of cost if we use $\rho'(s_i)$ instead of $\rho(s_i)$.

In the second case, for the assignment of $\rho'(s_i) = d(s_i, s_{\ell-1})$, we will surely get a maximal-left-cover m' where $m' \leq m$. Here the cost of the range assignments for the radio stations $S_{m-1} = \{s_1, s_2, \ldots, s_m\}$ using $\rho'(s_i)$ is greater than that using $\rho(s_i)$. The reason is that, in the former case we only use $\mathcal{R}^{m-1}_m$ (which in turn uses $\mathcal{R}_{m'}$), whereas in the latter case, we consider the optimal range assignment $\mathcal{R}_m$. In the part SS^i, surely, the range of one member in SS^i_L needs to be increased to reach s_{m-1}. Here the effect of increasing the range of s_i to $\rho'(s_i)$ is also considered. Thus the lemma follows. $\square$

Theorem 5.1 *Our proposed algorithm correctly computes $\mathcal{R}^i_{k+1}$.*

Proof: After assigning $\rho_i = d(s_i, s_{k+1})$ and computing the maximal-left-cover m, the cost of range assignments of SS^i is $\sum_{\alpha=m}^{i-1}(d(s_\alpha, s_{\alpha+1}))^2 + (d(s_i, s_{k+1}))^2 + \sum_{\alpha=i+1}^{k+1}(d(s_\alpha, s_{\alpha-1}))^2$, which is equal to $(d(s_i, s_{k+1}))^2$ + the sum of square of the length of the gap between each pair of consecutive radio stations. In order to assign ranges to the members in S_{m-1}, we have chosen the minimum cost range assignment $\mathcal{R}_m$. This ensures communication between S_{m-1} and SS^i. The communication from SS^i to S_{m-1} is established by increasing the range of only one radio station. Each element $s_\alpha \in SS_L^i$ is considered for this purpose. For each s_α, only two feasible choices of range (see Lemma 5.9) is considered, and the cost is computed by increasing its range and doing necessary modifications of the range of other radio stations considering two *critical-gaps* Δ_1 and Δ_2. Thus, the correctness of the algorithm follows. □

5.3.6 Complexity analysis

The worst case time complexity of computing $\mathcal{R}_{k+1}$ assumes the fact that no element $s_i \in S_k$ exists with $\rho_i \geq d(s_i, s_{k+1})$. If T_i denotes the time complexity of computing $\mathcal{R}_{k+1}^i$, then the total time complexity of computing $\mathcal{R}_{k+1}$ is $k \times \max_{i=1}^{k} T_i$. We now calculate the worst case value of T_i.

In Step 2, the computation of maximal-left-cover (m) needs $O(k+1-m)$ time. But, Lemma 5.10 says that, the total time needed for computing the maximal-left-cover for all the range assignments $\{\mathcal{R}_{k+1}^i, i = k, k-1, \ldots, 1\}$ is $O(k)$.

While computing $\mathcal{R}_{k+1}^i$ for some i, the worst case situation with respect to the time complexity arises when $m \neq 1$. Here, Steps 2.5 and 2.6 execute in $O(\mu)$ time using the preprocessed matrices $T1$ and $T2$. This may be $O(k)$ in the worst case.

Step 2.7 needs to be repeated for each $s_\alpha \in SS_L^i$ with two feasible ranges. In each case, the computation of critical-gap for s_α needs amortized $O(1)$ time using the array LC. Finally, Step 3 needs another $O(k)$ time. Thus, we have the following theorem stating the worst case time complexity of the algorithm.

Theorem 5.2 *The time complexity of our proposed algorithm for the optimal range assignment of the 1D unbounded range assignment problem is $O(n^3)$ in the worst case. The space complexity is $O(n^2)$.*

Proof: The preprocessing time complexity is $O(n^2)$. The above discussions say that $T_i = O(k)$ in the worst case. Thus, the time required for computing R_{k+1} is $O(k^2)$. The time complexity result follows from the fact that our incremental algorithm inserts n radio stations on the line one by one in order.

The space complexity result follows from the requirement of space for the matrices $T1$ and $T2$, and the space required for storing $\mathcal{R}_i$ for all $i = 1, 2, \ldots, n-1$ while computing $\mathcal{R}_n$. $\square$

5.4 Summary

The unbounded version of the range assignment problem for all-to-all communication in linear radio network is studied. An incremental algorithm for this problem is proposed. It uses dynamic programming paradigm, and produces optimum solution in $O(n^3)$ time and $O(n^2)$ space. This is an improvement in the running time by a factor of n over the best known existing algorithm for the same problem [76]. Two important properties of this problem are mentioned in Lemmas 5.10 and 5.11, but we could not use it for the further acceleration of the algorithm. We hope, a careful analysis using these results may improve both the time and space complexities of the problem.

Chapter 6

All-to-all Communication in 2D

6.1 Introduction

In this chapter, we extend the problem of Chapter 5 in 2D. Here, the given set $S = \{s_1, s_2, \ldots, s_n\}$ of n radio stations are placed in a 2D region, and an integer h is given. The objective is to assign range $\rho(s_i)$ to each radio station $s_i \in S$ such that each members of S can communicate with the others in at most h hops, and the total power consumption $(\sum_{s_i \in S}(\rho(s_i))^2)$ of the entire network is minimum. Here, we assume the value of distance power gradient β is equal to 2, but our proposed algorithms are valid for any $\beta \geq 1$. The general 2D h-hop range assignment problem is known to be NP-hard [40]. We first consider two simplified variations of the problem and propose efficient polynomial time algorithm for obtaining optimal solution. In the homogeneous case, where the range assigned to each radio station is same (ρ), we can obtain the minimum value of ρ in $O(n^3(\frac{\log \log n}{\log n})^{\frac{5}{4}}\log n)$ time in the worst case. In addition, if we consider the unbounded version ($h = n - 1$) of the homogeneous range assignment problem, then the optimal value of ρ can be obtained in $O(n^2 \log n)$ time. Finally, we propose an efficient heuristic algorithm for the general h-hop range assignment problem. Here the range of the radio stations may not be equal. Experimental results demonstrate that our heuristic algorithm runs fast and produces near-optimal solutions on randomly generated instances.

6.2 Homogeneous h-hop range assignment problem

Let ρ be the common range assigned to each radio station ρ. The objective is to find the minimum value of ρ such that each element $s_i \in S$ can communicate with all other radio stations in S in at most h hops. In other words, if we denote the communication graph corresponding to the range assignment $\{\rho, \rho, \ldots, \rho\}$ by G_ρ then the objective is to find a $\rho > 0$ such that there exists a directed path of length at most h between every pair of vertices in G_ρ, and for any $\epsilon > 0$ however small, there exists at least one pair of vertices s_i, s_j such that there does not exist any directed path of length at most h from s_i to s_j in $G_{\rho-\epsilon}$.

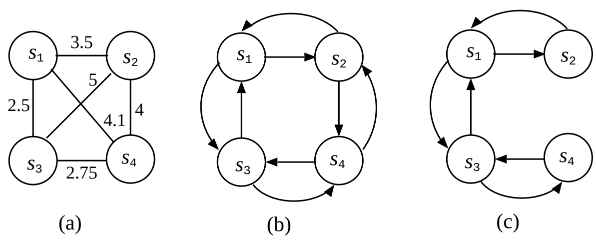


Figure 6.1: Demonstration of homogeneous range assignment

Figure 6.1 demonstrates the theme of the algorithm for homogeneous range assignment with n = 4 and h = 2. Figure 6.1(a) shows the distances between each pair of vertices. Figure 6.1(b) shows the communication graph corresponding to $\rho = 4$, where each node can communicate with every other nodes in 2 hops. Figure 6.1(c) shows the communication graph corresponding to $\rho = 3.5$, which is not 2-hop connected between every pair of radio stations.

6.2.1 Overview of the algorithm

For each $s_i \in S$, we maintain an array D_i which contains the set of distances $\{d(s_i, s_j), j = 1, \ldots, n\}$ in increasing order. Now we have the following lemma.

Lemma 6.1 *For any given h, if $\mathcal{R} = \{\rho(s_1),\ \rho(s_2),\ \ldots,\ \rho(s_n)\}$ denotes the optimum solution of the range assignment problem for h-hop all-to-all communication, then $\rho(s_i) \in D_i$ for all $i = 1, 2, \ldots, n$.*

Proof: Same as Lemma 2.1. □

In the homogeneous range assignment problem, the range ρ assigned to every element in S must be an element in $\cup_{i=1}^{n} D_i$ (by Lemma 6.1). We create an array D of size $\binom{n}{2}$ containing the elements in $\cup_{i=1}^{n} D_i$ in increasing order of their magnitude. We also allocate a $n \times n$ matrix to store the communication digraph. Now we need to consider two important things - (i) choose an element ρ in the array D, and (ii) check the h-hop connectivity using the following steps:

Step 1: Construct the graph G_ρ.

Step 2: Test whether the graph G_ρ is strongly connected.

Step 3: If the test in Step 2 returns *true*, then we compute a parameter $\Delta(\rho)$ of the digraph G_ρ as follows:

> Compute *all pair shortest paths* [64] in G_ρ. $\Delta(\rho)$ is the length of the longest one among these paths.

The parameter ρ is selected as follows: choose $\rho = D[2^\alpha]$, for $\alpha = 0, 1, \ldots, k$, where k is such that the $\Delta(\rho) > h$ for $\alpha = k - 1$, but $\Delta(\rho) \leq h$ for $\alpha = k$. Now, we need to perform a binary search among the indices $[2^{k-1}, 2^{k-1} + 1, \ldots, 2^k]$ to find an element k'

such that $\Delta(\rho) > h$ for $\rho = D[k' - 1]$, and $\Delta(\rho) \leq h$ for $\rho = D[k']$. Thus, we need to inspect at most $O(\log n)$ elements of the array D.

6.2.2 Complexity

Theorem 6.1 *The worst case time and space complexities for the homogeneous version of 2D h-hop range assignment problem are* $O(n^3(\frac{\log\log n}{\log n})^{\frac{5}{4}}\log n)$ *and* $O(n^2)$ *respectively.*

Proof: The space complexity follows from the fact that, for each required value of ρ, we need to create the graph G_ρ, which may need $O(n^2)$ space in the worst case. Step 1 needs $O(n^2)$ time for creating the graph. Step 2 can be executed in $O(|E|)$ time [37], where E is the set of edges in G_ρ. The best known algorithm for Step 3 runs in $O(n^3(\frac{\log\log n}{\log n})^{\frac{5}{4}})$ time [64]. Thus, the time required for Step 3 dominates the worst case time complexity for processing a single value of the parameter ρ. As we need to inspect $O(\log n)$ entries of the array D for finding the optimal values of ρ in the worst case, the overall time complexity of the algorithm is $O(n^3(\frac{\log\log n}{\log n})^{\frac{5}{4}}\log n)$. $\square$

Corollary 6.1.1 *The worst case time complexity of the unbounded homogeneous version of 2D range assignment problem is* $O(n^2\log n)$*.*

Proof: Follows from the fact that, here Step 3 of the algorithm is not required. For each required value of ρ, we need only to test the strong connectivity of the directed graph G_ρ, which needs $O(n^2)$ time [37]. $\square$

For the unbounded version of the problem, if the assumption of homogeneous range assignment is relaxed, a 2-approximation algorithm can easily be obtained in $O(n\log n)$ time using the following two steps:

Step 1: compute the the minimum spanning tree of the planar point set in $O(n\log n)$ time (see Theorem 1 of [50]).

Step 2: If s_i is connected to s_{j_1}, s_{j_2}, …, s_{j_k} in the minimum spanning tree, then assign $\rho(s_i) = \max(d(s_i, s_{j_1}),\ d(s_i, s_{j_2}),\ \ldots,\ d(s_i, s_{j_k}))$ [76].

6.3 General h-hop range assignment problem

The algorithm described in the earlier section works well for the general problem if the radio stations are uniformly distributed in the 2D plane. But, such an ideal situation is practically impossible due to various physical constraints, for example, presence of mountains/lakes, existing street layouts, conservation of large historical places etc. Thus, our earlier algorithm for homogeneous range assignment will not produce an optimum solution in general. Here the ranges assigned to different radio stations $S = \{s_1, s_2, \ldots, s_n\}$ may differ. Given a finite set S of n radio stations in 2D, and a positive integer h ($1 \leq h < n$), the 2D h-hop range assignment problem is NP-hard [40]. We describe an efficient heuristic approach for this problem. For small examples, the result produced by our algorithm is very close to the optimum solution obtained by exhaustive search. We have also performed simulation experiments for demonstrating the change in cost of range assignment for different values of h on the same set of radio stations. The change in cost for different n for a fixed value of h is also studied. In order to demonstrate the efficiency of our algorithm with respect to the quality of the solution and running time, we have run our algorithm on randomly generated large examples and compared the results with the 2-approximation algorithm based on minimum spanning tree for the unbounded version of 2D range assignment problem [76].

6.3.1 Proposed heuristic algorithm

We first compute the optimum solution for homogeneous h-hop range assignment problem. For each element $s_i \in S$, we have already computed the array D_i containing its distances from all other $n-1$ radio stations in increasing order. Next, we perform the following refinement steps to get solution for the general problem.

Step 1: For all the radio stations $s_i \in S$, if $\rho(s_i)$ does not match with any element in D_i, we update $\rho(s_i)$ by an element $\delta \in D_i$ which is largest among all the elements in D_i that are less than or equal to $\rho(s_i)$. It can be obtained by applying binary search in D_i.

This step does not change the communication graph. So, we can easily reduce the total cost of power assignment maintaining the h-hop connectivity.

Step 2: We compute the *gain* of reducing range (by one step) for each radio station $s_i \in S$ as follows: if $\rho(s_i) = D_i[j]$ (the j-th element in D_i), then $gain(s_i) = ((D_i[j])^2 - (D_i[j-1])^2)$. This indicates the reduction in the cost if the range of s_i is changed from $D_i[j]$ to $D_i[j-1]$. Note that, in this step we are not checking whether the h-hop connectivity is maintained or not due to this reduction of range of s_i. We store $gain(s_i)$ (along with i and j) for all $s_i \in S$ in a max-*heap*.

Step 3: Pick up (and delete) the maximum element from the *heap*. Let it corresponds to the radio station s_i. If $\rho(s_i) = D_i[j]$ (the j-th element in D_i), then replace $\rho(s_i)$ by $D_i[j-1]$, and check for h-hop connectivity. If this test succeeds we set $\rho(s_i) = D_i[j-1]$. Note that, $\rho(s_i)$ may further be reduced maintaining the h-hop connectivity. So, we compute $gain(s_i) = (D_i[j-1])^2 - (D_i[j-2])^2$, and put it in the *heap* along with i and $(j-1)$. If the test fails, the range of s_i can not be reduced further maintaining h-hop connectivity. Thus, $gain(s_i)$ need not be retained in *heap*.

Step 4: We iterate Step 3 repeatedly until the heap becomes empty.

Note that further reduction of total cost by applying of Step 3 will destroy the h-hop connectivity. But it may so happen that, a little increase in $\rho(s_i)$ for some

$s_i \in S$ creates a number of edges in the communication graph. This may allow reduction in the range assigned to many other elements of S (i.e., deletion of many edges from the communication graph) preserving the h-hop connectivity. This motivates us to execute Step 5 and 6 stated below.

Step 5: This step invokes Step 4 repeatedly for each $s_k \in S$. We use two scalars B and μ. B is initialized with 0. For each element $s_k \in S$ we inspect the following:

Assign $\rho(s_k) = D_k[n-1]$. This incorporates many edges in the communication graph. Thus, we may apply Step 4 repeatedly to observe the gain in the total cost. The total gain, if any, is compared with the existing value of B. If it is profitable, then the B is updated with the amount of gain, and k is stored in μ.

Step 6: We apply Step 5 repeatedly until no further gain is possible.

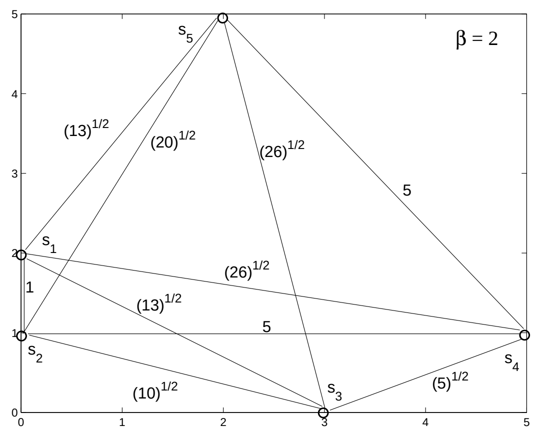
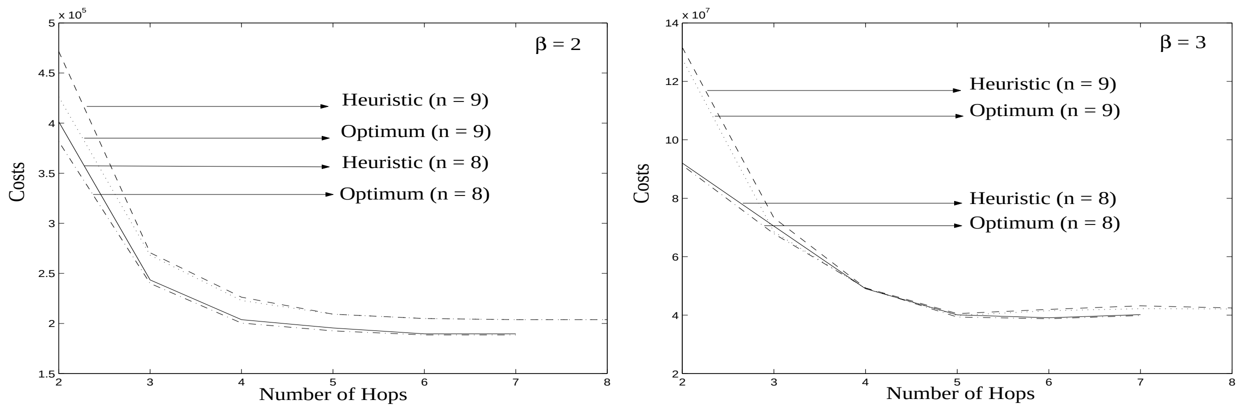


Figure 6.2: An instance of 2D range assignment

In Figure 6.2, an arrangement of 5 radio stations is shown in a 5×5 grid for the demonstration of our heuristic algorithm with $n = 5$ and $h = 2$. The initial homogeneous range assignment is $\{5, 5, 5, 5, 5\}$, and the cost is 125. After Step 1, the range vector becomes $\{\sqrt{13}, 5, \sqrt{13}, 5, 5\}$. This keeps the communication graph invariant, and the total cost becomes 101. After two iterations of Step 2, the resulting assignments are



(a) (b)

Figure 6.3: Comparison of heuristic estimate of power requirement and the optimal power requirement for different values of n and β

$\{\sqrt{13}, \sqrt{10}, \sqrt{10}, 5, 5\}$. This maintains 2-hop connectivity, and the cost reduces to 83. In Step 4, observe that an increase of the range of s_3 to $\sqrt{26}$, causes a decrease in total cost to 79; the corresponding range vector is $\{\sqrt{13}, \sqrt{10}, \sqrt{26}, \sqrt{5}, 5\}$.

6.4 Experimental Results

We have performed the experiment using DEC-ALPHA 233 MHz workstation. The points are generated on a 500×500 square grid. We have been able to compute the optimum solution of the 2D h-hop range assignment problem by performing exhaustive enumeration for $n = 8$ and $n = 9$. Figures 6.3(a) and 6.3(b) demonstrate the performance of our heuristic as compared to the optimum solution for $\beta = 2$ and $\beta = 3$ respectively (β is the distance power gradient in the cost function). It is observed that the solution produced by our heuristic is very close to the optimum solution in most of the cases. In particular, for large values of h the result of the two experiments produce almost same result.

Figure 6.4 demonstrates the change in the total cost of range assignment for different values of β. The four different plots correspond to (a) $\beta = 1$, (b)$\beta = 2$, (c) $\beta = 3$ and (d) $\beta = 4$. We have chosen different values of n ranging from $n = 10$ to $n = 30$, keeping h fixed. We have produced the result for $h = 2, 3, 4$ and 5. It is observed that, if the value of h is high, then the variation in the total cost does not vary much if n increases. But if h is small, then as n increases, the total cost increases significantly.

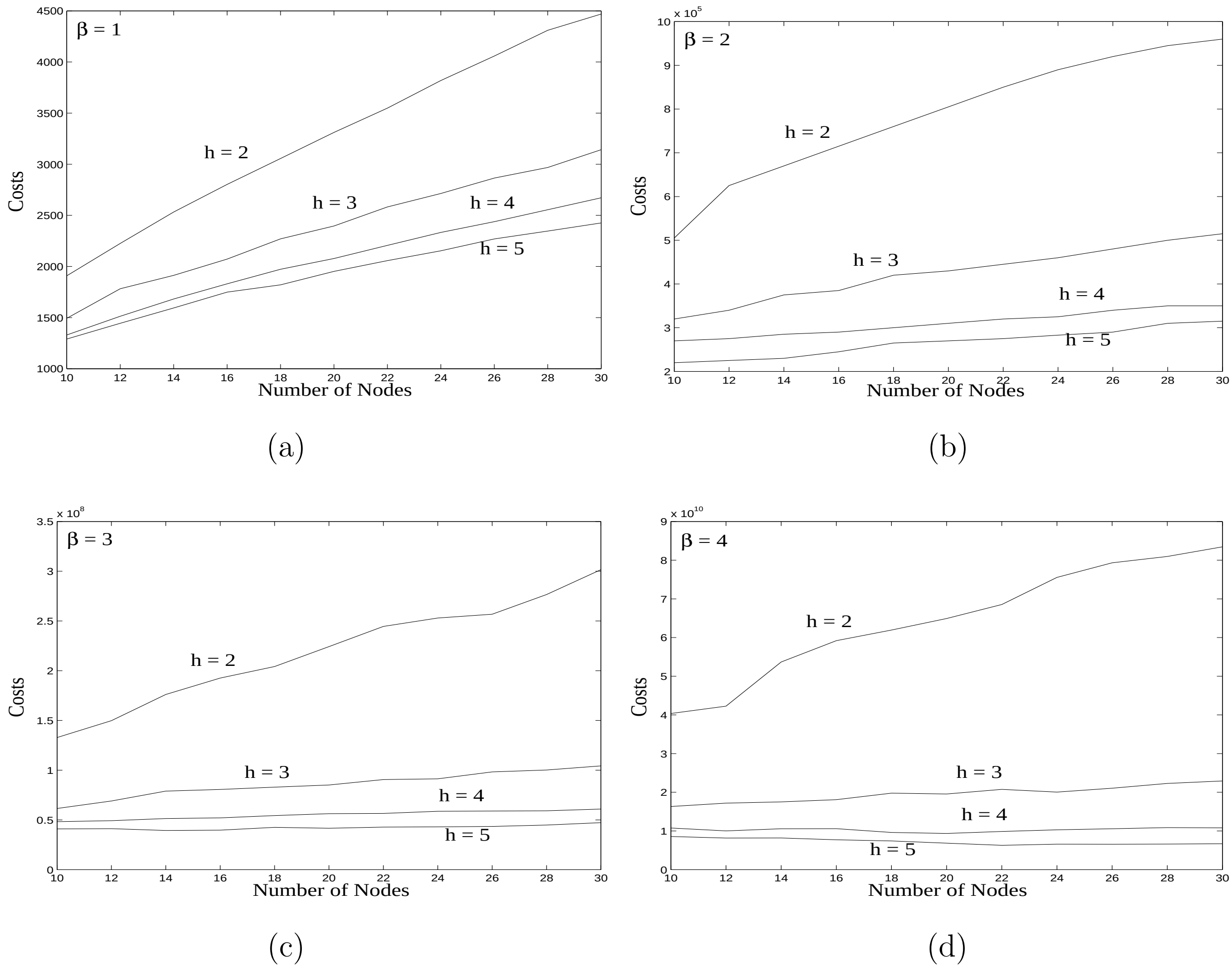


Figure 6.4: Plots of the power requirement vs. number of radio stations (for different values of h)

Figures 6.5 demonstrates the variation of total cost of range assignment for different values of β, namely (a) $\beta = 1$, (b) $\beta = 2$, (c) $\beta = 3$ and (d) $\beta = 4$. We have considered different values of h keeping n fixed ($n = 30, 40, 50$). It is obvious that for higher values of h the total cost for the range assignment should be less, and for lower values of h, the cost is high. But the important observation is that the each of these curves looks like a rectangular hyperbola. This implies, the product of the cost and the value of h is almost constant. The same observation is demonstrated in Figure 6.3.

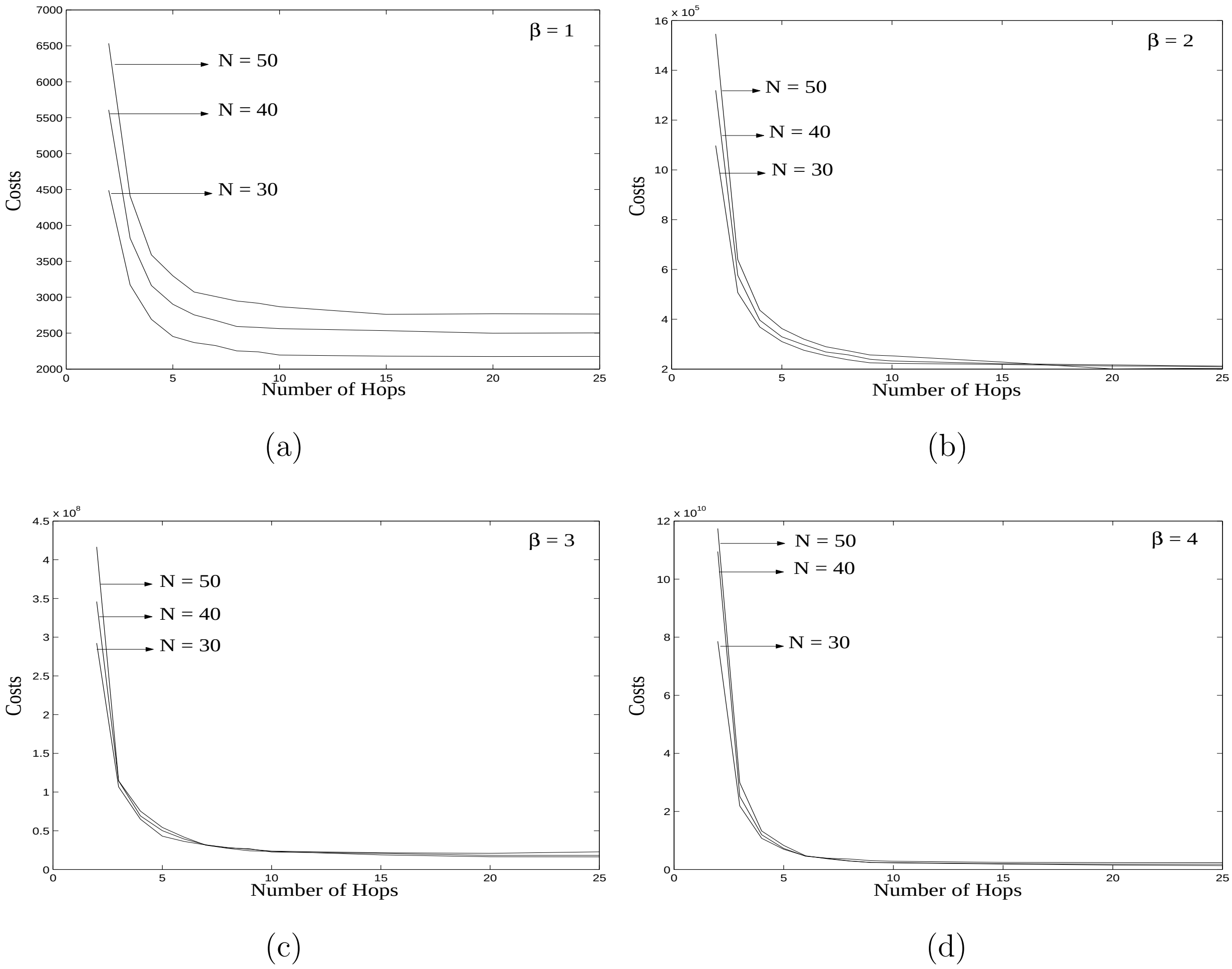


Figure 6.5: Comparison of the total power with h for fixed n

We now analyze the time requirement for running our heuristic algorithm. It starts with the homogeneous range assignment and tries to reduce the cost in successive steps keeping the h-hop connectivity preserved. The time requirement for the homogenous range assignment is $O(n^3(\frac{\log \log n}{\log n})^{\frac{5}{4}} \log n)$ and that of checking h-hop connectivity of the communication graph is $O(n^3(\frac{\log \log n}{\log n})^{\frac{5}{4}})$ at each step [64]. Thus the total time required for running our program depends on the number of iterations needed. Table 6.1 demonstrates the running time of our heuristic for different values of n and h.

Table 6.1: Variation of time (in second) with different n and h

	h = 2	h = 5	h = 10	h = 15	h = 20
n = 50	7.0	5.0	2.0	1.2	0.9
	h = 5	h = 10	h = 15	h = 20	h = 25
n = 75	35.0	11.0	6.0	4.5	4.0
	h = 10	h = 15	h = 20	h = 25	h = 30
n = 100	47.0	17.0	14.0	13.0	12.5

Each entry in Table 6.1 corresponds to the given value of n and h, and it shows the CPU time required for h-hop range assignment using our program if n points are distributed randomly over a square grid of size 500×500. As the execution time depends on the distribution of points, we consider 50 different instances and report the average time.

It has been observed that, for a fixed h, the time increases with n. This is obvious because it requires more time for both the initial homogenous range assignment and checking the h-hop connectivity. On the other hand, for fixed n, the time decreases as h increases. This is because, for a given n, the higher value of h needs less number of iterations in Step 3 and 5 of the heuristic. The same thing is observed in the homogeneous range assignment problem. Corollary 6.1.1 says that for the unbounded case, i.e., for $h = n - 1$, the time complexity is $O(n^2 \log n)$, whereas for a specific value of h, we may need $O(n^3(\frac{\log \log n}{\log n})^{\frac{5}{4}} \log n)$ time in the worst case.

In Figure 6.6, we compare the cost of range assignment using our heuristic with that using minimum spanning tree based 2-approximation algorithm [76]. For given n, we have generated several instances and have run both the algorithms. The plot in Figure

6.6 shows the normalized cost of MST based algorithm with respect to the cost of the solution produced by our proposed heuristic algorithm. Though our algorithm takes reasonably more time than the minimum spanning tree based heuristic, it outputs improved solution than the latter one.

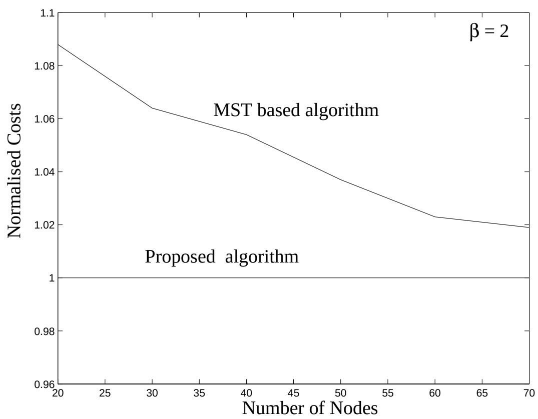


Figure 6.6: Comparison of results obtained by our algorithm with the MST based approximation algorithm

6.5 Summary

To our knowledge, no prior attempt is made for designing good heuristic algorithm for the generalized version of the 2D h-hop range assignment problem. Only one heuristic algorithm is available in the literature which considers a restricted case where $h = n-1$ and the symmetric connectivity is assumed [6]. But the performance of that algorithm is much worse than that of ours with respect to the running time. We hope that the running time can be improved by considering the exact geometry of the positioned radio stations.

Chapter 7

Base Station Placement Problem

7.1 Introduction

In this chapter, we have considered the base station placement problem in the context of mobile communication. The objective is to place a given number of base stations in a given convex region, and to assign range to each of them such that every point in the region is covered by at least one base station, and the maximum range assigned is minimized. It is basically covering a region by a given number of equal radius circles (see Figure 7.1), and the objective is to minimize the radius. We develop an efficient algorithm for this problem using Voronoi diagram [19]. Existing results for this problem are available when the region is a square [97] and an equilateral triangle [96]. The minimum radius obtained by our method favorably compares with the results presented in [96, 97]. The execution time of our algorithm is a fraction of a second, whereas the existing methods may even take about two weeks' time for a reasonable value of the number of circles (≥ 27) as reported in [96, 97].

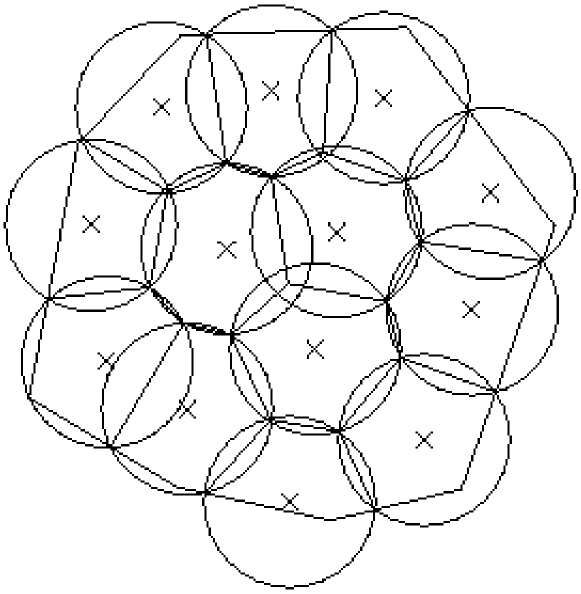


Figure 7.1: Illustration of our problem

7.2 Preliminaries

Consider a set of points $P = \{p_1, p_2, \ldots, p_k\}$ inside a convex polygon Π where the i-th base station is located at point $p_i \in \Pi$. We use $VOR(P)$ to refer the Voronoi diagram of the set of points P, and $vor(p_i)$ to denote the Voronoi cell corresponding to the point $p_i \in P$. Since we need to establish communication inside Π, if a part of the region $vor(p_i)$ goes outside Π for some i, then the region $vor(p_i) \bigcap \Pi$ is used as $vor(p_i)$.

Note that, all the points inside $vor(p_i)$ are closer to p_i than any other point $p_j \in P$, $j \neq i$. Thus, all these points communicate with p_i. As all the base stations are of equal range, our objective is to arrange the points in P inside the region such that the maximum range required (ρ) among the points in P is as minimum as possible. Our algorithm is an iterative one. At each step, it perturbs the point set P as described below, and finally, it attains a local minimum.

7.3 Algorithm

In each iteration, we compute $VOR(P)$, and then compute the circumscribing circle C_i of each $vor(p_i)$ using the algorithm proposed in [87], for each $i = 1, 2, \ldots, k$. Let r_i denote the radius of C_i. It is easy to understand that in order to cover a convex polygon by a base station with minimum range, we need to place the base station at the center of the circumscribing circle of that convex region, and the range assigned to that base station is equal to the radius of that circle. Thus, for each $i = 1, 2, \ldots, k$, we move p_i to the center of C_i and assign range r_i to it. Next, we compute $\rho = max\{r_i, i = 1, 2, \ldots, k\}$. The stepwise description of the algorithm for an iteration is given below.

Input: (* of the j-th iteration *) k points $P = \{p_1, p_2, \ldots, p_k\}$

If $j = 1$, then the members of P are k random points inside the polygon Π, and if $j > 1$, then the members of P are the output of the $(j-1)$-th iteration.

Output: (* of the j-th iteration *) The range ρ obtained in this iteration, and the centers of k circles for the next iteration.

Step 1: Compute $VOR(P)$.

Step 2: Compute the minimum enclosing circle C_i of $vor(p_i)$.

Step 3: Compute $\rho = \max_{i=1}^{k} r_i$, where r_i is the radius of the circle C_i.

Step 4: Output ρ, and the centers of C_i for $i = 1, 2, \ldots, k$.

Lemma 7.1 *At each iteration, (i) the newly assigned position of each point p_i lies inside the corresponding $vor(p_i)$, and (ii) the value of ρ decreases.*

Proof: (i) The smallest enclosing circle of a convex polygon either passes through the farthest pair of vertices of the polygon, and the line segment joining that pair of vertices define the diameter of the smallest enclosing circle, or it passes through three or more vertices of the polygon. In the first case, lemma obviously follows. In the second case, if the center lies outside the convex polygon, there exists a point-free arc of the enclosing circle having length greater than half of the perimeter of the said circle. Thus it is not the minimum enclosing circle of that convex polygon (see Chapter 16 of [109]).

(ii) Let $\{p_1, p_2, \ldots, p_k\}$ be the position of the base stations prior to an iteration, and ρ_{old} be the corresponding value of ρ. We have drawn the Voronoi diagram of $\{p_1, p_2, \ldots, p_k\}$ and then computed the smallest enclosing circle of each $vor(p_i)$. Let $\{p'_1, p'_2, \ldots, p'_k\}$ be the center of these circles, and C^* be the largest one among these circles. In other words, ρ_{old} is equal to the radius of C^*. In this iteration, the positions of the base stations are revised to $\{p'_1, p'_2, \ldots, p'_k\}$. The Voronoi polygon around each p'_i in the next iteration is obtained as follows:

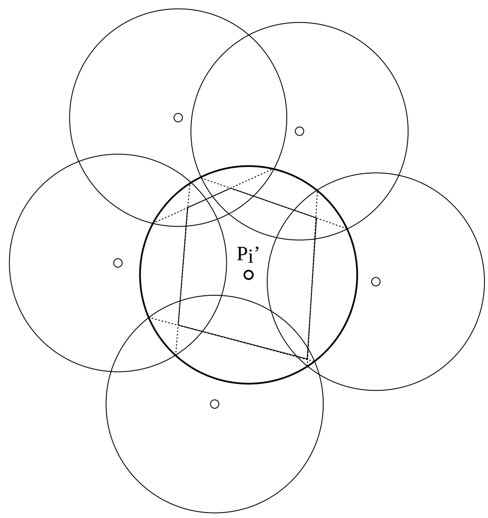


Figure 7.2: Illustration of $vor(p'_i)$

Draw copies of C^* with centers at $\{p'_1, p'_2, \ldots, p'_k\}$. Let C_i be the copy of C^* with center at p'_i. Consider the line segment defining the chords-of-intersection of each C_j with C_i for all $j \neq i, j = 1, 2, \ldots, k$, and the edges of the convex polygon Π. Next, compute the *envelope* of these line segments around p'_i. This defines the

Voronoi polygon $vor(p'_i)$ around p'_i for the next iteration. In Figure 7.2, the region bounded by the solid line segments demonstrates the voronoi region of p'_i, here the chords-of-intersection are shown using the dotted line segments.

Note that, $vor(p'_i)$ is properly inscribed by the corresponding circle C_i with center at p'_i, and having radius ρ_{old}. In the next iteration, we compute the smallest enclosing circle C'_i of each $vor(p'_i)$ which is completely enclosed in C_i. This proves that, if ρ_{new} is the revised value of ρ in the next iteration, then $\rho_{new} \leq \rho_{old}$. $\square$

Remark 7.1 *The iteration terminates when the value of ρ reaches to a local minima, or in other words, $\rho_{new} = \rho_{old}$ is attained.*

In order to come out from the local minima, we apply a refinement step during the iterative process. Note that, if a point (base station) p_i is on the boundary of Π, then at least 50% of the area of C_i lies outside Π, and hence this region need not be covered. This indicates the scope of further reduction in the area of C_i. Thus, if a point goes very close to the boundary of Π, we move it to the centroid of Π, which is computed as follows:

Let Π be a m vertex convex polygon, and (x_j, y_j) denote the j-th vertex of Π, $j = 1, 2, \ldots, m$. The centroid of Π is the point having the coordinates $(\frac{1}{m}\sum_{j=1}^{m} x_j, \frac{1}{m}\sum_{j=1}^{m} y_j)$. It can be shown that, the centroid of a convex region is always inside that region.

It is observed that, such a major perturbation brings the solution away from a local minima, and it leads to a scope of further reduction in ρ. We again continue the iteration with this initial placement until it again reaches another local minima.

The following theorem analyzes the time complexity of each iteration of our heuristic algorithm.

Theorem 7.1 *The worst case time complexity of an iteration is $O(n + k\log k)$.*

Proof: The factors involved in this analysis are as follows:

• Computing $VOR(P)$ - this can be done in $O(k\log k)$ time [19].

• $VOR(P)$ splits the convex polygonal region Π into k closed cells. Each edge of $VOR(P)$ appears in at most two cells. As the number of edges of the region Π is n, identifying these k cells need $O(n + k)$ time.

• Computing the minimum enclosing circle of a convex polygon needs time linear in its number of edges [87]. Thus, computing $\{C_i, i = 1, 2, \ldots, k\}$, needs $O(n + k)$ time. □

It is observed that the number of iterations needed to reach to a local optima from an initial configuration is reasonably small. The overall time complexity depends on the number of times we apply the refinement step.

7.4 Experimental results

An exhaustive experiment is performed with several convex shapes of the given region Π and with different values of k. It is easy to show that, for a given initial placement of P, at each iteration the value of ρ is decreased. As the process reaches a local minima, the quality of the result completely depends on the initial choice of the positions of P. We have studied the problem with random distribution of P. It shows that in an ideal solution, the distribution of points is very regular. So, while performing experiment with unit square region, we choose the initial placement of the points in P as follows: compute $m = \lfloor\sqrt{k}\rfloor$. If $m^2 = k$, we split the region into $m \times m$ cells, and in each cell place a point of P randomly. If $(k - m^2) < m$, then split the region into m rows of equal width. Then, arbitrarily choose $(k - m^2)$ rows and split each of these rows into $(m + 1)$ cells; the other rows are split into m cells. Now place one point in each cell. If $(k - m^2) > m$, then split the square region into $(m + 1)$ rows, and each row is split into m or $(m + 1)$ rows to accommodate all the points in P.

Table 7.1: Covering a unit square

k	ρ_{opt} using method in [97]	ρ^*_{opt} using our method	% increment
4	0.35355339059327376220	0.353553	0.0
5	0.32616054400398728086	0.326165	0.0
6	0.29872706223691915876	0.298730	0.0
7	0.27429188517743176508	0.274295	0.0
8	0.26030010588652494367	0.260317	0.0
9	0.23063692781954790734	0.230672	0.02
10	0.21823351279308384300	0.218239	0.0
11	0.21251601649318384587	0.212533	0.01
12	0.20227588920818008037	0.202395	0.06
13	0.19431237143171902878	0.194339	0.01
14	0.18551054726041864107	0.185527	0.01
15	0.17966175993333219846	0.180208	0.30
16	0.16942705159811602395	0.169611	0.11
17	0.16568092957077472538	0.165754	0.04

k	ρ_{opt} using method in [97]	ρ^*_{opt} using our method	% increment
18	0.16063966359715453523	0.160682	0.03
19	0.15784198174667375675	0.158345	0.32
20	0.15224681123338031005	0.152524	0.18
21	0.14895378955109932188	0.149080	0.08
22	0.14369317712168800049	0.143711	0.01
23	0.14124482238793135951	0.141278	0.02
24	0.13830288328269767697	0.138715	0.30
25	0.13354870656077049693	0.134397	0.63
26	0.13176487561482596463	0.132050	0.23
27	0.12863353450309966807	0.128660	0.02
28	0.12731755346561372147	0.127426	0.08
29	0.12555350796411353317	0.126526	0.77
30	0.12203686881944873607	0.123214	0.96

Table 7.2: Covering a equilateral triangle

k	ρ_{opt} using method in [96]	ρ^*_{opt} using our method	% increment
4	0.2679491924311227065	0.267972	0.01
5	0.2500000000000000000	0.250006	0.0
6	0.1924500897298752548	0.192493	0.02
7	0.1852510855786008545	0.185345	0.05
8	0.1769926664029649641	0.177045	0.03
9	0.1666666666666666667	0.166701	0.02
10	0.1443375672974064411	0.144681	0.24
11	0.1410544578570137366	0.141252	0.14
12	0.1373236156889236662	0.137633	0.23
13	0.1326643857765088351	0.133379	0.54
14	0.1275163863998600644	0.127829	0.25
15	0.1154700538379251529	0.115811	0.30
16	0.1137125784440782042	0.114574	0.76
17	0.1113943099632405880	0.112141	0.67
18	0.1091089451179961906	0.109890	0.72
19	0.1061737927289732618	0.107288	1.05
20	0.1032272183417310354	0.104049	0.80

k	ρ_{opt} using method in [96]	ρ^*_{opt} using our method	% increment
21	0.0962250448649376274	0.099165	3.06
22	0.0951772351261450917	0.095877	0.74
23	0.0937742911094478264	0.094625	0.91
24	0.0923541375945022204	0.093982	1.76
25	0.0906182448311340175	0.091688	1.18
26	0.0887829248953373781	0.090231	1.63
27	0.0868913397937031505	0.088238	1.15
28	0.0824786098842322521	0.086795	5.23
29	0.0818048133956910115	0.084545	3.35
30	0.0808828500258641436	0.082246	1.69
31	0.0798972448089536737	0.081665	2.21
32	0.0788506226168764215	0.080457	2.04
33	0.0776371221483728244	0.079604	2.53
34	0.0763874538343494465	0.078827	3.19
35	0.0751604548962267707	0.076918	2.34
36	0.0721687836487032206	0.075950	5.24

For each k, we have chosen 1000 initial instances. For each of these instances, we have computed the value of ρ executing our iterative algorithm. Finally, we report ρ^*_{opt} = minimum value of ρ over all the 1000 instances. In Table 7.1, we have compared ρ^*_{opt} with the value of ρ_{opt} obtained by the algorithm in [97] for different values of k. In the same table, we have also reported the % increment of the ρ value by our method, i.e., $\frac{(\rho^*_{opt} - \rho_{opt})}{\rho_{opt}} \times 100$. This indicates that the range ρ produced by our method is very close to that in [97].

We have also compared our method with that of [96] when the region is an equilateral triangle. The experimental results for different values of k appear in Table 7.2. Figure 7.1 demonstrates the output of our algorithm for covering a given convex polygon with 13 circles. Since there is no prior study of this problem when Π is an arbitrary convex polygon, we could not do any comparative study in this case.

Table 7.3: Performance evaluation of our algorithm

k	ρ^*_{opt}	$\rho_{average}$	SD	Time	k	ρ^*_{opt}	$\rho_{average}$	SD	Time
4	0.353553	0.395284	0.040423	0.052	18	0.160682	0.164347	0.001092	0.351
5	0.326165	0.326247	0.000201	0.073	19	0.158345	0.160797	0.000885	0.377
6	0.298730	0.309837	0.008433	0.090	20	0.152524	0.156772	0.000877	0.405
7	0.274295	0.27603	0.001668	0.107	21	0.149080	0.153131	0.001253	0.436
8	0.260317	0.26131	0.003079	0.124	22	0.143711	0.148640	0.000582	0.465
9	0.230672	0.231119	0.000540	0.143	23	0.141278	0.145498	0.001738	0.499
10	0.218239	0.218244	0.000004	0.164	24	0.138715	0.142105	0.001507	0.531
11	0.212533	0.213855	0.000894	0.184	25	0.134397	0.139549	0.001572	0.557
12	0.202395	0.205567	0.000908	0.206	26	0.132050	0.136489	0.001618	0.587
13	0.194339	0.194960	0.000645	0.228	27	0.128660	0.133725	0.001298	0.623
14	0.185527	0.189217	0.001722	0.258	28	0.127426	0.131589	0.001357	0.655
15	0.180208	0.182782	0.001883	0.279	29	0.126526	0.129241	0.000964	0.688
16	0.169611	0.174669	0.003178	0.303	30	0.123214	0.127069	0.000881	0.719
17	0.165754	0.168231	0.002336	0.327					

In order to present the performance of our heuristic, we report the minimum, average and standard deviation (SD) of the value of ρ over all the 1000 instances for different values of k, and with unit square region as Π (see Table 7.3). Thus, column 3 of Table 7.1 is identical with the column 2 of Table 7.3. We have performed the entire experiment in SUN BLADE 1000 machine with 750 MHz CPU speed, and have used LEDA [91] for computing the Voronoi diagram. The average time (in seconds) for processing each instance is also given. Similar results are observed with equilateral triangle.

7.5 Summary

A simple algorithm for placing a given number of base stations in a convex region, and assigning range to them is presented. The problem is equivalent to covering a convex region by equal radius circles such that the radius of the circles is minimized. This problem is very much important in the context of mobile communication. To our knowledge, this is the first attempt to cover an arbitrary convex region with a given number of circles of minimum radius. The earlier works on this problem have considered squares, rectangles and equilateral triangles only [97, 92, 96].

We have compared the results produced by our algorithm with that of the existing ones when the region under consideration is a square or an equilateral triangle. Experimental results indicate that the solutions produced by our algorithm are very close to those of the existing results where the region is a square [97] and an equilateral triangle [96]. It is mentioned in [96, 97] that for a reasonably large value of k (≥ 27), it needs to run several weeks to get the solution, whereas our method needs a fraction of a second. Thus this result is highly acceptable in the context of mobile communication applications.

Chapter 8

Base Station Placement Problem - A Constrained Variation

8.1 Introduction

In this chapter, we consider a constrained variation of the base station placement problem which has a very important application in establishing the mobile communication service in a hazardous area. Here, we need to place k base stations of equal range on the boundary of a convex polygonal region P such that each point inside P is covered by at least one base station. The objective is to reduce the cost of the network by minimizing the (common) range. Sometimes it is observed that some portions of the target region are unsuitable for placing the base stations, but the communication inside those regions need to be provided. As an example, we may consider a huge water body, say lake or river, where the base station can not be placed, but communication inside that region must be provided for the fishermen. In such cases, we need some specialized algorithms to tackle this problem.

We consider several versions of this problem. The general problem is named as *region-*

cover(k) problem, where k stands for the specified number of base stations to be placed. A simplified form of this problem is the *vertex-cover*(k) problem, where the objective is to communicate with only the vertices of P instead of covering the entire region inside the polygon. This problem is also useful in some specified applications, where guards are placed only at the vertices of the polygonal region P and the communication among them is provided using these base stations. We first present efficient algorithms for *vertex-cover(2)* and *region-cover(2)* problems, where the base stations are to be installed on a pair of specified edges of P. The time complexity of these algorithms are $O(n \log n)$ and $O(n^2)$ respectively. Next, we consider the case where $k \geq 3$. We first concentrate on the restricted version of the *vertex-cover(k)* and *region-cover(k)* problems, where all the k base stations are to be installed on the same edge of P. Our proposed algorithm for the restricted *vertex-cover(k)* problem produces optimum result in $O(\min(n^2, nk \log n))$ time, whereas the algorithm for the restricted *region-cover(k)* problem produces an $(1 + \epsilon)$-factor approximation result in $O((n + k)\log(n + k) + n\log(\lceil \frac{1}{\epsilon} \rceil))$ time. Finally, we propose an efficient heuristic algorithm for the general version of the *region-cover(k)* problem for $k \geq 3$. Experimental results demonstrate that our algorithm runs fast and produces near optimum solutions.

8.2 Preliminaries

Let P be a convex polygon with vertices $\{p_0, p_1, \ldots, p_{n-1}\}$ in anticlockwise order. Depending on the context, we will use p_n to denote the vertex p_0. The edge (p_i, p_{i+1}) is denoted by e_i. Without loss of generality, we assume that the edge e_0 is horizontal and the points in P are on or above the line containing e_0. We will use $MEC(Q)$ to denote the minimum enclosing circle of a sub-polygon Q of P whose center is constrained to lie on e_0. We use P_m to denote the convex polygon with vertices $\{p_1, \ldots, p_m\}$, where $2 \leq m \leq n$. Thus, we have $P_n = P$. Depending on the context, we will also use P_m to denote the set of vertices $\{p_1, \ldots, p_m\}$.

As a preprocessing step, we compute $MEC(P_m)$ for $2 \leq m \leq n$ in an incremental manner. The minimum enclosing circle of a convex polygon with its center on a specified edge of the said polygon can be computed in linear time [87, 105]. Thus the straight forward application of that algorithm returns $MEC(P_m)$ for all $m = 2, 3, \ldots, n$ in $O(n^2)$ time. We present a linear time algorithm for this problem as the preprocessing step. It uses the property of farthest point Voronoi diagram of the vertices of a convex polygon.

Let $FV(P_m)$ be the farthest point Voronoi diagram of the vertices of P_m for a particular value of m. Let $\mathcal{F}_m = \{f_1, f_2, \ldots, f_\alpha\}$ be the intersection points of the edges of $FV(P_m)$ with edge e_0 in left-to-right order. Lemma 8.1, stated below, says that α $(= |\mathcal{F}_m|) \leq m$. We will use r_i to denote the radius of the minimum enclosing circle of P_m with center at f_i. The members in $\mathcal{F}_m$ define $(\alpha + 1)$ intervals on e_0, namely $\{\ell_0, \ell_1, \ldots, \ell_\alpha\}$, where ℓ_0 is the interval on e_0 to the left of f_1, ℓ_α is the interval to the right of f_α, and for each $i = 1, 2, \ldots, \alpha - 1$, ℓ_i is bounded by f_i and f_{i+1}. Each open interval ℓ_i entirely lies in the farthest point Voronoi cell of a single vertex of P_m.

Lemma 8.1 *[105] Let $e = [u, v]$ and $f = [v, w]$ be two consecutive edges of a convex polygon Q. If the perpendicular bisectors of these two edges intersect outside Q, then there exists no point on the boundary of Q which lies in the farthest point Voronoi region of v.*

We now introduce a function ψ as follows: if the farthest vertex corresponding to the cell containing ℓ_i is p_j, then $\psi(\ell_i) = j$. Lemma 8.2, stated below, is an important property of the point set $\mathcal{F}_m$.

Lemma 8.2 $\psi(\ell_0) < \psi(\ell_1) < \ldots, < \psi(\ell_\alpha)$.

Proof: Let $\psi(\ell_i) = j$ and $\psi(\ell_{i+1}) = k$. Clearly, f_{i+1} is the intersection point of the perpendicular bisector of line segment $\overline{p_j p_k}$ with e_0. Since ℓ_{i+1} is to right side of ℓ_i along the directed edge $\overrightarrow{p_0p_1}$, and the vertices of P are in anticlockwise order, we have $k > j$ from the definition of farthest point Voronoi diagram. $\square$

8.2.1 Preprocessing

In this phase, we compute the minimum enclosing circle of P_m with center on e_0, for each $m = 2, 3, \ldots, n$. This helps in designing fast algorithm for the problems considered in this chapter.

Note that, we do not compute the farthest point Voronoi diagram $FV(P_m)$ explicitly for each m; but we compute the points $\mathcal{F}_m$ for each m in an incremental manner. We use an array M of size $n-1$ (indexed as $2, 3, \ldots, n$) to store the center and radius of $MEC(P_m)$ for $m = 2, 3, \ldots, n$. We also maintain a link-list $\mathcal{F}$ as the working storage. After the m-th iteration, $\mathcal{F}$ contains the members of $\mathcal{F}_m$ in left-to-right order. Let us start with $m = 2$; we compute $\mathcal{F}_2$, and store it in $\mathcal{F}$. By Lemma 8.1, $|\mathcal{F}_2| \leq 2$. In addition, for each ℓ_i, if the perpendicular projection of $\psi(\ell_i)$ on e_0 lies in ℓ_i, then it is the center of $MEC(P_2)$. We add this point in $\mathcal{F}$ as a member of $\mathcal{F}_2$, and store the center and radius of $MEC(P_2)$ in $M[2]$.

Next, we explain the incremental step. At the m-th iteration, the array M contains the information about $MEC(P_i)$ for $i = 2, 3, \ldots, m$. In addition, the link-list $\mathcal{F}$ stores the members in $\mathcal{F}_m$. At each $f_i \in \mathcal{F}$, we have stored $\min_{j=0}^{i} r_j$. We now consider p_{m+1}, and compute the $MEC(P_{m+1})$.

Let $|\mathcal{F}| = |\mathcal{F}_m| = \alpha$. Let $\mu = \psi(\ell_\alpha)$. We draw the perpendicular bisector of $\overline{p_\mu p_{m+1}}$. If it does not hit the edge e_0, then this step stops. But, if it hits e_0 at say $\hat{f}$, then we need to consider the following two cases:

$\hat{f}$ is to the right of f_α: Here we add $f_{\alpha+1} = \hat{f}$ in the array $\mathcal{F}$, and attach $\min_{j=0}^{m+1} r_j = \min(\min_{j=0}^{m} r_j, r_{m+1})$ with it as the radius.

$\hat{f}$ is to the left of f_α: Here we repeatedly apply the following procedure until $\hat{f}$ is observed at the right side of all the element of $\mathcal{F}$.

delete f_α from $\mathcal{F}$, and decrement the value of α by 1. Next, draw the perpendicular bisector of $\overline{p_\mu p_{m+1}}$, where $\mu = \psi(\ell_\alpha)$ for the current value of α. We again use $\hat{f}$ to denote the point of intersection of this straight line with e_0.

Finally, increment α by 1 and add $\hat{f}$ in $\mathcal{F}$.

The radius attached with $\hat{f}$ is $\min_{j=0}^{\alpha} r_j = \min(\min_{j=0}^{\alpha-1} r_j, r_\alpha)$. If the minimum is achieved for $j = j^*$, then the tuple (f_{j^*}, r_{j^*}) is stored in $M[m+1]$ as the center and radius of $MEC(P_{m+1})$. We also compute the perpendicular projection of p_{m+1} on e_0. If it is inside $FV(p_{m+1})$, then it is the center of $MEC(P_{m+1})$. We add this point in $\mathcal{F}$ and store it with the corresponding radius in $M[m+1]$. The process terminates after considering p_n.

Lemma 8.3 *The total time required for computing $MEC(P_m)$ for all $m = 2, 3, \ldots, n$ is $O(n)$.*

Proof: The time complexity is determined by that of maintaining the link-list $\mathcal{F}$. Note that, when we consider a new vertex p_{m+1}, we repeatedly perform the following steps:

- compute the bisector of the line segment joining p_{m+1} and another vertex of P_{m+1},
- test it with the last entry of $\mathcal{F}$, and then
- either insert it in $\mathcal{F}$ or delete the last entry in $\mathcal{F}$.

At each iteration only one element is inserted in $\mathcal{F}$. Thus, in total n entries are inserted in $\mathcal{F}$. An inserted element is deleted at most once. Thus, the lemma follows. $\square$

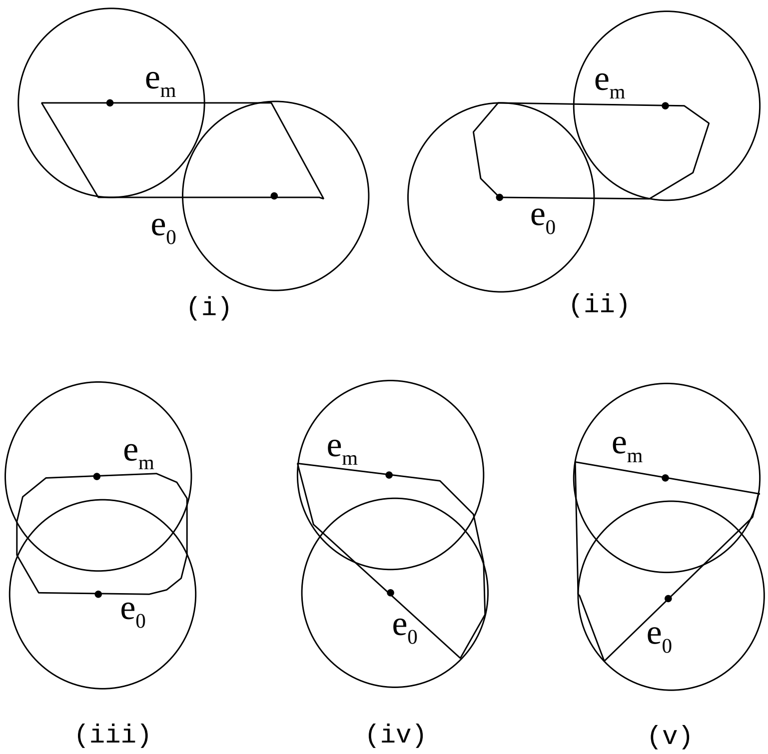


Figure 8.1: Different configurations of C^{1*} and C^{2*}

8.3 Vertex-cover(2) problem

We first consider the problem of placing two circles C^{1*} and C^{2*} of equal radii on two specified edges such that each vertex of P is covered by at least one among C^{1*} and C^{2*} and the radius becomes minimum. Without loss of generality, we assume that the center of C^{1*} is on e_0. Let the center of C^{2*} be on e_m.

Observation 8.1 *C^{1*} and C^{2*} may satisfy any one of the following five configurations (see Figure 8.1):*

(i) *C^{1*} covers $\{p_1, p_2, \ldots, p_m\}$ and C^{2*} covers $\{p_{m+1}, p_{m+2}, \ldots, p_n\}$,*

(ii) *C^{1*} covers $\{p_{m+1}, p_{m+2}, \ldots, p_n\}$ and C^{2*} covers $\{p_1, p_2, \ldots, p_m\}$,*

(iii) *there exists two vertices of P, namely p_α and p_β $(1 \leq \alpha \leq m, m+1 \leq \beta \leq n)$, such that C^{1*} covers the vertices $Q = \{p_1, p_2, \ldots, p_\alpha\} \cup \{p_n, p_{n-1}, \ldots, p_\beta\}$ and C^{2*} must cover the vertices $R = P \setminus Q$.*

(iv) *there exists a vertex of P, namely p_α $(1 \leq \alpha < m)$, such that C^{1*} covers the vertices $Q = \{p_1, p_2, \ldots, p_\alpha\}$ and C^{2*} must cover the vertices $R = P \setminus Q$.*

(v) *there exists a vertex of P, namely p_β $(m+1 < \beta \leq n)$, such that C^{2*} covers the vertices $R = \{p_1, p_2, \ldots, p_m, p_{m+1}, p_{m+2}, \ldots, p_{\beta-1}\}$ and C^{1*} must cover the vertices $Q = P \setminus R$.*

For configurations (i) and (ii), the center and radius of C^{1*} and C^{2*} can be obtained by the technique described for the preprocessing. Observe that, the center and radius of C^{1*} satisfying configuration (i) is already stored in m-th element of the array M. We now describe the method of computing the circles C^{1*} and C^{2*} for configuration (iii). This handles the configurations (iv) and (v) also. Finally, we choose the one for which maximum radius among C^{1*} and C^{2*} is minimum.

Computation of C^{1*} and C^{2*} satisfying configuration (iii)

Let us fix the vertex p_ℓ among $\{p_1, p_2, \ldots, p_m\}$ as p_α and choose every vertex p_j, $j = m+1, m+2, \ldots, n$ in order as p_β. For each p_j, we compute two circles C_j^{1*} and C_j^{2*} (of minimum radii and centered on e_0 and e_m respectively) to cover the points $Q = \{p_1, p_2, \ldots, p_\ell\} \cup \{p_j, p_{j+1}, \ldots, p_n\}$ and $R = P \setminus Q$ respectively. We use the procedure described in Section 8.2.1, which incrementally computes C_j^{1*} and C_j^{2*} for each $j = n, n-1, n-2, \ldots, m+1$. Let ρ_j^1 and ρ_j^2 denote the radii of the circles C_j^{1*} and C_j^{2*} respectively, and $\chi(j) = \max(\rho_j^1, \rho_j^2)$. We then compute $\rho(\ell) = \min_{j=m+1}^{n} \chi(j)$.

Thus, for the given ℓ, $\rho(\ell)$ is the radius of the smallest pair of circles (with centers on e_0 and e_m) such that C^{1*} covers p_ℓ and the pair of circles C^{1*} and C^{2*} cover all the vertices of P. But, if we choose some different value of ℓ, we may get a pair of circles with smaller radius (with centers on e_0 and e_m) which can also cover P. In order to compute the smallest radius, we may need to try with several values of ℓ using binary search among $\{1, 2, \ldots, m\}$.

We first choose p_ℓ to be the middle-most vertex between p_1 and p_m. In other words, $\ell = \lfloor \frac{1+m}{2} \rfloor$, and compute $\rho(\ell)$. Let $\rho(\ell) = \chi(j^*) = \max(\rho^1_{j^*}, \rho^2_{j^*})$. In the next level of recursion, the search space for p_α is split into two parts. In each part, the possible values of β are also mentioned.

- if $\alpha \in \{1, 2, \ldots, \lfloor \frac{1+m}{2} \rfloor\}$, then $\beta \in \{m+1, m+2, \ldots, j^*\}$, and
- if $\alpha \in \{\lfloor \frac{1+m}{2} \rfloor + 1, \ldots, m\}$, then $\beta \in \{j^*+1, j^*+2, \ldots, n\}$.

So, we need to perform the following two steps.

- Choose $\ell = \frac{1+\lfloor \frac{1+m}{2} \rfloor}{2}$ and compute C^{1*}_j and C^{2*}_j for each value of $j = j^*, j^*-1, \ldots, m+2, m+1$. Finally, identify the value of j (say j^*_1) for which $\max(\rho^1_{j^{1*}}, \rho^2_{j^{1*}})$ is minimum.
- Similarly, choose $\ell = \frac{1+\lfloor \frac{1+m}{2} \rfloor + m+1}{2}$, and compute C^{1*}_j and C^{2*}_j for each value of $j = n, n-1, \ldots, j^*+2, j^*+1$ in order. Finally, identify the value of j (say j^*_2) for which $\max(\rho^1_{j^{2*}}, \rho^2_{j^{2*}})$ is minimum..

Again, using the procedure described in Section 8.2.1, the time required for these two steps is $O(n)$ in total. It splits the entire search space in 4 parts for the third level of recursion. The procedure continues until a part becomes of size zero.

Complexity analysis

Theorem 8.1 *The time complexity of the above algorithm for the vertex-cover(2) problem is $O(n \log n)$.*

Proof: For configuration (i) and (ii), the optimum C^{1*} and C^{2*} can be obtained in $O(n)$ time (see Subsection 8.2.1). The time needed for handling configuration (iii) is $O(n \log n)$, since α values are chosen in at most $O(\log n)$ levels of recursion, and the execution time of each level of recursion is $O(n)$. $\square$

8.4 Region-cover(2) problem

We will use C^1 and C^2 to denote the pair of smallest radius circles for the *region-cover(2)* problem with centers on e_0 and e_m respectively. It places two circles of same radii centered on edges e_0 and e_m respectively, such that the entire region inside the closed polygon P is covered by at least one of these two circles, and the common radius ρ is minimum.

We first solve the *vertex-cover*(2) problem. It outputs two circles C^{1*} and C^{2*}, which may or may not completely cover the entire region inside P. In the former case, C^{1*} and C^{2*} intersect, and the two intersection points lie on or outside the boundary of P (see Figure 8.2(a)). In this case $C^1 = C^{1*}$ and $C^2 = C^{2*}$; or in other words, the solution of *vertex-cover(2)* problem is the same as that of the *region-cover(2)* problem. In the latter case, the circles C^{1*} and C^{2*} may or may not intersect. If they intersect, then also at least one intersection point lies inside P. In this case, we need to manipulate C^{1*} and C^{2*} by increasing their common radius and/or shifting their centers to obtain the circles C^1 and C^2. If p and q are the points of intersection of the circles C^1 and C^2, then we have the following two possible forms of the optimum solution of the *region-cover(2)* problem.

(i) p and q lie on two edges of the polygon P (Figure 8.2(b)).

(ii) One of p and q lies on an edge of P, but the other one lies outside P (Figure 8.2(c)).

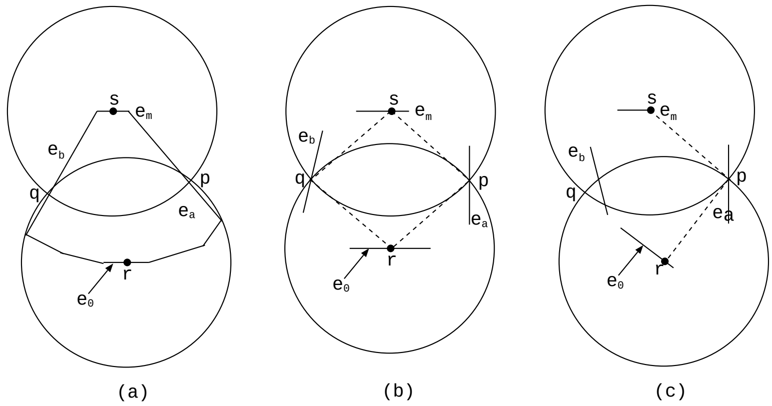


Figure 8.2: Demonstrations of the case where (a) $C^1 = C^{1*}$ and $C^2 = C^{2*}$, (b) p and q lie on the boundary of P, and (c) at least one of p and q lies outside P

We consider each edge $e_a \in \{e_1, e_2, \ldots, e_m\}$ and assume that p lies on it, and compute C^1, C^2 by trying with both the cases (i) and (ii) as mentioned above. The one producing the minimum radius is recorded. The same procedure is adopted considering each edge $e_b \in \{e_m, e_{m+1}, \ldots, e_n(= e_1)\}$ and assuming q to lie on it for computing C^1 and C^2. Finally, the pair of circles with minimum radius is reported.

As in Section 8.3, we apply recursive procedure to choose e_a. The recursion starts with choosing the middle-most edge in $\{e_1, e_2, \ldots, e_m\}$. For this choice, say e_{a^*}, we consider each edge $e_b \in \{e_m, e_{m+1}, \ldots, e_n\}$ and execute the procedures for Case (i) and Case (ii) described below to solve the *region-cover(2)* problem. Suppose the optimum solution is obtained for $b = b^*$. At the next stage of recursion, the problem splits into two parts as in the *vertex-cover(2)* problem. We choose (a) e_a as the middle-most edge in $\{e_1, e_2, \ldots, e_{a^*-1}\}$ and search for e_b in $\{e_m, e_{m+1}, \ldots, e_{b*}\}$, and (b) e_a as the middle-most edge in the subset of edges $\{e_{a^*+1}, e_{a^*+2}, \ldots, e_m\}$ and search for e_b in the subset of edges $\{e_{b^*}, e_{b^*+1}, \ldots, e_n\}$.

8.4.1 Case (i)

Given the two edges e_a and e_b, we test whether the optimum solution satisfies Case (i) or not, and if the test returns *true*, it also returns the optimum solutions C^1 and C^2.

Let the equations of the lines corresponding to the four relevant edges be $e_0 : y = 0$, $e_a : y = \mu_1 x + c_1$, $e_m : y = \mu_2 x + c_2$ and $e_b : y = \mu_3 x + c_3$. Also, let $r = (\alpha, 0)$ and $s = (\beta, \mu_1\beta + c_2)$ be the centers of C^1 and C^2 respectively. Let the points of intersection of C^1 and C^2 on e_a and e_b be $p = (\gamma, \mu_1\gamma + c_1)$ and $q = (\nu, \mu_3\nu + c_3)$ respectively. The points r, p, s and q form a rhombus of minimum edge length. As the diagonals of a rhombus intersect at their mid-point, we have

$$\alpha + \beta = \gamma + \nu \tag{8.1}$$

$$\mu_1\gamma + \mu_3\nu + c_1 + c_3 = \mu_2\beta + c_2 \tag{8.2}$$

Again, since the diagonals of a rhombus are perpendicular to each other, we have

$$\frac{\mu_2\beta + c_2}{\beta - \alpha} \times \frac{\mu_1\gamma + c_1 - \mu_3\nu - c_3}{\gamma - \nu} = -1 \tag{8.3}$$

From the equations 8.1, 8.2 and 8.3 we have the following relation (eliminating α and β).

$$(a_1\gamma + a_1'\nu + a_1'') \times (\gamma - \nu) + (a_2\gamma + a_2'\nu + a_2'') \times (a_3\gamma + a_3'\nu + a_3'') = 0 \tag{8.4}$$

Where, $a_1 = \frac{2\mu_1 - \mu_2}{\mu_2}$, $a_1' = \frac{2\mu_3 - \mu_2}{\mu_2}$, $a_1'' = \frac{2}{\mu_2}(c_1 + c_2 + c_3)$,
$a_2 = \mu_1$, $a_2' = \mu_3$, $a_2'' = c_1 + c_2$, and
$a_3 = \mu_1$, $a_3' = -\mu_3$, $a_3'' = c_1 - c_2$.

We recast the equation 8.4 as a quadratic equation of γ as follows:

$$a_4\gamma^2 + (a_5\nu + a_5')\gamma + a_6\nu^2 + a_6'\nu + a_6'' = 0 \tag{8.5}$$

Where, $a_4 = a_1 + a_2 a_3$,

$a_5 = a_1' - a_1 + a_2' a_3 + a_2 a_3'$, $a_5' = a_1'' + a_2'' a_3 + a_2 a_3''$, and

$a_6 = -a_1' + a_2' a_3''$, $a_6' = -a_1'' + a_2'' a_3' + a_2' a_3''$, $a_6'' = a_2'' a_3''$.

The solutions of the quadratic equation 8.5 are

$$\gamma_1 = b_1 \nu + b_1' + \sqrt{b_2 \nu^2 + b_2' \nu + b_2''} \tag{8.6}$$

and

$$\gamma_2 = b_1 \nu + b_1' - \sqrt{b_2 \nu^2 + b_2' \nu + b_2''} \tag{8.7}$$

Where, $b_1 = \frac{-a_5}{2a_4}$, $b_1' = \frac{a_5'}{2a_4}$, and
$b_2 = \frac{a_5^2 - 4a_4 a_6}{2a_4}$, $b_2' = \frac{2a_5 a_5' - 4a_4 a_6'}{2a_4}$, $b_2'' = \frac{(a_5')^2 - 4a_4 a_6''}{2a_4}$.

Again, we consider equations 8.1 and 8.2, and eliminating β from these two equations, we have

$$\alpha = d_1 \gamma + d_1' \nu + d_1'' \tag{8.8}$$

where, $d_1 = \frac{\mu_2 - \mu_1}{\mu_2}$, $d_1' = \frac{\mu_2 - \mu_3}{\mu_2}$ and $d_1'' = \frac{c_2 - c_1 - c_3}{\mu_2}$.

Let us now consider the squared length of the one edge of the rhombus, which is $(d(r,q))^2 = (\nu - \alpha)^2 + (\mu_3 \nu + c_3)^2 = (\nu - d_1 \gamma - d_1' \nu - d_1'')^2 + (\mu_3 \nu + c_3)^2$.

Substituting γ_1 and γ_2 for γ, we have two expressions of $(d(r,q))^2$ in terms of ν as follows:

$$\psi_1(\nu) = (g_1 \nu + g_1' + \sqrt{b_2 \nu^2 + b_2' \nu + b_2''})^2 + (m_3 \nu + c_3)^2 \tag{8.9}$$

and

$$\psi_2(\nu) = (g_1 \nu + g_1' - \sqrt{b_2 \nu^2 + b_2' \nu + b_2''})^2 + (m_3 \nu + c_3)^2 \tag{8.10}$$

Where, $g_1 = 1 - b_1 d_1 - d_1'$ and $g_1' = -b_1' d_1 - d_1''$.

(These are obtained by some algebraic simplifications using equations 8.6 and 8.7.)

Minimizing these two expressions with respect to ν, we have the following 2 equations in ν.

$\frac{\partial}{\partial \nu}\psi_1(\nu) = 0$
$\Longrightarrow \sqrt{b_2\nu^2 + b_2'\nu + b_2''} \times (h_1\nu + h_1') + h_2\nu^2 + h_2'\nu + h_2'' = 0$

and

$\frac{\partial}{\partial \nu}\psi_2(\nu) = 0$
$\Longrightarrow -\sqrt{b_2\nu^2 + b_2'\nu + b_2''} \times (h_1\nu + h_1') + h_2\nu^2 + h_2'\nu + h_2'' = 0$

where, $h_1 = 2g_1^2 + 2b_2 + 2\mu_3^2$, $h_1' = 2g_1g_1' + b_2' + 2\mu_3c_3$, and
$h_2 = 4b_2g_1$, $h_2' = 3g_1b_2' + 2g_1'b_2$, and $h_2'' = g_1'b_2'$.

Both the equations are of degree 4 in ν, and can be solved using the technique due to Ferrari [52]. These produce at most eight different values of ν. We consider each of them and test whether the corresponding points r, s, p and q lie on e_0, e_m, e_a and e_b respectively. If one of these tests returns *true*, we compute the radius ρ. We also do the same test and compute the value of ρ by choosing r at two end-points of e_0 and s at two end points of e_m. Finally, we choose the one producing the minimum value of ρ, and compute the corresponding center points r and s for the circles C^1 and C^2. If none of these tests returns *true*, then the optimum solution does not satisfy Case (i).

8.4.2 Case (ii)

In this case, we compute C^1 and C^2 such that they intersect at the points p and q such that p lies on the edge e_a, q is outside P, but both the circles contain the edge e_b or a part of it. Let $P = P_{SW} \cup P_{SE} \cup P_{NE} \cup P_{NW}$, where $P_{SE} = \{p_1, p_2, \ldots, p_a\}$, $P_{NE} = \{p_{a+1}, p_{a+2}, \ldots, p_m\}$, $P_{NW} = \{p_{m+1}, p_{m+2}, \ldots, p_b\}$, and $P_{SW} = \{p_{b+1}, p_{b+2}, \ldots, p_{n-1}, p_n\}$. Note that, the size of each of these subsets is at least 1.

Let $r^* \in e_0$ and $s^* \in e_m$ be the centers of C^{1*} and C^{2*} respectively. We need to compute

r and s, the centers of C^1 and C^2 respectively. As p is on the edge e_a, we have r and s on the line segments $\overline{r^*p_1}$ and $\overline{s^*p_m}$ respectively, and $d(p,r) = d(p,s)$. The objective is to identify p, r and s.

We compute the Voronoi partition line $V(e_0, e_m)$ of the line segments e_0 and e_m. It consists of parabolic arcs and line segments as shown in Figure 8.3. For each point $\tau \in V(e_0, e_m)$, its smallest distances from e_0 and e_m are same. Let π be the intersection point of $V(e_0, e_m)$ and the edge $e_a = [p_a, p_{a+1}]$. If no such intersection exists then choose $\pi = p_a$ or p_{a+1}. We choose $\Pi = p_a$ if $\max(dist(p_a, e_0), dist(p_a, e_m)) \leq \max(dist(p_{a+1}, e_0), dist(p_{a+1}, e_m))$; otherwise we choose $\Pi = p_{a+1}$. Here $dist(p, e) =$ minimum distance of a line segment e from a point p. Let $\hat{s}$ be the closest point of e_m from π. We compute $\hat{\rho} = d(\hat{s}, \pi)$ and mark $\hat{s}$ as s (the initial estimate of the center of C^2). Also mark $\hat{r}$ on e_0 as r (the estimate of the center of C^1) such that $d(\pi, \hat{r}) = \hat{\rho}$. Let $\hat{C}^1$ and $\hat{C}^2$ are the initial estimate of C^1 and C^2 respectively.

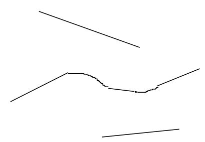


Figure 8.3: Voronoi diagram of two line segments

Lemma 8.4 *The circle $\hat{C}^2$ centered at $\hat{s}$ with radius $\hat{\rho}$ covers all the points in P_{NE}, and the circle $\hat{C}^1$ of radius $\hat{\rho}$ and centered at $\hat{r}$ covers all the points in P_{SE}.*

Proof: Follows from the fact that we have started with C^{1*} and C^{2*} (the solution of vertex-cover(2) problem) and have obtained $\hat{C}^1$ and $\hat{C}^2$ by increasing their radii and shifting their center towards right on e_0 and e_m. $\square$

If $\hat{C}^1$ and $\hat{C}^2$ completely encloses $P_{SE} \cup P_{SW}$ and $P_{NE} \cup P_{NW}$ respectively, then $\hat{C}^1$ and $\hat{C}^2$ correspond to the optimum solution C^1 and C^2 respectively. But such a situation may not arise always. There may exist situation where one/more point(s) P_{NW} (resp. P_{SW}) which lie outside $\hat{C}^2$ (resp. $\hat{C}^1$). This is due to the fact that the centers of $\hat{C}^1$ and $\hat{C}^2$ are to the right side of the centers of C^{1*} and C^{2*} on e_0 and e_m respectively.

Consider a situation where $\hat{C}^2$ does not cover all the vertices of P_{NW}. Here, $\hat{s}$ lies to the farthest point Voronoi region of a vertex $p_\theta \in P_{NW}$, and $d(p_\theta, \hat{s}) > \hat{\rho}$. The same situation may happen for $\hat{C}^1$ with respect to the point set P_{SW}. We compute the intersections of the farthest point Voronoi diagram $FV(P_{NW})$ with the edge e_m using the method described in Section 8.2.1. All these intersection points lie on the line segment $\overline{s^* p_m}$ [107]. Let us name these members as $FV^m = \{f_1^m, f_2^m, \ldots\}$ from left to right along e_m. Similarly, we compute the intersection points of the farthest point Voronoi diagram $FV(P_{SW})$ with e_0, and name these points as $FV^0 = \{f_1^0, f_2^0, \ldots\}$ from left to right. We will use these two lists (FV^0 and FV^m) for searching r and s.

Lemma 8.5 *If s (on e_m) lies inside the farthest point Voronoi region of $p_\theta \in P_{NW}$, then the radius of C^2 is at least $d(p_\theta, s)$. Similarly, If r (on e_0) lies inside the farthest point Voronoi region of $p_\phi \in P_{SW}$, then the radius of C^1 is at least $d(p_\phi, r)$.*

Proof: Follows from the property of the farthest point Voronoi diagram. $\square$

Lemma 8.6 *[107] The radius of the smallest enclosing circle of $P_{NW} \cup P_{NE}$ increases if we move the center s from s^* towards p_m along e_m. Similarly, the radius of the smallest enclosing circle of $P_{SW} \cup P_{SE}$ will increase if we move the center r from r^* towards p_1 along e_0.*

Lemma 8.7 *If at least one of the two inequalities $C^1 \neq \hat{C}^1$ and $C^2 \neq \hat{C}^2$ is true, then C^1 and C^2 pass through at least one point in P_{NW} and P_{SW} respectively.*

Proof: $\hat{C}^2$ includes all the vertices in P_{NE} (by Lemma 8.4). Let $C^2 \neq \hat{C}^2$, and $\hat{C}^2$ can not include all the vertices in P_{NW}. Let p_θ be the only vertex which is not covered by

$\hat{C}^2$. In order to include p_θ, we need to place s in the farthest point Voronoi region of p_θ. Let the interval $[f_i^m, f_{i+1}^m]$ be this region on e_m which is to the right of s^*, and s lies in the proper interior of (f_i^m, f_{i+1}^m). Thus $d(s, p_\theta)$ is the radius of C^2. Knowing s and $d(s, p_theta)$, we can compute the point π on e_a (where C^1 and C^2 intersect) and the point r on e_0 (the center of C^1). If C^1 does not pass through a point in P_{SW}, then we can reduce the radius of both C^1 and C^2 by shifting the center of C^1 to the right along e_0, and adjusting the radii of C^1 and C^2 so that they intersect on e_a. Thus, we have a contradiction. □

In order to find s, we consider each member $f_i^m \in FV^m$ in order, and compute the radius of the smallest enclosing circle of P_{NW} with center at f_i^m, and then compute the center r of C^1 on e_0. If C^1 does not covers P_{SW}, we need to consider other members of FV^m to the right of f_i^m. Finally, we can identify two points f_i^m, f_{i+1}^m such that the circles C^1 and C^2 computed with f_i^m does not cover all the points of P, but those with f_{i+1}^m covers all the points of P. This implies that s (the center of C^2) must be a point in the interval $[f_i^m, f_{i+1}^m]$. We repeat the same process to identify the vertex $p_\phi \in P_{SW}$ through which C^1 passes.

Summarizing the above discussions, we have the following information: C^1 passes through p_ϕ, the center of C^1 is $r \in e_0$, C^2 passes through p_θ, the center of C^2 is $s \in e_m$, and C^1 and C^2 intersect at $\pi \in e_a$.

We also have $d(p_\theta, s) = d(s, \pi) = d(\pi, r) = d(r, p_\phi)$. Assuming the coordinate of π as $(\beta, \mu_1\beta + c_1)$, and substituting it in the above equality, we have a fourth degree equation in β. We test the feasibility of its each solution with respect to $\pi \in e_a$ and the $s \in [f_i, f_{i+1}]$, and choose the one producing the minimum radius. Thus, we have the following lemma.

Lemma 8.8 *If Case (ii) of* region-cover(2) *problem appears, then the optimum solution can be obtained in $O(n)$ time.*

Proof: After computing π, we compute s by considering each pair of consecutive members in FV^m, which may be $O(n)$ in number. After identifying a pair (f_i, f_{i+1}) in which s lies, we can identify the vertex $p_\theta \in P_{NW}$ through which C^2 passes. Next we can apply the same process to get the center of C^1 and the point p_ϕ through which C^1 passes in $O(n)$ time [52]. Solving the 4th degree equation needs $O(1)$ time. This generates at most 4 real solutions. So, the feasibility test also needs $O(1)$ time. Thus, the lemma follows. $\square$

Theorem 8.2 *The* region-cover(2) *problem can be solved in $O(n^2)$ time.*

Proof: The recursive procedure is similar to the *vertex-cover(2)* problem. In each level of recursion (i.e. for each choice of e_a), we need to consider $O(n)$ members of $P_{NW} \cup P_{SW}$. For each of these members, the test of Case (i) needs $O(1)$ time but the test of Case (ii) may need $O(n)$ time (see Lemma 8.8). Thus, the recursion relation expressing the time complexity is

$$T(n) = 2T(\frac{n}{2}) + O(n^2).$$

$\square$

8.5 Restricted vertex-cover(k) problem

In this section, we present the algorithm for placing k equal circles centered at k points on an edge of the polygon P such that each of the vertices of P is covered by at least one circle and the radius becomes minimum. Without loss of generality, we assume that the base stations are to be placed on e_0. To solve this problem, we first solve the following decision problem which helps us to solve the original problem.

RVCD($\mathbf{k}, \rho$): Given a convex polygon P and a real number ρ test whether it is possible to cover all vertices of P by k circles of radius ρ centered at k points on e_0.

We present an $O(n)$ time algorithm for solving $RVCD(k, \rho)$. As an obvious check, we compute the distance of each vertex of P from e_0. If any one of these distances is greater than ρ, then the test return *false*. Otherwise, we put k circles of radius ρ with centers at k points on e_0 one by one in right-to-left order as follows. We use C^i and s_i to denote the i-th circle in the order, and its center.

In order to find C^1, we choose a point u on e_0 such that $d(p_1, u) = \rho$. We draw the circle of radius ρ with center at u and find the maximum index m such that each of the vertices in $\{p_1, p_2, \ldots, p_{m-1}\}$ are inside the circle, but p_m is outside of that circle (see Figure 8.4(a) where $m = 2$). Consider the vertical line ℓ drawn at u. If p_m is in the right side of ℓ, then move the center of circle from u to a point u' on the line segment $[u, p_1]$ such that $d(u', p_m) = \rho$ (see Figure 8.4(b)). Note that, this right shifted circle covers all the vertices $p_1, p_2, \ldots, p_m$. We rename u' as u and repeat the same process for shifting the circle towards right to include the next vertex of P. As soon as the next vertex (say p_{m^*}) is to the left of the corresponding vertical line ℓ^*, the circle C^1 is finalized with center $s_1 = u'$.

Let $p_1, p_2, \ldots, p_{m^*}$ be the vertices of P which are inside C^1. For the second circle C^2, we choose its center s_2 such that $s_2 \in e_0$ and $d(p_{m^*+1}, s_2) = \rho$. In order to enclose maximum number of vertices of P inside C^2, we repeat the same process as described for C^1. The same process is repeated for placing the other circles. Finally, if p_0 is covered by a circle C^i for some $i \leq k$, then the algorithm returns *true*, otherwise it returns *false*. The time complexity of this procedure is clearly $O(n)$.

Now, we provide an algorithm for the optimization version of the restricted vertex cover problem as follows:

RVCO(k): Compute the minimum value of ρ such that the circles of radius ρ with center on k points of the edge e_0 cover all the vertices of P.

Let C_{ij} denote the minimum radius circle with center on e_0 which covers the vertices

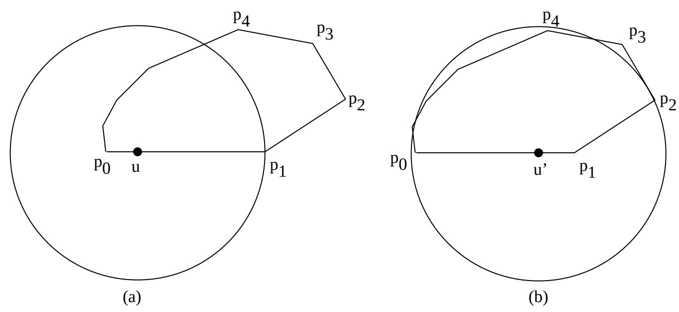


Figure 8.4: (a) Initial placement of circle C^1, and (b) modified placement of C^1 to cover the next vertex

$\{p_i, p_{i+1}, \ldots, p_j\}$ of P. Let the radius of C_{ij} be ρ_{ij}. We first identify two consecutive vertices p_α and $p_{\alpha+1}$ such that k circles of radius $\rho_{1\alpha}$ unable cover all the vertices of P, but k circles of radius $\rho_{1(\alpha+1)}$ can do it. We can identify p_α by applying binary search in the array M (created in preprocessing step). At each step, we get the radius from the corresponding entry of the array M, and execute the decision procedure $RVCD$. If the optimum value of the radius is ρ^*, then $\rho_{1\alpha} < \rho^* \leq \rho_{1(\alpha+1)}$. We choose $\rho^* = \rho_{1(\alpha+1)}$ as initial estimate of ρ^*. In next step, we try to improve the estimate of ρ^*. Note that, the first circle with radius $\rho'(< \rho^*)$ is unable to contain $p_{\alpha+1}$ by the definition of $\rho_{1(\alpha+1)}$. So, we repeat the same procedure to cover $\{p_{\alpha+1}, p_{\alpha+2}, \ldots, p_{n-1}, p_n\}$ with $k-1$ circles centered on e_0. This returns a pair of radii $\rho_{(\alpha+1)j}$ and $\rho_{(\alpha+1)(j+1)}$, and we have $\max(\rho_{1\alpha}, \rho_{(\alpha+1)j}) < \rho^* \leq \min(\rho_{1(\alpha+1)}, \rho_{(\alpha+1)(j+1)})$. As a revised estimate we take $\rho^* = \min(\rho_{1(\alpha+1)}, \rho_{(\alpha+1)(j+1)})$. We may have to repeat the same experiment at most k times in total to determine the value of ρ^*.

8.5.1 Correctness and complexity

The correctness proof of the algorithm is as follows. It is clear that $\rho^* \leq \rho_{1(\alpha+1)}$ (the initial estimate of ρ^*). In second step, if $\rho_{(\alpha+1)(j+1)} \leq \rho_{1(\alpha+1)}$ then $\rho_{(\alpha+1)(j+1)}$ will be chosen as the revised estimate of ρ^*; otherwise the initial estimate of ρ^* is retained. The same process is repeated k times. At each step, we have computed the optimum radius of the first circle. It is selected for an estimate of ρ^* or not depending on whether its value is less than the existing estimate of ρ^* or not. Thus, at each step, the estimate of ρ^* decreases maintaining its feasibility to cover all the vertices of P.

Lemma 8.9 *The time complexity of the proposed algorithm for the $RVCO(k)$ problem is $O(nk \log n)$.*

Proof: As mentioned earlier, we may need to run *RVCO* procedure with parameters $k, k-1, \ldots, 1$. The $O(n \log n)$ time complexity for each call of *RVCO* procedure follows from the fact that we have applied binary search, and at each step of the binary search, the *RVCD* procedure needs $O(n)$ time. $\square$

If $k = O(n)$, then the time complexity of the above algorithm is $O(n^2 \log n)$. In this case, instead of using the binary search, we may use a sequential search for solving $RVCO(k)$ problem as follows: Run *RVCD* procedure with $M[2], M[3], \ldots$ until the procedure returns *true* at a vertex p_α, say. Again apply linear search to solve $RVCO(k-1)$ with the vertex set $\{p_{\alpha+1}, p_{\alpha+2}, \ldots, p_{n-1}, p_n\}$. It is easy to see that this procedure needs $O(n^2)$ time. Thus, we have the following theorem.

Theorem 8.3 *The time complexity of the $RVCO(k)$ problem is $O(\min\{nk\log n, n^2\})$.*

8.6 Restricted region-cover(k) problem

We now concentrate on covering the entire region inside the convex polygon P. Here we propose an $(1+\epsilon)$-approximation algorithm for the restricted region-cover(k) problem. As in the earlier problems, here also we assume that the k centers of the covering circles appear on $e_0 = [p_0, p_1]$.

Lemma 8.10 *Let ℓ_0 and ℓ_1 be two lines perpendicular to the edge $e_0 = [p_0, p_1]$ at p_0 and p_1 respectively. If ℓ_0 (resp. ℓ_1) cuts the boundary of P at q_0 (resp. q_1), then in the optimal solution, the portion to the left (resp. right) of ℓ_0 (resp. ℓ_1) will be covered by only one circle.*

Proof: Consider the optimum solution, where $s_1, s_2, \ldots, s_k$ are the centers of the circles on the edge e_0 from left to right. We will show that if $q_0 \in e_i$ then the sub-polygon Q of P with vertices $\{p_n, p_{n-1}, \ldots, p_{i+1}, q_0\}$ will be covered by the circle centered at s_1 (see Figure 8.5(a)).

Let p be a point inside or on the boundary of Q. Thus, $d(p, s_1) < d(p, s_i)$, for all $i = 2, 3, \ldots, k$ (see Figure 8.5(b)). Thus, if p is covered by any one of the circles centered at $s_2, s_3, \ldots, s_k$, then p is also covered by the circle centered at s_1 as the radius of all the k circles are same. The same argument holds for the portion of P to the right of ℓ_1. $\square$

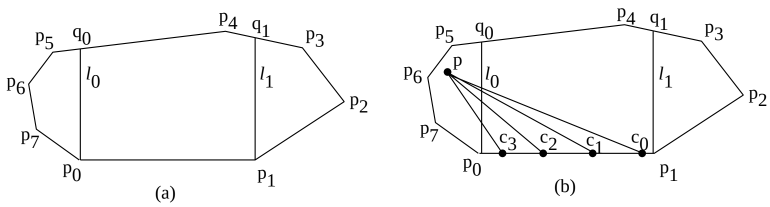


Figure 8.5: Proof of Lemma 8.10

Lemma 8.11 *The set of k equal circles with centers on e_0 and of minimum radius that can cover the perimeter of P, is the solution of the restricted-region-cover(k) problem.*

Proof: The proof follows from the fact that in order to cover the entire perimeter of P, each two consecutive circles C^i and C^{i+1} will intersect on or outside P. $\square$

As in the earlier section, here also we can design the *restricted perimeter cover* decision and optimization procedures as follows:

RPCD($\mathbf{k}, \rho$): Test whether k circles of radius ρ and centered at k points on e_0 can cover the entire perimeter of P.

RPCO($\mathbf{k}$): Find the minimum radius ρ and the centers on e_0 of k circles such that these k circles can cover the entire region inside the polygon P.

The procedure $RPCD(k, \rho)$ is similar to the procedure $RVCD(k, \rho)$ and can be executed in $O(n + k)$ time.

We present an $(1+\epsilon)$-approximation algorithm for the optimization problem $RPCO(k)$. Thus, if the optimum value of the radius of k equal circles centered at e_0 for covering the polygon P is ρ^*, then our procedure outputs a radius of value at most $(1+\epsilon)\rho^*$, and the corresponding centers (on e_0) of those k circles. The basic idea is to identify two points q' and q'' on an edge, say e_α, of the polygon P with $d(q', q'') \leq \epsilon\rho^*$, such that if ρ' and ρ'' are the radii of the minimum enclosing circles of the convex polygons $\{p_1, p_2, \ldots, p_\alpha, q'\}$ and $\{p_1, p_2, \ldots, p_\alpha, q''\}$ respectively, both centered on e_0, then $RPCD(k, \rho') = \mathit{false}$, but $RPCD(k, \rho'') = \mathit{true}$. This implies, $\rho' < \rho^* \leq \rho''$, and $\rho' + d(q', q'') \geq \rho''$. Thus, we have $\rho'' \leq (1 + \epsilon)\rho^*$. Our objective is to identify such a pair of points (q', q'').

We first apply binary search on the vertices of the polygon P to identify the edge e_α of P containing the pair of points (q', q''). Next, we split the edge into k equal sized pieces by introducing $(k-1)$ points, namely $\{q_1, q_2, \ldots, q_{k-1}\}$. Again, we apply binary search to identify a pair of consecutive points q_i and q_{i+1} such that $q', q'' \in [q_i, q_{i+1}]$. Next, we identify q' and q'' by searching in the interval $[q_i, q_{i+1}]$ using bisection method. The splitting of e_α into k equal parts will help in analyzing the approximation factor of the algorithm. Below we present the stepwise description of the algorithm.

Step 1: (* Compute the edge e_α *)

Set $i = 1; j = n$; (* Initialization Step *)

Step 1.1: Set $m = \frac{i+j}{2}$ (* p_m be the middle-most vertex in $\{p_i, p_{i+1}, \ldots, p_j\}$ *)

Step 1.2: Compute the radius ρ of the minimum enclosing circle of the polygon $P_m = \{p_1, p_2, \ldots, p_m\}$ centered on e_0.

Step 1.3: If $RPCD(k, \rho) = true$, then set $j = m$, otherwise set $i = m$.

Step 1.4: If $i < j - 1$ then Go To Step 1.1, otherwise set $\alpha = i$.

Step 2: (* Identify q_i and q_{i+1} *)

Divide the edge $e_\alpha = [p_\alpha, p_{\alpha+1}]$ in k equal parts by introducing $k-1$ points $\{q_1, q_2, \ldots, q_{k-1}\}$ in order along $\overrightarrow{p_\alpha p_{\alpha+1}}$.

Step 2.1: Apply the technique of Step 1 to identify two consecutive points q_i and q_{i+1} such that if ρ' and ρ'' are the radii of the minimum enclosing circles of the convex polygons $\{p_1, p_2, \ldots, p_\alpha, q_i\}$ and $\{p_1, p_2, \ldots, p_\alpha, q_{i+1}\}$ respectively, both centered on e_0, then $RPCD(k, \rho') = false$, but $RPCD(k, \rho'') = true$.

Step 3: (* Identify q' and q'' *)

Let $d_\Pi(p_1, q) =$ the distance of the point q on the boundary of the polygon P from p_1 in anticlockwise direction along the boundary of P.

Set $\mu = 0$, $\nu = d_\Pi(p_1, q_{i+1})$ (* Initialization step *)

Step 3.1: Executes the Step 3.2 $\log\lceil \frac{2}{\epsilon} \rceil$ times.

Step 3.2: Set $\gamma = \frac{\mu+\nu}{2}$. Let $q \in e_\alpha$ be a point such that $d_\Pi(p_1, q) = \gamma$. Compute the radius ρ of the minimum enclosing circle of the convex polygon $\{p_1, p_2, \ldots, p_\alpha, q\}$ centered on e_0. If $RPCD(k, \rho) = true$ then set $\nu = \gamma$, otherwise set $\mu = \gamma$.

Step 3.3 Let $q \in e_\alpha$ be a point such that $d_\Pi(p_1, \nu) = \gamma$. Compute the radius ρ of the minimum enclosing circle of the convex polygon $\{p_1, p_2, \ldots, p_\alpha, \nu\}$ centered on e_0. (* ρ be the output of our algorithm *)

8.6.1 Correctness and complexity

Theorem 8.4 *If ρ^* and ρ are the radii of optimum solution and the solution obtained by executing the above algorithm respectively, then $\rho \leq (1+\epsilon)\rho^*$.*

Proof: In Step 2, we obtain two points q_i and q_{i+1} on e_α such that if ρ' and ρ'' are the radii of the minimum enclosing circle of the polygons $\{p_1, p_2, \ldots, p_\alpha, q_i\}$ and $\{p_1, p_2, \ldots, p_\alpha, q_{i+1}\}$ respectively, centered on e_0, then $RPCD(k, \rho') = \mathit{false}$, but $RPCD(k, \rho'')$ $= \mathit{true}$. Thus, $\rho' < \rho^* \leq \rho''$ and there exists a point $q \in e_\alpha$ such that ρ^* is the radius of minimum enclosing circle of the convex polygon $\{p_1, p_2, \ldots, p_\alpha, q\}$ centered on e_0. We now consider the following two cases:

Case 1: $q \in [q_i, q_{i+1}]$ for some $i \geq 1$, and

Case 2: $q \in [p_\alpha, q_1]$.

In Case 1, $\overline{p_\alpha q_1}$ lies completely inside the right-most circle of radius ρ^* centered on e_0. This implies, $d(p_\alpha q_1) \leq 2\rho^*$. In Case 2, the line segments $\overline{q_{i+1}q_{i+2}}$, $\overline{q_{i+2}q_{i+3}}$, ..., $\overline{q_{k-2}q_{k-1}}$, $\overline{q_{k-1}p_{\alpha+1}}$ lie completely inside ℓ circles of radius ρ^*, where $\ell \leq k-1$. Since the length of all the line segments $\overline{q_j q_{j+1}}$, $j = i+1, i+2, \ldots, k-1$ are same (by construction in Step 2), we have $(k-1) \times d(p_\alpha, q_1) \leq 2\rho^* \times \ell$. This, in turn, implies $d(p_\alpha, q_1) \leq 2\rho^*$. Thus, in both the cases, $d(q_i, q_{i+1}) \leq 2\rho^*$.

At each iteration of Step 3.2, the value of $(\nu - \mu)$ reduces to half of its previous value. Initially, we have $(\nu - \mu) \leq 2\rho^*$. As we have executed Step 3.2 $\log\lceil\frac{2}{\epsilon}\rceil$ times, $(\nu - \mu)$ reduces to a value less than or equal to $\frac{2\times\rho^*}{2^{\log\lceil\frac{2}{\epsilon}\rceil}} = \epsilon\rho^*$.

Let the points $q', q'' \in e_\alpha$ be the output of Step 3.2 such that $d_\Pi(p_1, q') = \mu$ and $d_\Pi(p_1, q'') = \nu$. This implies, $d(q', q'') \leq \epsilon\rho^*$. Let ρ' and ρ'' be the radii of the minimum enclosing circle of the polygons $\{p_1, p_2, \ldots, p_\alpha, q'\}$ and $\{p_1, p_2, \ldots, p_\alpha, q''\}$ respectively, centered on e_0. This implies, $RPCD(k, \rho') = \mathit{false}$, but $RPCD(k, \rho'') = \mathit{true}$. Thus,

$\rho' < \rho^* \leq \rho''$. Since $d(q', q'') \leq \epsilon\rho^*$, we have $\rho'' \leq (1+\epsilon)\rho^*$. Thus, the theorem follows. $\square$

Theorem 8.5 *The time complexity of our algorithm for the* restricted region-cover(k) *problem is* $O((n+k)\log(n+k) + n\log(\lceil\frac{1}{\epsilon}\rceil))$.

Proof: The decision procedure $RPCD(k, \rho)$ runs in $O(n)$ time. The Step 1 of $RPCO(k)$ needs $O(n \log n)$ time, since we call $RPCD(k, \rho)$ at most $O(\log n)$ times in this step. Similarly, the worst case time complexity of Step 2 is $O((n+k)\log(n+k))$. In Step 3, number of calls of the routine $RPCD(k, \rho)$ is $O(\log\lceil\frac{1}{\epsilon}\rceil)$. Thus, the theorem follows. $\square$

8.7 Heuristic algorithm for region-cover(k) problem

We now present a heuristic algorithm for placing k circles of equal radii and centered on the boundary of P such that every point inside the closed polygon P is covered by at least one circle, and the common radius of these circles is minimum. Thus, the restriction of lying the centers of all the covering circles on a single edge of P is relaxed. Our approach is an iterative one. We use ρ_j as the estimate of ρ^* obtained in the j-th iterative step. We show that, $\rho_j \leq \rho_{j-1}$ for every iteration j. The execution starts with k randomly chosen points on the boundary of P, and this iterative process terminates when the difference of the values of ρ produced in two consecutive iterations is less than ϵ, a preassigned small positive real number. We now present the steps to be executed in each iteration.

Input: (* of the j-th iteration *) k points on the boundary of P which are the output of the $(j-1)$-th iteration.

Output: (* of the j-th iteration *) the radius ρ_j, and the centers of k circles on the boundary of P for the next iteration.

Step 1: Let $Q = \{q_1, q_2, \ldots, q_k\}$ be the k points obtained from the $(j-1)$-th iteration; each q_i is on the boundary of P.

Step 2: Compute the Voronoi diagram of Q inside P. Let $V(q_i)$ be the Voronoi cell of q_i, for $i = 1, 2, \ldots, k$.

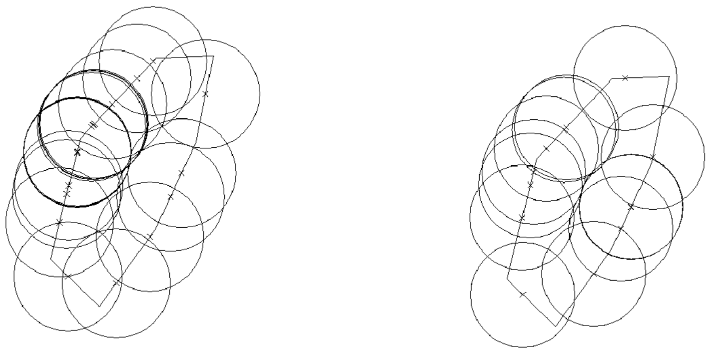
Step 3: Compute the minimum enclosing circle C^i of $V(q_i)$ with center on the boundary of P. The corresponding centers will be the input of the next iteration.

Step 4: If r_i is the radius of the circle C^i, $i = 1, 2, \ldots, k$, then $\rho_j = \max_{i=1}^{k} r_i$.

Step 5: During the iteration, it may happens that there exists a vertex in $V(q_i)$ which is far from every other vertices of $V(q_i)$, and its covering circle encloses some other $V(q_\ell)$. Thus, some times it may happen that the minimum radius required to cover P with k circles is same as the minimum radius required to cover P with k' $(\leq k)$ circles. See Figure 8.6 for the demonstration. Such a situation may be faced if the distance of the centers of two adjacent circles C^i and C^ℓ becomes very close; in other words, $d(q_i, q_\ell) < \epsilon$, where ϵ is a small constant, mentioned a priori. In such case, we drop one center, say q_ℓ, and continue iteration with $k = k - 1$.

Lemma 8.12 *If ρ_j and ρ_{j+1} are the radii obtained in the j-th and $(j+1)$-th iterations, then $\rho_j \geq \rho_{j+1}$.*

Proof: Consider $V(q_i)$, the Voronoi cell of q_i in the j-th iteration. Let $\mathcal{C}_i$ $(i = 1, 2, \ldots, k)$ be the circles with radius ρ_j and centered at q_i', where q_i' is the center of the minimum enclosing circle corresponding to $V(q_i)$. Also assume that, $\eta_{i\ell}$ be the common chord of $\mathcal{C}_i$ and $\mathcal{C}_\ell$, $\ell = 1, 2, \ldots k$, $\ell \neq i$. $\eta_{i\ell}$ may be ϕ if $\mathcal{C}_i$ does not intersect with $\mathcal{C}_\ell$ for some ℓ. Consider the region obtained by the intersection of at most $(k-1)$ half planes (bounded by the lines passing through $\eta_{i\ell}$, $\ell \neq i$ and $\ell = 1, 2, \ldots, k$) containing q_i'. This defines the Voronoi cell $V(q_i')$ at the $(j+1)$-th iteration. Note that, $\mathcal{C}_i$ encloses $V(q_i')$ completely. We may even reduce the size of the minimum enclosing circle of $V(q_i')$ by



(a) (b)

Figure 8.6: Example of the increase in the number of circles can not reduce the radius

shifting its center. This observation is true for all q_i, $i = 1, 2, \ldots, k$. Thus the lemma follows. □

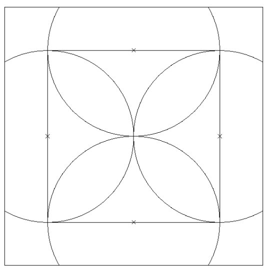
The running time of the proposed algorithm depends on the number of iterations and the worst case time complexity of an iteration. The time complexity of an iteration follows from the following theorem:

Theorem 8.6 *The worst case time complexity of an iteration of the proposed algorithm is* $O(n + k \log k)$.

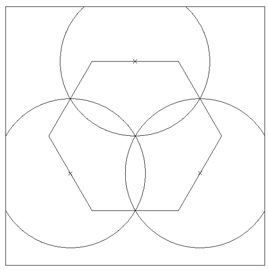
Proof: The time needed for computing the Voronoi polygons $V(q_i)$ for all the points $q_i \in Q$ inside the polygon P is $O(n+k \log k)$, where n is the number of vertices in P and $O(k \log k)$ is the time complexity for computing the Voronoi diagram of the k points in Q [19]. The computation of the minimum enclosing circles $\{C^i, i = 1, 2, \ldots, k\}$ needs another $O(n + k)$ time since (i) the total number of edges of all the Voronoi cells is $O(n + k)$, and (ii) the time complexity of computing the minimum enclosing circle of a convex polygon is linear in its number of vertices [87]. □

8.7.1 Experimental results

An exhaustive experiment is performed with several randomly generated convex polygons with different values of k. The polygon is generated in an unit square, the set Q of k centers are chosen randomly on the boundary of P as an initial solution, and the iteration starts. The iteration ends when a local minima is reached. It is observed that the quality of the result depends on the initial positions of the points in Q. It is some times observed that the range required for $k'(\leq k)$ circles is same as that for k circles. Thus, during the execution if it is observed that the Euclidean distance between the centers of two circles is less than a given small positive quantity ϵ, then we drop one of these two centers and reduce k by 1. Finally, k' is the value of k when the iteration stops.



(a)



(b)

Figure 8.7: Experimental results for (a) a square region with 4 base stations, and (b) a regular hexagon with 3 base stations

The entire experiment is performed in SUN BLADE 1000 machine with 750 MHz CPU speed, and using LEDA software [91]. Table 8.1 shows the time requirement of our algorithm for different values of n and k. For $\epsilon = 0.001$, the value of k' is also shown. Since there exists no existing result on this problem in the literature, we could not do the comparative study on the performance of our algorithm. In order to justify the quality of the solution produced by our algorithm, we have performed experiments on

Table 8.1: Experimental results

n	k	k'	time (in seconds)	n	k	k'	time (in seconds)
7	20	12	1.49	9	20	12	1.51
7	25	13	1.80	9	25	12	1.63
7	30	13	1.80	9	30	12	1.76
7	35	12	1.92	9	35	12	1.98
7	40	12	2.02	9	40	11	2.11
7	50	12	2.10	9	50	12	2.09
8	20	10	1.52	10	20	12	1.70
8	25	10	1.82	10	25	12	2.00
8	30	11	1.80	10	30	12	2.20
8	35	12	2.00	10	35	12	2.25
8	40	12	2.14	10	40	13	2.25
8	50	12	2.20	10	50	13	3.00

two specific instances for which the optimum solution is known. These are (i) P = a unit square and $k = 4$, and (ii) P = a regular hexagon with side length 1 unit and $k = 3$. For each of these two experiments, 10 different instance of initial placements are generated; for each initial placement our algorithm is executed, and the minimum value of ρ is recorded. The result obtained for experiment (i) and (ii) are $\rho = 0.5$ $(= \frac{1}{2})$ and $\rho = 1.732051(= \frac{\sqrt{3}}{2})$ respectively (see Figure 8.7). Thus, in both the cases, the optimum value is obtained. The time taken for each experiment is a fraction of a second.

8.8 Summary

Several variations of restricted region cover problem are studied in the context of the range assignment problems in mobile communication. To be specific, here the objective is to place k base stations of equal range on the boundary of a convex polygonal region P such that each point inside P is covered by at least one base station. It is shown that, both the *vertex-cover*(2) and *region-cover*(2) problems can be solved in $O(n \log n)$ and $O(n^2)$ time respectively, where n is the number of vertices of the polygon. For general k (≥ 3), if the positions of the k base stations appear on a single edge of the polygon, then *vertex-cover(k)* can be solved in $O(\min(n^2, nk\log n))$ time, whereas the algorithm for the restricted *region-cover(k)* problem produces an $(1 + \epsilon)$-approximation result in $O((n + k)\log(n + k) + n\log(\lceil \frac{1}{\epsilon} \rceil))$ time. Finally, an efficient heuristic algorithm for the general version of the *region-cover(k)* problem is proposed. Experimental results demonstrate that our algorithm runs fast and produces near optimum solutions. The proof of hardness result, and an efficient approximation algorithm for the general *region-cover*(k) problem is an useful extension of research on this problem.

Chapter 9

Conclusion

In this thesis, we considered the algorithmic issues related to the optimal placement and range assignment of radio stations in the context of a mobile radio network. Two different variations are studied - (i) the radio stations are pre-placed, and (ii) the radio stations are to be placed. In both cases, the number of radio stations are given a priori, and the objective is to minimize the total power requirement of the entire radio network. Needless to mention that the power requirement mainly depends on the ranges assigned to the radio stations. The exact function indicating the total power requirement also involves the environmental factors where the network is to be installed.

We first studied the case where a set of pre-placed radio stations is given. The problem is to assign ranges to these radio stations so that the network satisfies some desired connectivity requirement, for example, broadcast from a designated node, all-to-all communication, etc., and the total power consumption of the entire radio network is minimized. We have considered both the 1D and 2D variations of the problems.

We have also studied the some variations of the range assignment problem where the radio stations are not pre-placed, and the objective is to compute the optimal locations of the radio stations in the network such that the desired communication can be established with minimum power. Here, we have only considered the homogeneous case, where the range of every radio station is same.

The specific problems considered in this thesis are

Broadcast range assignment in a linear radio network: Here a set S of n radio stations are located on a straight line, and a designated member $s^* \in S$) is given as a source node. An integer h ($1 \leq h \leq n-1$) is also given. The objective is to assign ranges to the members in S so that s^* can send message to all other members in S using at most h hops, and the total power consumption is minimum. Two variations of this problem are considered - (i) unweighted, and (ii) weighted. For the unweighted version, we propose an $O(n^2)$ time algorithm for this problem. An $O(hn^2)$ time algorithm for this problem was presented in [31]. Thus our algorithm is an improvement over the existing result by a factor of h. It seems that, one may improve the time complexity to $O(nh \times polylog(h))$ by further investigating the geometric properties of the problem. For the weighted version, the time complexity of our proposed algorithm is $O(hn^2 \log n)$. It is also shown that in the unbounded case (i.e. $h = n-1$), the algorithm runs in $O(n^2)$ time.

Broadcast range assignment in 2D: Here the set of the pre-placed radio stations are in 2D. We consider the homogeneous case only, and have considered the following two variations: (i) find the range value r such that 2-hop homogeneous broadcast from s^* is possible with minimum cost, and (ii) given a real number r, check whether *homogeneous 2-hop broadcast* from s^* to all members in S is possible with range r, and if so, identify the smallest subset of S, to which the range r is to be assigned to accomplish the broadcast. The first problem is solved in $O(n^{2.376} \log n)$ time and $O(n^2)$ space. For the second problem, we present a 2-approximation algorithm, that runs in $O(n^2)$ time. In this context, we need to mention that the hardness result of the second problem is still undecided.

Range assignment for all-to-all communication: Here also a set S of radio stations are given, and the objective is to assign ranges to the members in S such that each of them can communicate with the others in h hops, where h is specified in advance, and the total power consumption over the entire network is minimized. We have considered both 1D and 2D versions of the problem.

In 1D version, we have considered only the unbounded case, and proposed an algorithm of $O(n^3)$ time and $O(n^2)$ space for producing the optimum range assignment. This is an improvement in the running time by a factor of n over the best known existing algorithm for the same problem [76].

For the 2D version, we have considered several variations of the problem. In the homogeneous case, we have proposed an $O(n^3(\frac{\log \log n}{\log n})^{\frac{5}{4}} \log n)$ time algorithm for producing the optimum solution. In particular, if we consider the unbounded case and homogeneous range assignment, then the proposed algorithm runs in $O(n^2)$ time. For the general h-hop all-to-all communication problem, computing the optimum solution is proved to be APX-hard, and so it does not admit a PTAS unless P = NP [76]. Since the problem is very useful in terms of its manifold applications, we have proposed a very efficient heuristic algorithm for this problem. Experimental results indicate that our heuristic algorithm runs fast and produces near-optimal solutions on randomly generated instances.

Base station placement problem: Here the radio stations are not pre-placed; the objective is to place the base stations and assign ranges to them such that a mobile terminal at each point of the region under consideration can communicate with at least one base station (i.e., each point inside the region is within the range of at least one base station) and the total power consumption of all the base stations is minimum. We have made two assumptions for solving this problem - (i) the region under consideration is convex, and (ii) the base stations are homogeneous in the sense that the range assigned to each base station is same. Two variations of this problem are considered. In the unconstrained version, the base stations can be placed anywhere inside the region. But in the constrained version, the base stations can be placed only on the boundary of the region.

The unconstrained version of the problem can be mapped to the traditional circle covering problem in computational geometry. Given a convex polygonal region, the objective is to cover the region by k equal radius circles of minimum radius. The parameter k is specified a priori. Existing studies on this problem considered the region to be a square or a triangle. The proposed algorithms use numerical techniques and are very slow in general. Our proposed algorithm works for arbitrary convex polygon. It is iterative in nature, and is based on the concept of Voronoi diagram. The execution time of our algorithm is a fraction of a second in a SUN Blade 1000 computing platform with 750 MHz CPU speed, whereas the existing methods may even take about two weeks' time for a reasonable value of the number of circles (≥ 27), as reported in [97].

Next, we considered the constrained version of the base station placement problem, where the k base stations can be placed on the boundary of the given convex region, and k is specified a priori. We considered two variations of this problem, namely *vertex-cover*, and *region-cover*. In the vertex-cover problem, the objective is to cover the vertices of the given convex polygonal region, and in the region-cover problem, the objective is to cover the entire region inside the given polygonal region. We have shown that for $k = 2$, the vertex-cover and the region-cover problems can solved in $O(n \log n)$ and $O(n^2)$ time respectively. For $k \geq 3$, we have considered a restricted case of these problems where all the base stations can be installed on a specified edge of the given polygon. Our proposed algorithm for the restricted vertex-cover(k) problem produces the optimum result in $O(min(n^2, nk \log n))$ time, whereas the algorithm for the restricted region-cover(k) problem produces an $(1+\epsilon)$-factor approximation result in $O((n+k)\log(n+k)+n\log\lceil\frac{1}{\epsilon}\rceil)$ time. Finally, we propose an efficient heuristic algorithm for the generalized version of the region-cover(k) problem, when $k \geq 3$. Experimental results demonstrate that our proposed algorithm runs fast and produces near-optimum solutions. Our next concern is to get an polynomial time approximation scheme for the generalized version of the region-cover(k) problem.

Bibliography

[1] C. Ambuhl, *An optimal bound for the MST algorithm to compute energy efficient broadcast tree in wireless networks*, Proc. of the 32th International Colloquium on Automata, Languages and Programming (ICALP'05), LNCS-3580, pp. 1139-1150, 2005.

[2] H. Alt, E. M. Arkin, H. Bronnimann, J. Erickson, S. P. Fekete, C. Knauer, J. Lenchner, J. S. B. Mitchell, and K. Whittlesey, *Minimum-cost coverage of point sets by disks*, Proc. of the 22nd ACM Symposium on Computational Geometry (SoCG'06), pp. 449-458, 2006.

[3] A. P. Azad and A. Chockalingam, *Mobile base stations placement and energy aware routing in wireless sensor networks*, Proc. of the IEEE Wireless Communication and Networking Conference (WCNC'06), pp. 264-269, 2006.

[4] C. Ambuhl, A. E. F. Clementi, M. D. Ianni, N. Lev-Tov, A. Monti, D. Peleg, G. Rossi and R. Silvestri, *Efficient algorithms for low-energy bounded-hop broadcast in ad-hoc wireless networks*, Proc. of the 21st Annual Symposium on Theoretical Aspects of Computer Science (STACS'04), LNCS-2996, pp. 418-427, 2004.

[5] C. Ambuhl, A.E.F. Clementi, M. D. Ianni, G. Rossi, A. Monti, and R. Silvestri, *The range assignment problem in non-homogeneous static ad-hoc networks*, Proc. of the 18th International Parallel and Distributed Precessing Symposium (IPDPS'04), 2004.

[6] E. Althaus, G. Calinescu, I. I. Mandoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky, *Power efficient range assignment for symmetric connectivity in static ad-hoc wireless networks*, Wireless Networks, vol. 12, pp. 287-299, 2006.

[7] C. Ambuhl, A. E. F. Cleminti, P. Penna, G. Rossi and R. Silvestri, *Energy consumption in radio networks: selfish agents and rewarding mechanisms*, Proc. of the 10th International Colloquim on Structural Information Complexity (SIROCCO'03), pp. 1-16, 2003.

[8] A. Abdalla, N. Deo and P. Gupta, *Heuristics to compute a diameter-constrained MST*, Congressus Numerantium, vol. 144, pp. 161-182, 2000.

[9] E. Althaus, S. Funke, S. Har-Peled, J. Konemann, E. A. Ramos and M. Skutella, *Approximating k-hop minimum spanning trees*, Operations Research Letters, vol. 33, pp. 115-120, 2005.

[10] A. Aggarwal and M. Klawe, *Applications of generalized matrix searching to geometric algorithms*, Discrete Applied Mathematics, vol. 27, pp. 3-23, 1990.

[11] L. Alfandari and V. T. Paschos, *Approximating minimum spanning tree of depth 2*, International Trans. in Operation Research, vol. 6, pp. 607-622, 1999.

[12] S. Arora, P. Raghavan and S. Rao, *Approximation schemes for Euclidean k-medians and related problems*, Proc. of the 30th ACM Symposium on Theory of Computing (STOC'98), pp. 106-113, 1998.

[13] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, *A survey on sensor networks*, IEEE Communication Magazine, vol. 40, pp. 102-114, 2002.

[14] A. Balakrishnan and K. Altinkemer, *Using a hop constrained model to generate alternative communication network design*, INFORMS Journal of Computing, vol. 4, pp. 192-205, 1992.

[15] L. Booth, J. Bruck, M. Franceschetti and R. Meester, *Covering algorithms, continuum percolation and the geometry of wireless networks*, The Annals of Applied Probability, vol. 13, pp. 722-741, 2003.

[16] M. A. Bassiouni and C. Fang, *Dynamic channel allocation for linear macrocellular topology*, Proc. of the 14th ACM Symposium on Applied Computing (SAC'99), pp. 382-388, 1999.

[17] A. Boukerche, X. Fei and R. B. Araujo, *An energy aware coverage-preserving scheme for wireless sensor networks*, Proc. of the 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN'05), pp. 205-213, 2005.

[18] H. Bronnimann and M. T. Goodrich, *Almost optimal set covers in finite VC-dimension*, Discrete & Computational Geometry, vol. 14, pp. 463-479, 1995.

[19] M. de Berg, M. Van Kreveld, M. Overmarse and O. Schwarzkopf, *Computational geometry algorithms and applications*, Springer-Verlag, 1997.

[20] P. Bose and G. T. Toussaint, *Computing the constrained Euclidean geodesic and link center of a simple polygon with applications*, Proc. of the 14th Conference on Computer Graphics International (CGI'96), pp. 102-110, 1996.

[21] T. M. Chan, *More planar two-center algorithms*, Computational Geometry: Theory and Application, vol. 13, pp 189-198, 1999.

[22] F. A. Chudak, *Improved appromation algorithms for uncapacited facility location*, Proc. of the 6th Conference on Integer Programming and Combinatorial Optimization (IPCO'98), LNCS-1412, pp. 180-194, 1998.

[23] T. Calamoneri, A. E. F. Clementi, M. D. Ianni, M. Lauria, A. Monti and R. Silvestri, *Minimum energy broadcast and disk cover in grid wireless networks*, Proc. of the 13th International Colloquium, Structural Information and Communication Complexity (SIROCCO'06), LNCS-4056, pp. 227-239, 2006.

[24] A. E. F. Clementi, P. Crescenzi, P. Penna, G. Rossi and P. Vocca, *On the complexity of computing minimum energy consumption broadcast subgraphs*, Proc. of the 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS'01), LNCS-2010, pp. 121-131, 2001.

[25] M. X. Cheng, M. Cardei, J. Sun, X. Cheng, L. Wang, Y. Xu and D. -Z. Du, *Topology control of ad hoc wireless networks for energy efficiency*, IEEE Trans. on Computers, vol. 53, pp. 1629-1635, 2004.

[26] J. Choong and D. Everitt, *Outage analysis and spectral efficiency of indoor mobile radio systems*, Proc. of the 45th IEEE Vehicular Technology Conference (VTC'95), pp. 434-438, 1995.

[27] B. Chazelle and L. J. Guibas, *Fractional cascading: I. a data structuring technique*, Algorithmica, vol. 1, pp. 133-162, 1986.

[28] M. Cagalj, J. P. Hubaux and C. Enz, *Minimum-energy broadcast in all-wireless networks : NP-completeness and distribution issues*, Proc. of the 8th Annual ACM SIGMOBILE International Conference on Mobile Computing and Networking (MOBI-COM'02), pp. 172-182, 2002.

[29] A. E. F. Clementi, G. Huiban, P. Penna, G. Rossi and Y. C. Verhoeven, *Some recent theoretical advances and open questions on energy consumption in ad-hoc wireless networks*, Proc. of the 3rd Workshop on Approximation and Randomization Algorithms in Communication Networks (ARACNE'02), pp. 23-38, 2002.

[30] A. E. F. Clementi, M. D. Ianni, A. Monti, G. Rossi and R. Silvestri, *Experimental analysis of practically efficient algorithms for bounded-hop accumulation in ad-hoc wireless networks*, Proc. of the 19th International Parallel and Distributed Processing Symposium (IPDPS'05), 2005.

[31] A. E. F. Clementi, M. D. Ianni and R. Silvestri, *The minimum broadcast range assignment problem on linear multi-hop wireless networks*, Theoretical Computer Science, vol. 299, pp. 751-761, 2003.

[32] P. Carmi and M. J. Katz, *Power assignment in radio networks with two power levels*, Algorithmica, vol. 47, pp. 183-201, 2007.

[33] P. Crescenzi and V. Kann, *A compendium of NP optimization problems*, http://www.nada.kth.se/~viggo/problemlist/compendium.html.

[34] I. Caragiannis, C. Kaklamanis and P. Kanellopoulos, *A logarithmic approximation algorithm for the minimum energy consumption broadcast subgraph problem*, Information Prosessing Letters, vol. 86, pp. 149-154, 2003.

[35] G. Calinescu, S. Kapoor, A. Olshevsky and A. Zelikovsky, *Network lifetime and power assignment in ad hoc wireless networks*, Proc. of the 11th annual European Symposium (ESA'03), LNCS-2832, pp. 114-126, 2003.

[36] G. Calinescu, S. Kapoor and M. Sarwat, *Bounded-hops power assignment in ad hoc wireless networks*, Discrete Applied Mathematics, vol. 154, pp. 1358-1371, 2006.

[37] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to algorithms*, Prentice Hall (India) Pvt. Ltd., New Delhi, 1998.

[38] G. Calinescu, I. I. Mandoiu, P. -J. Wan and A. Z. Zelikovsky, *Selecting forwarding neighbors in wireless ad hoc networks*, Mobile Networks and Applications, vol. 9, pp. 101-111, 2004.

[39] A. E. F. Clementi, P. Penna, A. Ferreira, S. Perennes and R. Silvestri, *The minimum range assignment problem on linear radio networks*, Algorithmica, vol. 35, pp. 95-110, 2003.

[40] A. E. F. Clementi, P. Penna and R. Silvestri, *The power range assignment problem in radio networks on the plane*, Proc. of the 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS'00), LNCS-1770, pp. 651-660, 2000.

[41] A. E. F. Clementi, P. Penna and R. Silvestri, *On the power assignment problem in radio networks*, Mobile Networks and Applications, vol. 9, pp. 125-140, 2004.

[42] Y. -C. Cheng and T. G. Robertazzi, *Critical connectivity phenomena in multihop radio models*, IEEE Trans. on Communications, vol. 37, pp. 770-777, 1989.

[43] J. Cartigny, D. Simplot and I. Stojmenovic, *Localized energy efficient broadcast for wireless networks with directional antennas*, Proc. of the 1st Annual Mediterranean Workshop on Ad Hoc Networks (Med-Hoc-Net'02), 2002.

[44] J. Cartigny, D. Simplot and I. Stojmenovic, *Localized minimum energy broadcasting in ad hoc networks*, Proc. of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03), pp. 2210-2217, 2003.

[45] D. Coppersmith and S. Winograd, *Matrix multiplication via arithmetic progression*, Journal of Symbolic Computation, vol. 9, pp. 251-280, 1990.

[46] J. Chlebikova, D. Ye and H. Zhang, *Assign ranges in general ad hoc networks*, Journal of Parallel and Distributed Computing, vol. 66, pp. 489-498, 2006.

[47] N. Deo and A. Abdalla, *Computing a diameter constrained minimum spanning tree in parallel*, Proc. of the 4th Italian Conference on Algorithms and Complexity (CIAC'00), LNCS-1767, pp. 17-31, 2000.

[48] K. Diks, E. Kranakis, D. Krizanc and A. Pelc, *The impact of knowledge on broadcasting time in radio networks*, Proc. of the 7th Annual European Symposium on Algorithms (ESA'99), LNCS-1643, pp. 41-52, 1999.

[49] A. K. Das, R. J. Marks, M. E. -Sharkawi, P. Arabshahi and A. Gray, *Minimum power broadcast trees for wireless networks: integer programming formulations*, Proc. of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03), pp. 1001-1010, 2003.

[50] D. Eppstein, *Spanning trees and spanners* in Handbook of Computational Geometry, Ed. J. R. Sack and J. Urrutia, Elsevier, New York, 2000.

[51] J. Elzinga and D. W. Hearn, *Geometrical solutions to some minimax location problems*, Transputer Science, vol. 6, pp. 379-394, 1972.

[52] L. Ferrari, *Solving a Quartic (4th degree) Polynomial*, http://www.karlscalculus.org/quartic.html.

[53] B. Fuchs, *On the hardness of range assignment problem*, Proc. of the Tech. Rep. TR05-113, Electronic Colloquium on Computational Complexity (ECCC), 2005.

[54] S. J. Fortune, D. M. Gay, B. W. Kernighan, O. Landron, R. A. Valenzuela and M. H. Wright, *WISE design of indoor wireless systems: practical computation and optimization*, IEEE Computational Science Engineering, Vol. 2, pp. 58-68, 1995.

[55] M. Flammini, A. Navarra, R. Klasing and S. Perennes, *Improved approximation results for the minimum energy broadcasting problem*, Proc. of the 2004 joint Workshop on Foundation of Mobile Computing, pp. 85-91, 2004.

[56] R. J. Fowler, M. S. Paterson and S. L. Tanimoto, *Optimal packing and covering in the plane are NP-complete*, Information Processing Letters, vol. 12, pp. 133-137, 1981.

[57] L. Gouveia, *Using the Miller-Tucker-Zemlin constraints to formulate a minimal spanning tree problem with hop constraints*, Computers and Operations Research, vol. 22, pp. 959-970, 1995.

[58] L. Gouveia, *Multicommodity flow models for spanning tree with hop constraints*, European Journal of Operational Research, vol. 95, pp. 178-190, 1996.

[59] H. N. Gabow, Z. Galil, T. H. Spencer and R. E. Tarjan, *Efficient algorithms for finding minimum spanning trees in undirected and directed graphs*, Combinatorica, vol. 6, pp. 109-122, 1986.

[60] S. Guha and S. Khuller, *Greedy strikes back: improved facility location algorithms*, Journal of Algorithms, vol. 31, pp. 228-248, 1999.

[61] C. Gaibisso, G. Proietti, and R. B. Tan, *Efficient management of transient station failures in linear radio communication networks with bases*, Journal of Parallel and Distributed Computing, vol. 66, pp. 556-565, 2006.

[62] L. Gouveia and C. Requejo, *A new lagrangian relaxation approach for the hop-constrained minimum spanning tree problem*, European Journal of Operational Research, vol. 132, pp. 539-552, 2001.

[63] M. Haenggi, *Twelve reasons not to route over many short hops*, Proc. of the 60th IEEE Vehicular Technology Conference (VTC'04), pp. 3130-3134, 2004.

[64] Y. Han, *An $O(n^3(\frac{\log \log n}{\log n})^{\frac{5}{4}})$ time algorithm for all pairs shortest paths*, Proc. of the 14th Annual European Symposium (ESA'06), LNCS-4168, pp. 411-417, 2006.

[65] R. Z. Hwang, R. C. Chang and R. C. T. Lee, *The searching over separators strategy to solve some NP-hard problems in subexponential time*, Algorithmica, vol. 9, pp. 398-423, 1993.

[66] I. Howitt and S.-Y. Ham, *Base station location optimization*, Proc. of the 50th IEEE Vehicular Technology Conference (VTC'99) , pp. 2067-2071, 1999.

[67] A. Heppes and H. Melissen, *Covering a rectangle with equal circles*, Periodica Mathematica Hungarica, vol. 34, pp. 65-81, 1997.

[68] D. S. Hochbaum and W. Maass, *Approximation schemes for covering and packing problems in image processing and VLSI*, Journal of the ACM, vol. 32, pp. 130-136, 1985.

[69] D. S. Hochbaum and D. Shmoys, *A best possible heuristic for the k-center problem*, Mathematics of Operations Research, vol. 10, pp. 180-184, 1985.

[70] F. Hurtado, V. Sacristan and G. Toussaint, *Some constrained minimax and maximin location problems*, Studies in Locational Analysis, vol. 15, pp. 17-35, 2000.

[71] Z. J. Haas and S. Tabrizi, *On some challenges and design choices in ad-hoc communications*, Proc. of the IEEE military communication conference (MILCOM'98), pp. 187-192, 1998.

[72] F. Ingelrest and D. Simplot-Ryl, *Localized broadcast incremental power protocol for wireless ad hoc networks*, Proc. of the 10th IEEE Symposium on Computers and Communications (ISCC'05), pp. 28-33, 2005.

[73] J. W. Jaromczyk and M. Kowaluk, *An efficient algorithm for the Euclidean two-center problem*, Proc. of the 10th Annual Symposium on Computational Geometry (SoCG'94), pp. 303-311, 1994.

[74] L. Kucera, *Low degree connectivity in ad hoc networks*, Proc. of the 13th Annual European Symposium on Algorithms (ESA'05), LNCS-3669, pp. 203-214, 2005.

[75] E. D. Kaplan, *Understanding GPS: Principles and Applications*, Second Edition, Artech House, Norwood, MA 02062, USA.

[76] L. M. Kirousis, E. Kranakis, D. Krizanc and A. Pelc, *Power consumption in packet radio networks*, Theoretical Computer Science, vol. 243, pp. 289-305, 2000.

[77] E. Kranakis, D. Krizanc and A. Pelc, *Fault-tolerant broadcasting in radio networks*, Proc. of the 6th Annual European Symposium on Algorithms (ESA'98), LNCS-1461, pp. 283-294, 1998.

[78] S. O. Krumke, R. Liu, E. L. Lloyd, M. V. Marathe, R. Ramanathan and S. S. Ravi, *Topology control problems under symmetric and asymmetric power thresholds*, Proc. of the 2nd International Conference on Ad-Hoc, Mobile, and Wireless Networks (ADHOC-NOW'03), LNCS-2865, pp. 187-198, 2003.

[79] R. Klasing, A. Navarra, A. Papadopoulos and S. Perennes, *Adaptive broadcast consumption(ABC), a new heuristic and new bounds for the minimum energy broadcast routing problem*, Proc. of the 3rd International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communication (Networking'04), LNCS-3042, pp. 866-877, 2004.

[80] E. Kantor and D. Peleg, *Approximate hierarchical facility location and applications to the shallow steiner tree and range assignment problems*, Proc. of the 6th Italian Conference on Algorithms and Complexity (CIAC'06), LNCS-3998, pp. 211-222, 2006.

[81] M. R. Korupolu, C. G. Plaxton and R. Rajaraman, *Analysis of a local search heuristic for facility location problems*, Proc. of the 9th Annual ACM/SIAM Symposium on Discrete Algorithms (SODA'98), pp. 1-10, 1998.

[82] S. K. Kim and C. -S. Shin, *Efficient algorithms for two-center problems for a convex polygon*, Proc. of the 6th Annual International Conference on Computing and Combinatorics (COCOON'00), LNCS-1858, pp. 299-309, 2000.

[83] G. S. Lauer, *Packet radio routing*, Chapter 11 of routing in communication networks, M. Streensstrup (ed.), pp. 351-396, Prentice-Hall, 1995.

[84] D. T. Lee, *Farthest neighbor Voronoi diagrams and applications*, Report80-11-FC-04, Dept. Elect. Engrg. Comput. Sci., Northwestern Univ., Evanston, IL, 1980.

[85] X. -Y. Li, *Algorithmic, geometric and graph issues in wireless networks*, Wireless Communications and Mobile Computing, vol. 3, pp. 119-140, 2003.

[86] A. Marchetti-Spaccamela, *The p-center problem in the plane is NP-complete*, Proc. of the 19th Allerton Conference on Communication, Control and Computing, pp. 31-40, 1981.

[87] N. Megiddo, *Linear-time algorithms for linear programming in R^3 and related problems*, SIAM Journal on Computing, vol. 12, pp. 759-776, 1983.

[88] R. Marks, A. K. Das, M. El-Sharkawi, P. Arabshahi and A. Gray, *Minimum power broadcast trees for wireless networks: Optimizing using the viability lemma*, Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS'02), pp. 273-276, 2002.

[89] S. Meguerdichian, F. Koushanfar, M. Potkonjak and M. B. Srivastava, *Coverage problems in wireless ad-hoc sensor networks*, Proc. of the 20th Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM'01), pp. 1380-1387, 2001.

[90] R. Mathar and J. Mattfeldt, *Optimal transmission ranges for mobile communication in linear multihop packet radio networks*, Wireless Networks, vol. 2, pp. 329-342, 1996.

[91] K. Mehlhorn and S. Naher, *The LEDA platform of combinatorial and geometric computing*, Cambridge University Press, 1999.

[92] J. B. M. Melissen and P. C. Schuur, *Improved coverings of a square with six and eight circles*, Electronic Journal on Combinatorics, vol. 3, R32, 1996.

[93] J. B. M. Melissen and P. C. Schuur, *Covering a rectangle with six and seven circles*, Discrete Applied Mathematics, vol. 99, pp. 149-156, 2000.

[94] M. Mahdian, Y. Ye and J. Zhang, *Approximation algorithms for the metric facility location problems*, SIAM Journal on Computing, vol. 36, pp. 411-432, 2006.

[95] A. Navarra, *Tight bounds for the minimum energy broadcasting problem*, Proc. of the 3rd International Symposium on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks (WiOpt'05), pp. 313-322, 2005.

[96] K. J. Nurmela, *Conjecturally optimal coverings of an equilateral triangle with up to 36 equal circles*, Experimental Mathematics, vol. 9, pp. 241-250, 2000.

[97] K. J. Nurmela and P. R. J. Ostergard, *Covering a square with up to 30 Equal Circles*, Research Report HUT-TCS-A62, Laboratory for Theoretical Computer Science, Helsinky University of Technology, 2000.

[98] P. Piret, *On the connectivity of radio networks*, IEEE Trans. on Information Theory, vol. 37, pp. 1490-1492, 1991.

[99] J. Plesnik, *A heuristic for the p-center problems in graphs*, Discrete Applied Mathematics, vol. 17, pp. 263-268, 1987.

[100] I. Papadimitriou and L. Geordiadis, *Minimum energy broadcasting in multi-hop wireless networks using a single broadcast tree*, Mobile Networks and Applications, vol. 11, pp. 361-375, 2006.

[101] K. Pahlavan and A. Levesque, *Wireless Information Networks*, John Wiley, New York, 1995.

[102] J. Park and S. Sahni, *Maximum lifetime broadcasting in wireless networks*, IEEE Trans. on Computers, vol. 54, pp. 1081-1090, 2005.

[103] S. A. Plotkin, D. B. Shmoys and E. Tardos, *Fast approximation algorithms for fractional packing and covering problems*, Mathematics of Operations Research, vol 20, pp. 257-301, 1995.

[104] G. Rossi, *The range assignment problem in static ad hoc networks*, Ph. D. Thesis, University of Siena, 2003.

[105] S. Roy, D. Bardhan and S. Das, *Efficient algorithm for placing base stations by avoiding forbidden zone*, Proc. of the 2nd International Conference on Distributed Computing and Internet Technology (ICDCIT'05), LNCS-3816, pp. 105-116, 2005.

[106] G. R. Raidl and B. A. Julstrom, *Greedy heuristics and an evolutionary algorithm for the bounded-diameter minimum spanning tree problem*, Proc. of the 18th ACM Symposium on Applied Computing (SAC'03), pp. 747-752, 2003.

[107] S. Roy, A. Karmakar, S. Das and S. C. Nandy, *Constrained minimum enclosing circle with center on a query line segment*, Proc. of the 31st International Symposium on Mathematical Foundations of Computer Science (MFCS'06), LNCS-4162, pp. 765-776, 2006.

[108] Y. Rabinovich and R. Raz, *Lower bounds on the distortion of embedding finite metric spaces in graphs*, Discrete Computational Geometry, vol. 19, pp. 79-94, 1998.

[109] H. Rademacher and O. Toeplitz, *The Enjoyment of Mathematics*, Princeton University Press, 1957.

[110] D. Raychaudhuri and N. D. Wilson, *ATM-based transport architecture for multiservices wireless personal communication networks*, IEEE Journal on Selected Areas in Communications, vol. 12, pp. 1401-1414, 1994.

[111] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*, John Wiley & Sons, 2005.

[112] P. Santi, *Topology control in wireless ad-hoc and sensor networks*, ACM Computing Surveys, vol. 37, pp. 164-194, 2005.

[113] M. Sharir, *A near-linear algorithm for the planar 2-center problem*, Discrete Computational Geometry, vol. 18, pp. 125-134, 1997.

[114] J. J. Sylvester, *A question in the geometry of situation*, Quarterly Journal of mathematics, vol. 1, p. 79, 1857.

[115] P. Santi, D. M. Blough and F. S. Vainstein, *A probabilistic analysis for the range assignment problem in ad hoc networks*, Proc. of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'01), pp. 212-220, 2001.

[116] D. Stamatelos and A. Ephremides, *Spectral efficiency and optimal base placement for indoor wireless networks*, IEEE Journal on Selected Areas in Communications, Vol. 14, pp. 651-661, 1996.

[117] M. I. Shamos and D. Hoey, *Closest-point problems*, Proc. of the 16th Annual IEEE Symposium on Foundations of Computer Science (FOCS'75), pp. 151-162, 1975.

[118] S. Sohn and G. Jo, *Optimization of base stations positioning in mobile networks*, Proc. of the 2006 International Conference on Computational Science and its Applications (ICCSA'06), LNCS-3981, pp. 779-787, 2006.

[119] C. -S. Shin, J. -H. Kim, S. K. Kim and K. -Y. Chwa, *Two-center problems for a convex polygon*, Proc. of the 6th Annual European Symposium on Algorithms (ESA'98), LNCS-1461, pp. 199-210, 1998.

[120] H. D. Sherali, C. H. Pendyala and T. S. Rappaport, *Optimal location of transmitters for micro-cellular radio communication system design*, IEEE Journal on Selected Areas in Communications, vol. 14, pp. 662-673, 1996.

[121] D. B. Shmoys, E. Tardos and K. Aardal, *Approximation algorithm for the facility location problems*, Proc. of the 29th Annual ACM Symposium on Theory of Computing (STOC'97), pp. 265-274, 1997.

[122] I. Stojmenovic and J. Wu, *Broadcasting and activity scheduling in ad hoc networks*, Mobile Ad Hoc Networking, (S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, eds.), IEEE/Wiley, pp. 205-229, 2004.

[123] G. T. Toussaint, *The relative neighborhood graph of a finite planar set*, Pattern Recognition, vol. 12, pp. 261-268, 1980.

[124] T. Tarnai and Z. Gasper, *Covering a square by equal circles*, Elem. Math., vol. 50, pp. 167-170, 1995.

[125] S. Ulukus and R. D. Yates, *Stochastic power control for cellular radio systems*, IEEE Trans. on Communications, vol. 46, pp. 784-798, 1998.

[126] S. Verblunsky, *On the least number of unit circles which can cover a square*, Journal of London Mathematical Society, vol. 24, pp. 164-170, 1949.

[127] S. Voß, *The steiner tree problem with hop constraint*, Annals of Operations Research, vol. 86, pp. 321-345, 1999.

[128] M. H. Wright, *Optimization methods for base station placement in wireless applications*, Proc. of the 48th IEEE Vehicular Technology Conference (VTC'98), pp. 387-391, 1998.

[129] P. -J. Wan, G. Calinescu, X. -Y. Li and O. Frieder, *Minimum-energy broadcasting in static ad hoc wireless networks*, ACM Wireless Networks, vol. 8, pp. 607-617, 2002.

[130] J. E. Wieselthier, G. D. Nguyen and A. Ephremides, *On the construction of energy efficient broadcast and multicast trees in wireless networks*, Proc. of the 19th Annual

Joint Conference of the IEEE Computer and Communication Societies (INFOCOM'00), pp. 585-594, 2000.

[131] J. E. Wieselthier, G. D. Nguyen and A. Ephremides, *The energy efficiency of distributed algorithms for broadcasting in ad hoc networks*, Proc. of the 5th International Symposium on Wireless Personal Multimedia Communications (WPMC'02), pp. 499-503, 2002.

[132] S. -Y. Yang and A. Ephremides, *Optimal network design : the base station placement problem*, Proc of the 36th Conference on Decision and Control (CDC'97), pp. 2381-2386, 1997.

Publications from the Contents of the Thesis

Journal Publications

1. G. K. Das and S. C. Nandy, *Weighted broadcast range assignment in linear radio networks*, Information Processing Letters (accepted), 2007.

2. G. K. Das, S. Roy, S. Das and S. C. Nandy, *Variations of base station placement problem on the boundary of a convex region*, International Journal of Foundations of Computer Science, (accepted), 2007.

3. G. K. Das, S. C. Ghosh and S. C. Nandy, *Improved algorithm for minimum cost range assignment problem for linear radio networks*, International Journal of Foundations of Computer Science, vol 18, pp. 619-635, 2007.

4. G. K. Das, S. Das, S. C. Nandy and B. P. Sinha, *Efficient algorithm for placing a given number of base stations to cover a convex region*, Journal of Parallel and Distributed Computing, vol. 66, pp. 1353-1358, 2006.

5. G. K. Das, S. Das and S. C. Nandy, *Range assignment for energy efficient broadcasting in linear radio networks*, Theoretical Computer Science, vol. 352, pp. 332-341, 2006.

Conference/Workshop Publications

1. G. K. Das and S. C. Nandy, *Weighted broadcast in linear radio networks*, Proc. of the International Conference on Algorithmic Aspects in Information and Management (AAIM'06), LNCS-4041. pp. 343-353, 2006.

2. G. K. Das, S. Das and S. C. Nandy, *Homogeneous 2-hops broadcast in 2D*, Proc. of the International Conference on Computational Science and its Application (ICCSA'06), LNCS-3981, pp. 750-759, 2006.

3. G. K. Das, S. Das, S. C. Nandy and B. P. Sinha, *Placing a given number of base stations to cover a convex region*, Proc. of the International Workshop on Distributed Computing (IWDC'05), LNCS-3741, pp. 57-62, 2005.

4. G. K. Das, S. C. Ghosh and S. C. Nandy, *Improved algorithm for minimum cost range assignment problem for linear radio networks*, Proc. of the International Workshop on Distributed Computing (IWDC'04), LNCS-3326, pp. 412-423, 2004.

5. G. K. Das, S. Das and S. C. Nandy, *Efficient algorithm for energy efficient broadcasting in linear radio networks*, Proc. of the International Conference on High Performance Computing (HiPC'04), LNCS-3296, pp. 420-429, 2004.

6. G. K. Das, S. C. Ghosh and S. C. Nandy, *An efficient heuristic algorithm for 2D h-hops range assignment problem*, Proc. of the IEEE Global Telecommunication Conference (GLOBECOM'04), vol. 2, pp. 1051-1055, 2004.