Cooperative localization in wireless ad-hoc networks based on vector-addition localization scheme with fast message dissemination

Wang, Wei

2009

Wang, W. (2009). Cooperative localization in wireless ad-hoc networks based on vector-addition localization scheme with fast message dissemination. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/15158>

<https://doi.org/10.32657/10356/15158>

**Cooperative Localization in Wireless Ad-hoc Networks
based on Vector-addition Localization Scheme with Fast
Message Dissemination**

Wang Wei

School of Electrical & Electronic Engineering

Nanyang Technological University

A thesis submitted for the degree of

Master of Engineering

2009

Acknowledgements

I am deeply indebted to my thesis supervisor, Prof. Soong Boon-Hee, for his invaluable advice, support and patience during the course of my study. I also would like to thank our school for the financial support (grant RGM 12/03 and RGM 31/03) on my work.

I am also grateful to the technicians in PWTC who have consistently assisted my research work, as well as to the labmates there who have made my life in PWTC an enjoyable experience.

This thesis is dedicated to my parents and Emma, whose encouragements have impelled me in every aspect of my life.

Summary

Cooperative localization in wireless ad-hoc networks deals with how to utilize local ranging measurements for locating remote nodes. In view of both the pervasive deployment of wireless ad-hoc networks and the incapability of wireless nodes for long-distance ranging, cooperative localization can be found in various applications. This thesis presents the research contributions to a number of theoretical and practical problems about cooperative localization, which can be grouped into two categories.

The first contribution consists of using a simple Local Coordinate System for map construction, devising a Vector-addition Localization Scheme for anchor-free cooperative localization, and developing optimal path selection mechanisms for such vector-addition scheme to achieve better accuracy. Two mechanisms for optimal path selection are proposed in this thesis. The first mechanism is based on some geometric heuristics about local coordinate system, which is simple yet universal. Our second mechanism discloses some underlying properties of vector-addition localization scheme, and it makes use of Cramer-Rao Lower Bound (CRLB) as well as geometric principles for optimal path selection. Both mechanisms are able to produce, in term of localization accuracy, better solutions than conventional arbitrary shortest-path approach.

The second part of our research work is focused on how to quickly disseminate the local ranging measurement information to the whole wireless ad-hoc network. Such fast message dissemination strategies are generic and thus applicable to many other general operations in wireless ad-hoc networks. Our main contribution in this area is to provide, for collision and collision-free cases, an approximate algorithm that maximizes the number of receptions in each dissemination wave. Our research work does not only

produce faster dissemination strategies than previous methods as proved by numerical results, but also sheds light on the underlying theoretical aspects of the generic fast message dissemination problem.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Design Challenges	4
1.3	Thesis Contributions	6
1.4	Thesis Outline	6
2	Local Coordinate System and Vector-addition Localization Scheme	8
2.1	Local Coordinate System	8
2.1.1	Construction of Local Coordinate System	8
2.1.2	Manipulations of Local Coordinate System	10
2.2	Multi-Dimensional Scaling with Refinement	11
2.2.1	Least Square Error Minimization	12
2.2.2	Multidimensional Scaling for Position Estimation	13
2.2.3	Variants of “MDS with Refinement” Approach	14
2.3	Vector-addition Localization Scheme	16
3	A Heuristics for Vector-addition Path Selection	18
3.1	Confidence Index of Vector	18
3.2	Performance Evaluation and Analysis	20
4	CRLB-based Path Selection Algorithm	23
4.1	CRLB in Anchored Network	23
4.2	CRLB in Anchor-less Network	26
4.3	Path Selection based on CRLB	27
4.4	Performance Evaluation and Analysis	32
4.4.1	CRLB in Local Maps with Different Map Sizes	32

CONTENTS

4.4.2	Optimal Path Selection for Locating Remote Nodes	33
4.4.3	Deviation of Optimal Path	38
4.4.4	Communication and Computational Overheads of Vector Addition Scheme	38
5	Fast Message Dissemination: Collision-prohibited Case	40
5.1	The Wave Expansion Approach	41
5.2	Our Proposed Solution for Collision-free Case	42
5.3	Numerical Results	45
6	Fast Message Dissemination: Collision-allowed Case	48
6.1	Problem Formulation and Illustration	49
6.1.1	Problem Formulation	49
6.1.2	An Example for Illustration	50
6.2	Posiform Maximization	51
6.2.1	Basic Concepts	51
6.2.2	Constructing Conflict Graph	52
6.3	Maximum Stable Set of Conflict Graph	53
6.3.1	Greedy Algorithm	53
6.3.2	Performance Lower Bound	54
6.4	Numerical Results	57
7	Conclusions and Recommendations	60
A	Author's Publications	63
	References	64

List of Figures

1.1	Trilateration technique in GPS	3
1.2	Comparison between triangulation and cooperative localization	4
1.3	Example of relative localization	6
2.1	Setup of LCS	9
2.2	Reflection decision making	10
2.3	Computing rotation angle	11
2.4	A mass-spring system	12
2.5	Locating remote node through vector addition	16
3.1	Example of vector's over-determinacy	19
3.2	Impact of bearing errors on end-to-end distance estimate	19
3.3	Network maps used in simulation	21
3.4	Selected PDE results of square-shape map	21
3.5	Selected PDE results of C-shape map	22
4.1	Visualization of CRLB in anchored network	25
4.2	Visualization of CRLB in N_0 's local map	28
4.3	Selection of positive X-axis of LCS and rotation of LCS when adding vectors' variance	29
4.4	Impact of X-axis's vibration on variance of remote node's position estimate	30
4.5	Average CRLB of 1-hop neighbors in local maps with different sizes . .	33
4.6	Path setup in square-shape and U-shape topologies	35
4.7	Predicted variance bound versus simulated variance	37
4.8	Degree of path deviation with respect to different ranging errors	38

LIST OF FIGURES

5.1	Bipartite graph representation	42
5.2	Construction of weighted conflict graph	43
5.3	Pseudo-code of proposed algorithm.	44
5.4	Performance result of $D-max$ with respect to network depth	45
5.5	Performance result of the number of transmissions	46
6.1	An example of typical collision-allowed SE problem.	50
6.2	Constructing G_f	53
6.3	Overlap of $TRUE$ entries in connectivity matrix C	54
6.4	Structure of connectivity matrix C that generates maximal φ	56
6.5	Performance results of the difference between $D-max$ and network depth	58
6.6	Performance results of the number of transmissions	59

Chapter 1

Introduction

1.1 Motivation

Technological advancement in the past decade have made it possible for civilians to set up practical Wireless Ad-hoc Network whose concept was originally proposed by defense agencies almost half century ago. Thanks to the pervasive usage of personal laptops and Personal Digital Assistant (PDA), Mobile Ad-hoc Network (MANET), which is a typical wireless ad-hoc network, is no longer the proprietary technology of defense agencies. Instead, MANET has started to play an important role in civilian applications, such as rescue missions and expedition missions in remote areas where network infrastructure is unavailable. The spontaneous network [11] built on MANET is able to support virtual conferencing type of applications so that users could discuss their ongoing tactics to achieve the goal of their mission. Besides common multihop communication capability, MANET would bring users great convenience if it is able to report nodes' position information as well. Take a rescue mission for example. A group of rescuers is dispatched to search for a boy trapped in forest, and they use MANET to coordinate their movement. The availability of nodes' positions would greatly ease the task to coordinate the formation of this rescue team. Besides, when some rescuer finds the boy, the other rescuers could immediately know the boy's location and start to move towards that location to provide additional help.

Position information is also essential for Wireless Sensor Network (WSN), which is another subtype of wireless ad-hoc network. As Micro-Electro-Mechanical System (MEMS) and Radio Frequency (RF) technologies experienced drastic advancement in

1.1 Motivation

the past few decades, it becomes feasible for people to massively deploy tiny yet multi-functional sensors to complete certain dedicated task in the region of interest. These dedicated tasks of WSN may include inventory management [22], animal tracking [19], and military surveillance [16] etc, all of which share a common feature of providing sensed data related to a particular location. For example in military surveillance application where WSN is used to detect enemy's activities at battlefield, the system should not only alert commander about an intrusion as soon as possible but also precisely indicate where the intrusion comes from. Similarly in the application of inventory management in logistics, accurate location information about where each sensor resides is also crucial to the functioning of this monitoring system.

Therefore, besides fundamental communication capability, we shall also incorporate localization capability into the system of wireless ad-hoc network in order to enhance its applicability. It shall be noted that it may not be necessary for each node of wireless ad-hoc network to know other nodes' absolute positions, since the knowledge of *relative positions* is enough for completing certain tasks such as the coordination of nodes' formation in MANET and the decision making of geographical routing in WSN.

The most straightforward way to incorporate localization capability into wireless ad-hoc network is through Global Positioning System (GPS) receiver, which makes use of received radio signals from line-of-side (LOS) satellites to determine location and even speed. The underlying principle of GPS follows trilateration or multilateration techniques, as illustrated in Fig. 1.1. While GPS has been widely used in many aspects such as navigation and map-making, there are a number of implicit drawbacks that make GPS less favorable in wireless ad-hoc network. The most conspicuous reason is because of GPS receiver's cost, especially as compared with the cheap nodes in WSN. It is feasible to equip a few sensor nodes with GPS receiver, but cost would soar up if we do the same thing on every sensor node. Most sensor nodes in WSN systems are difficult to retract once they are deployed, and thus overall cost shall be one of the most important concerns to designers. Another drawback is that GPS receiver could dissipate additional power supply from wireless nodes. Owing to its operating scenario, node in wireless ad-hoc network is only able to draw power from attached battery which is bulky and heavy. The introduction of power-hungry GPS receiver would inevitably frustrates designers who have been struggling to strike a balance between reducing each node's size and prolonging its lifetime. Thirdly, the positioning accuracy of common

1.1 Motivation

civilian GPS receivers, which is about 10 meters in horizontal dimension, may not meet the requirement of some wireless ad-hoc network systems. Although the accuracy could be improved to centimeter scale through techniques like Differential GPS (DGPS), the issues of cost and energy would rise again. Last but not least, GPS receivers could not operate correctly when their direct LOS to satellites is blocked. In other words, the applicability of GPS receivers is restricted to open space only, and designers have to turn to some other localization techniques when their networks are to be deployed in non-LOS places such as warehouses and forest.

Fig. 1.1: Trilateration technique in GPS: distance estimates between GPS receiver and three satellites A, B and C, whose positions are pre-known, produce two possible position estimates (the two blue crosses); one of these position estimates could be eliminated if it is far from the place of interest; with more satellites, multilateration can be used to improve accuracy of position estimate.

The desire to obtain nodes' position information and the deficiency of GPS receivers in wireless ad-hoc network has motivated us to seek alternative method to locate nodes in a wireless ad-hoc network. Fortunately, thanks to accurate ranging techniques that give distance estimate between two adjacent nodes, we could follow the principle of GPS and obtain a node's position through trilateration or multilateration techniques. It shall be noted that, in order to apply GPS's principle for localization in wireless ad-hoc network, we shall have at least three non-collinear anchor nodes whose positions are available, and what is more important, each unknown node shall have ranging measurement to at least three non-collinear anchor nodes. While the first requirement can be easily met through deploying a few nodes equipped with advanced GPS receivers,

1.2 Design Challenges

it is generally difficult to obtain ranging measurements between each unknown node and those anchor nodes. Such limitation that nodes are only able to perform ranging measurement with their adjacent nodes is ascribed to properties of ranging techniques such as Ultra-Wide Band (UWB) Time-of-Arrival (TOA) [23], Received Signal Strength (RSS) [5], ultrasonic TOA [12] and hybrid approach [31] *etc.* For example, when node A is far away from node B, it is impossible for A to detect the signal transmitted from B and thus A is unable to estimate their range through RSS. In the past decade, researchers have formulated a series of localization algorithms under which nodes are able to cooperatively make use of each other's ranging measurement to exploit the position of all nodes in wireless ad-hoc network. Such kind of localization algorithms is called Cooperative Localization Algorithm (CLA), and its comparison with conventional triangulation method is depicted in Fig. 1.2.

Fig. 1.2: Comparison between triangulation method (left) and cooperative localization method (right): star represents anchor node with known position, whereas circle refer to unknown node whose position is to be estimated; although ranging measurements between blue unknown node and anchor nodes are unavailable, cooperative localization algorithm makes use of other pair's ranging measurement to “cooperatively” estimate blue node's position.

1.2 Design Challenges

The design challenges of CLA for wireless ad-hoc network are largely dependent on the characteristics of wireless ad-hoc network. For example, Akyildiz *et al.* [3] provide us a nice discussion on some important issues about WSN and sensor nodes, such as scalability issue and power consumption *etc.* During the design process of CLA, we

1.2 Design Challenges

must take the intrinsic characteristics of wireless ad-hoc network into consideration so that the algorithm's running will not encumber the normal operation of wireless ad-hoc network. Several challenging design requirements on CLA are summarized as following:

- **Computationally inexpensive.** Typically the speed of a PDA's microprocessor ranges from a few tens to several hundreds of MHz while the one of sensor nodes is much slower [1]. Although most Operating Systems (OS) of PDA and sensor nodes have been optimized to occupy little computational resources, the processing unit still has to cope with several tasks including interacting with peripherals, making routing decision and scheduling transmission *etc.* Given such stringent condition, CLA shall not be computationally expensive.
- **Efficient message exchange strategies.** The nature of CLA requires that each node send their ranging measurements to others through wireless link, *e.g.* exchanging ranging data with neighboring nodes or forwarding ranging data to a central processing station. According to [8], the energy consumed to transmit 1 bit of data is able to execute about 1,000-30,000 instructions, and thus the communication cost of CLA shall be carefully addressed. Apart from energy aspect, excessive communication of ranging data might also congest the network and hamper normal data transmission since they share the same radio hardware and wireless link.
- **Relative localization.** As mentioned earlier, it is generally infeasible to attach GPS receiver to every node, and the positioning results may not be accurate enough to be used as reference to determine a global coordinate system. Under such anchor-less circumstance, it will certainly give CLA an edge if it is able to find nodes' relative positions. For example in Fig. 1.3, without considering the global X - Y coordinate, node A still knows that node C is d_1 apart and has an azimuth of 30° with respect to node B ; similarly, node G is d_2 apart and has an azimuth of 120° but in an opposite direction. Besides, without absolute position information, relative position information is still sufficient to help wireless ad-hoc network (especially WSN) perform certain tasks, *e.g.* geographical routing [40] and node activity management [39] *etc.*

1.3 Thesis Contributions

- **Accuracy.** Owing to intrinsic error in ranging measurements, consequent position estimate of an unknown node would unavoidably have certain deviation from its true position. Sometimes the error of ranging measurement is so large that resulting map could even become completely different from its actual topology. Hence minimizing positioning error is the most fundamental job of CLA designers.

Fig. 1.3: Example of relative localization: when global X - Y coordinate is not obtainable, node A could make use of another node, say B , as reference direction (similar like North in navigation system) to determine a local coordinate.

1.3 Thesis Contributions

In this thesis, the key problems that we are interested in can be summarized as following: Given that each node in a wireless ad-hoc network has ranging measurements to its neighboring nodes and assuming that we are “standing” at an arbitrary node A , then how do we find the relative position of a remote node B with respect to a neighboring node of A without the aid of any explicit global coordinate system? And how to “efficiently” disseminate messages available at any node to the whole network, so that above localization procedure can be performed?

1.4 Thesis Outline

This thesis consists of two main parts, covering the two questions raised in Section 1.3, respectively. Question one is investigated from Chapter 2 to Chapter 4. In Chapter 2,

1.4 Thesis Outline

we illustrate three fundamental concepts behind our proposed cooperative localization methods, namely, Local Coordinate System, Multi-Dimensional Scaling plus Refinement for localization, and Vector Addition localization scheme. Relevant research work in literature is also introduced in this chapter. Chapter 3 and Chapter 4 discuss the two proposed “optimal path selection” mechanisms for Vector Addition localization scheme. We move on to present the second part of our work from Chapter 5, which includes relevant background information about fast message dissemination as well as the proposed method for collision-prohibited case. Chapter 6 is dedicated to our proposed method for fast message dissemination of collision-allowed case. This thesis is concluded in Chapter 7, together with some discussions on possible future extensions.

Chapter 2

Local Coordinate System and Vector-addition Localization Scheme

This chapter introduces various background knowledge and information related to our research work (especially to the first main part of our research work). These include the construction of Local Coordinate System, the review of Multi-Dimensional Scaling (MDS) technique, and the description of Vector-addition Localization Scheme.

2.1 Local Coordinate System

The desire to set up Local Coordinate System at each node comes from twofold: first, in order to make Cooperative Localization Algorithm (CLA) operate in a distributed fashion, we have to set up LCS at each node for expressing other nodes' position when Global Coordinate System (GCS) is not accessible; second, in the case of anchor-less wireless ad-hoc network, GCS does even not exist and thus we have to turn to LCS for locating other nodes relatively.

2.1.1 Construction of Local Coordinate System

Clearly, the node of interest (here we call it “central node”) shall sit at the origin of LCS which is normally an X-Y Cartesian coordinate system. Similar like Galton's coordinate system for describing human portrait [13], the central node selects one of

2.1 Local Coordinate System

its neighboring nodes to define the positive X-axis of its LCS and another neighboring node to define the positive Y-axis. From polar coordinate system's point of view, the neighboring node that defines positive Y-axis of Cartesian coordinate system is used to determine the positive direction of angle from polar axis. Similar approach for constructing LCS could also be found in [7].

Fig. 2.1: Setup of node A's LCS: node B is selected to determine positive X-axis and node D to determine positive Y-axis; the radius of this local map is 2-hop.

Owing to the presence of ranging errors, the two neighboring nodes that help define LCS shall be carefully selected so as to make a rigid coordinate system. We suggest the two adjacent nodes be selected in such a way that the resulting triangle is most approximate to equilateral triangle. In other words, the smallest angle in the resulting triangle should be the largest among all possible candidates. For example in Fig. 2.1, node A is central node, and node B and node D are selected to determine X-Y coordinate system since $\triangle ABD$ is the most approximate to equilateral triangle. Besides, the resulting triangle shall not be too small, and it is better not to select triangle whose side lengths are comparable to ranging error even though the triangle is the most approximate to equilateral. Certain threshold value, e.g. a few times of the standard deviation of ranging error, can be set to avoid small resulting triangle.

Each node in a wireless ad-hoc network has its own local map, which is described by its LCS, and the radius of local map can be simply defined by number of hops. The larger radius a local map has, the more information that it can retain from global map.

2.1 Local Coordinate System

2.1.2 Manipulations of Local Coordinate System

Prior to processing the data from different LCSs for cooperative localization, each LCS shall be manipulated in such a way that they all align with a common fixed coordinate system. Such manipulation consists of necessary *reflection* (or *mirroring*), *rotation* and *scaling* of LCSs. Scaling can be neglected if the scale is predefined at each node.

Between two LCSs whose central nodes are adjacent, whether one of them needs reflection for alignment can be determined by those common neighboring nodes of these two central nodes. Instead of the geometric method devised in [7], here we propose and illustrate a simple solution, based on cross-product of vectors, for determining whether reflection is needed. Take Fig. 2.2 for example. We want to know whether reflection is necessary between N_A and N_B 's LCSs. Node N_1 is one common neighbor of N_A and N_B . Our proposed solution states that:

If resulting vector of $\overrightarrow{N_A N_B} \times \overrightarrow{N_A N_1}$ in N_A 's LCS has the same direction as that of $\overrightarrow{N_B N_A} \times \overrightarrow{N_B N_1}$ in N_B 's LCS, reflection is needed between N_A and N_B 's LCSs; otherwise if opposite direction, no reflection is necessary.

Fig. 2.2: Reflection decision making: in the local map of node N_A , the result of $\overrightarrow{N_A N_B} \times \overrightarrow{N_A N_1}$ is negative whereas the result of $\overrightarrow{N_B N_A} \times \overrightarrow{N_B N_1}$ in node N_B 's local map is also negative; the same sign implies that these two maps' orientations are different and reflection is needed.

By applying the same procedure, each common neighbor of N_A and N_B can contribute one vote to determine whether reflection is necessary. Such voting mechanism could help mitigate the probability of wrong decision owing to ranging error, e.g. the case of N_3 . Although the two LCSs of N_A and N_B share some common ranging measurements, their local maps are not the same and hence resulting positions of N_3 in these two LCSs may not be consistent. Consequently, a wrong decision could be made, and hence voting mechanism is necessary to correct it.

2.2 Multi-Dimensional Scaling with Refinement

After necessary reflection is done, our next step is to rotate a node's LCS by some degrees so that it could completely align with one of its neighbors' LCSs. The computation of rotation angle follows simple geometric principle, as shown in Fig. 2.3. We define counter-clockwise as the positive direction of angle in an LCS. Then α is the angle from N_A 's positive X-axis to $\overrightarrow{N_A N_B}$, β is the angle from N_B 's positive X-axis to N_B 's positive X-axis (denoted by X'), and γ is the angle from N_B 's positive X-axis to N_A 's positive X-axis (i.e. the rotation angle for N_B 's LCS to align with N_A 's). Clearly, $\gamma = 180^\circ - \alpha - \beta$.

Fig. 2.3: Computing rotation angle (after necessary reflection)

2.2 Multi-Dimensional Scaling with Refinement

After the LCS at a particular central node is constructed, we can proceed to compute neighboring nodes' positions in that central node's local map. Capkun *et al.* [7] proposed a simple and efficient technique based on geometric principles for computing neighbors' positions. However, in term of accuracy, their geometric technique suffers from *error accumulation*, and not every ranging measurement has been utilized for computing positions. On the other hand, Shang *et al.* [37] made use of *Multi-Dimensional Scaling* (MDS) together with *Refinement* technique to process available ranging measurements in a holistic manner and generate more accurate position estimates. Besides, unlike conventional multilateration methods [27][33][34], MDS technique does not require the presence of anchor nodes. In this section, we are going to review such "MDS with Refinement" approach in details.

2.2 Multi-Dimensional Scaling with Refinement

Localization method through Refinement is inspired by the analogy between cooperative localization and mass-spring system [26][28]. As illustrated in Fig. 2.4, anchors can be viewed as masses (spool of thread) fixed on board, and ranging measurement between a pair of nodes is equal to the natural length of the spring connecting two masses. When this mass-spring system reaches equilibrium and total energy becomes minimal, the resulting masses' positions are our desired results. The basic task of Refinement-based methods is to iteratively refine each node's position until similar minimal-energy equilibrium point is reached. Note that such analogy can also be visualized without anchors: similar like the Local Coordinate System in section 2.1, the central node is a mass that is fixed on board, and the reference node determining positive X-axis is free to move along a fixed groove which passes through central node. After release, such a mass-spring system will also reach an equilibrium sooner or later.

Fig. 2.4: A mass-spring system: originally illustrated in [28]

2.2.1 Least Square Error Minimization

Least Square Error (LSE) minimization is one of the most common approaches for Refinement technique to define the equilibrium point. The objective function that is to be minimized is

$$\sum_{j=1}^k (\hat{d}_j - d_j)^2$$

where k is the number of ranging measurements, d_j is the j -th ranging measurement, and $\hat{d}_j$ is the corresponding distance after refinement. It is not difficult to find that

2.2 Multi-Dimensional Scaling with Refinement

LSE minimization is equivalent to energy minimization in above mass-spring system. The objective function of LSE minimization is nonlinear and solving such optimization problem is generally computationally expensive when the input size is large. LSE minimization is normally solved through numerical methods which start from an initial solution and run iteratively to find the minimum, e.g. the Levenberg-Marquardt algorithm [24].

Besides of its high requirement on computational resources, another shortcoming of LSE minimization is the possibility that numerical methods may reach local minima. In this case, the resulting network map might be completely different from its actual format[26]. In order to avoid this pitfall, the starting point that numerical methods take as the input shall be close to correct solution as much as possible, e.g. using MDS technique to find initial estimates. In spite of above drawbacks, LSE minimization is widely studied in literature on localization thanks to its high accuracy. According to classical estimation theory [21], LSE minimization is a Minimum Variance Unbiased (MVU) estimator when the noise (or ranging error here) is Gaussian. Compared to refinement-based methods like LSE minimization, conventional multilateration-based methods or geometric methods are inferior because not all information (i.e. ranging measurements) are utilized in estimating nodes' positions, and hence the accuracy will be intrinsically worse than that of LSE minimization.

In the following subsection, we investigate MDS technique, which provides accurate initial position estimates for LSE minimization process. Some other techniques, like the heuristics in [26], are generally not as good as MDS and thus are not discussed here.

2.2.2 Multidimensional Scaling for Position Estimation

MDS technique was originally studied in statistics and used to visualize items' configuration based on their similarities matrix. By substituting similarities matrix by nodes' distance matrix D , MDS technique could help us find a network map with nodes' relative positions embedded [37]. Distance matrix D is symmetric and can be constructed from the distance estimates of each pair of nodes in wireless ad-hoc network. The output of MDS is an $n \times p$ configuration matrix W , where $p < n$ is determined by the eigenvalues of the matrix product between D and its transpose. This process can be described by

2.2 Multi-Dimensional Scaling with Refinement

$$D = \begin{pmatrix} 0 & & & & \\ d_{21} & 0 & & & \\ d_{31} & d_{32} & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ d_{n1} & d_{n2} & d_{n3} & \cdots & 0 \end{pmatrix} \xrightarrow{MDS} W = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1p} \\ w_{21} & w_{22} & \cdots & w_{2p} \\ w_{31} & w_{32} & \cdots & w_{3p} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{np} \end{pmatrix}.$$

Configuration matrix W contains position information of all n nodes, but in p dimension. Nevertheless, the first two or three columns of W could still give us a nice approximation of nodes' position estimates in 2-dimension or 3-dimension. It shall be noted that only relative position information is contained in configuration matrix W , and the resulting network map has to undergo necessary reflection, rotation, translation and scaling so as to fit absolute coordinate system. For example, if there are three non-collinear anchors in the network, we shall manipulated (*i.e.* reflect, rotate, translate and scale) W in such a way that the summed LSE of all three anchors is minimized.

Now the problem is left to how to find the distance estimates of each node pair for constructing distance matrix D . The simplest approach is to compute the shortest-distance path between each pair of nodes using Dijkstra's Algorithm. This approach does not perform well for irregular network topology because a "curved" shortest-distance path would over-estimate the true Euclidean distance between two nodes. This is an intrinsic drawback of "MDS plus refinement" approach.

Another drawback is the time complexity. The time complexities of finding distance matrix D and running MDS are both $O(n^3)$, where n is the number of input nodes. It is difficult to formulate the time complexity of subsequent LSE minimization process, but according to [37], LSE minimization takes much longer time than finding D and running MDS, especially when the input size is large.

2.2.3 Variants of "MDS with Refinement" Approach

Owing to the high computational cost of LSE minimization, it is generally impractical to take all ranging measurements as input and refine the whole map. In order to alleviate this problem, Shang *et al.* [36] proposed a distributed approach as a variant of the original "MDS with Refinement" approach, called MDS-MAP(P), which refines only local maps and then stitches every local map together to form global map.

2.2 Multi-Dimensional Scaling with Refinement

In details, the first step is to define the radius of local maps in term of number of hops. Intuitively, the larger radius each local map has, the more close result to that of global map refinement. It has been shown in [36] that radius larger than 2-hop would not give considerable improvement. In the second step, we run standard “MDS with Refinement” procedure to find nodes’ relative positions in each local map. The third step is to randomly choose a node’s local map as core and then to grow the core map through map merging. The local map which has largest number of common nodes with core map is selected to merge with core map. The merging process is based on LSE minimization on the positions of common nodes between core map and selected local map.

This MDS-MAP(P) method does not only fit well in wireless ad-hoc network by running in a distributed fashion, but also performs much better than original “MDS with Refinement” approach in irregular network. That is mainly attributed to the fact that MDS-MAP(P) does not take as input those shortest-distance paths which are generally longer than actual Euclidian distances in irregular network topology.

Following similar strategy of MDS-MAP(P), Shang *et al.* [36] proposed MDS-MAP(R) to locate remote nodes’ relative positions in wireless ad-hoc network through map merging along shortest path. In details, if node A wants to get the relative position of a remote node B in A ’s local coordinate system, it will first find a shortest path (in term of number of hops) between A and B , then compute each local map through “MDS plus refinement” procedure along that path, and finally merge each local map (through LSE minimization) to get relative position of B . Because MDS-MAP(R) is distributed and able to find remote nodes’ relative positions in a local coordinate system, it meets the requirements of localization problem in wireless ad-hoc network, as discussed in section 1.2. Later in this thesis, we will use MDS-MAP(R) as a benchmark for our proposed solution to compare with.

Another notable variation of “MDS with Refinement” approach is the work present in [2]. The authors take a hybrid approach of “MDS with Refinement” method and multilateration-based method. At the beginning, a set of nodes are selected as reference nodes (or anchors). Such set of reference nodes can be randomly selected, or only nodes that lie on the perimeter are selected based on the method in [26]. Then “MDS plus refinement” method is utilized to compute the positions of those reference nodes.

2.3 Vector-addition Localization Scheme

Thanks to its small input size, this step is computationally much less expensive than refining the whole map.

2.3 Vector-addition Localization Scheme

The basic idea of our proposed cooperative localization strategy evolves from previously-mentioned MDS-MAP(R) method which suits wireless ad-hoc network very well. When locating a remote node in wireless ad-hoc network, we would like to enjoy the high accuracy of “MDS with Refinement” method and at the same time circumscribe its high communication cost and computational cost. Therefore, instead of using expensive LSE minimization to merge a series of local map along shortest path, we adopt the simple *Vector Addition* approach which was first suggested in [7] and focus our research work on how to select *the optimal path* for locating remote nodes using Vector Addition.

Fig. 2.5: Locating remote node through vector addition: each node along the path knows its neighboring nodes' positions (or bound vectors) in its own local coordinate system; assuming all coordinate systems are properly aligned with that of N_0 , N_5 's position in N_0 's coordinate system could be easily found by $\overrightarrow{N_0N_5} = \overrightarrow{N_0N_1} + \overrightarrow{N_1N_2} + \overrightarrow{N_2N_3} + \overrightarrow{N_3N_4} + \overrightarrow{N_4N_5}$.

As a vector $\overrightarrow{AB}$ can be represented by an arrow in Cartesian space, drawn from an initial point A to a terminal point B , we can make use of bound vector, whose initial

2.3 Vector-addition Localization Scheme

point is at origin, to describe position information in a Cartesian coordinate system. For example, if we want to locate a node N_k in a coordinate system whose origin is node N_0 , we simply need find bound vector $\overrightarrow{N_0N_k}$ in that coordinate system. Suppose there is a path from N_0 to N_k , and each node along this path has its own coordinate system to express its neighboring nodes' position. If all coordinate systems along the path are properly aligned (after necessary reflection, rotation and scaling), i.e. the same X and Y directions as N_0 's, N_0 is able to know N_k 's position by adding all vectors along the path. This process is exemplified in Fig 2.5.

Here we describe detailed procedure of our proposed Vector-addition Localization Scheme for wireless ad-hoc network:

1. Each node obtains ranging measurements between itself and its neighboring nodes through some ranging techniques. We assume that two adjacent nodes share the same ranging measurement, and ranging error follows Gaussian distribution with zero mean.
2. Each node sends its ranging measurements to all nodes within 2-hop or 3-hop range.
3. After obtaining all ranging measurements in its local map, each node starts to build its Local Coordinate System and run standard "MDS with Refinement" procedure to compute position estimates in its local map.
4. Each node disseminates the position estimates information of its local map to the whole wireless ad-hoc network. We are interested in how to quickly disseminate such information to the whole network (see Chapter 5 and Chapter 6).
5. After knowing the position estimates information of all local maps, each node is able to utilize Vector Addition, with necessary LCS manipulation, to find the relative position of any remote node. We are interested in looking for the optimal Vector Addition path with minimal localization error (see Chapter 3 and Chapter 4).

Chapter 3

A Heuristics for Vector-addition Path Selection

In this chapter, we describe the parameter of “Confidence Index” for each vector, which can be used as the metric for selecting optimal vector-addition path.

3.1 Confidence Index of Vector

The set of system equations of localization problem is normally overdetermined, *i.e.* the number of equations is larger than that of variables, especially when network connectivity is high. Refinement procedures, such as LSE minimization, make use of system equations’ over-determinacy to alleviate the effect of ranging error and generate accurate localization results. In this sense, regarding the path selection problem for vector addition, we shall choose a vector that has higher level of over-determinacy (or redundancy).

The four geometrics in Fig. 3.1 help illustrate the concept of over-determinacy. In (a), the simple triangle geometrics is not overdetermined. With the introduction of additional two ranging measurements AD and BD in (b), the level of over-determinacy does not change since at the same time two new variables (node D ’s coordinate in 2-Dimension) are also introduced. However, when ranging measurement of CD is available, as shown in (c), the network becomes overdetermined and such redundancy can be used to improve localization accuracy. Similarly in (d), node E brings “one redundancy” by introducing three new ranging measurements. For the over-determinacy

3.1 Confidence Index of Vector

Fig. 3.1: Example of vector's over-determinacy: vector $\overrightarrow{AB}$ has different levels of over-determinacy (or redundancy) in these four scenarios.

of each vector, we make use of a parameter called *Confidence Index* (CI) to describe its redundancy level. CI is defined as the number of ranging measurements among those nodes who have ranging measurements to the two nodes of that vector. The higher a vector's CI is, the closer that its refined length is to actual value. Take Fig. 3.1(d) for example. CI of vector $\overrightarrow{AD}$ is 1 since there is one ranging measurement between B and C; CI of $\overrightarrow{CD}$ is 2 since there are two ranging measurements among A, B and E who have ranging measurements to both C and D.

Fig. 3.2: Impact of bearing errors on end-to-end distance estimate: both polylines contain zero-mean bearing error with standard deviation equal to 5 degrees; apparently, the straight polyline of (a) has less error than the curved polyline of (b) in term of the end-to-end distance estimate between initial node and ending node.

Intuitively, CI is able to describe the estimation accuracy of a vector's length but fails to judge the estimation accuracy of the angle between two contiguous vectors. Thus we are only interested in how much CI could help improve end-to-end distance estimate between initial node and a remote node rather than the position estimate of a

3.2 Performance Evaluation and Analysis

remote node. With CI values of each vector available, our next step is to select optimal path from all possible shortest paths (in term of number of hops). There are two main reasons why we should use shortest paths as candidates for CI-based path selection. One obvious reason is because smaller number of hops would have less accumulation of errors. Another is because shortest path intends to be a straight polyline whose end-to-end distance is less sensitive to the errors of angle (or bearing) between two consecutive segments, as shown in Fig. 3.2.

Based on CI values of each vector, we propose two types of optimal shortest path (in term of the number of hops): one is the shortest path with largest sum of CI (*Heaviest Path*) and another one is the shortest path that has the largest minimum CI (*Stablest Path*). In the following section, we investigate the performance of our proposed heuristics on end-to-end distance estimate.

3.2 Performance Evaluation and Analysis

In order to evaluate end-to-end distance estimate, we define a parameter called Pair Distance Error (PDE) as

$$PDE = \frac{\hat{d} - d}{\sigma}$$

where $\hat{d}$ is the estimated distance, d is the actual distance and σ is the standard deviation of ranging error (zero mean is assumed). We perform simulation exercises in Matlab (version 7.1). Two types of network topologies are simulated: regular square-shape map and irregular C-shape map [37], as shown in Fig. 3.3. In the simulation of square-shape map, 150 nodes are randomly deployed in a 115×115 area. For C-shape map, we divide the 115×115 area into 3×3 grids, and randomly deploy 140 nodes into the area excluding the central grid and middle-right grid. Nodes that are within 20 are assigned a ranging measurement, as indicated by edges in the graphs. It makes sense since the energy of probes (either radio or ultrasound) generally degrades with respect to distance, especially in free space. The average degrees in these two maps are 11.4 and 12.1, respectively. Following the ranging error characteristics analyzed in [38], the ranging error used in our simulation is characterized as zero-mean normal-distributed. Thus ranging error is solely determined by its standard deviation. Here we use standard deviation of 1, which is 5% of the communication range of 20. As a lower benchmark for comparison, we investigate the PDE performance of *Lightest Path*, which has the

3.2 Performance Evaluation and Analysis

smallest sum of CI among all shortest-paths. We also harness “MDS-plus-refinement” method to process the network map as a whole, and take the results as the “optimal” benchmark for comparison.

Fig. 3.3: Network maps used in simulation: (a) square-shape map and (b) C-shape map.

Fig. 3.4: Selected PDE results of square-shape map: the sample sizes are 1450, 819, 278, and 74, respectively.

The results (mean and standard deviation) of square-shape and C-shape graphs are shown in Fig. 3.4 and Fig. 3.5, respectively. We group the PDE results of all pair of nodes that are the same number of hops apart. The largest number of hops in square-shape map is 9, whereas the one in C-shape map is 12. For square graph, it is clear that Heaviest Path and Stablest Path both have less standard deviation of

3.2 Performance Evaluation and Analysis

Fig. 3.5: Selected PDE results of C-shape map: the sample sizes are 388, 294, 135, and 81, respectively.

PDE than Lightest Path, and Heaviest Path is slightly better than Stablest Path. In addition, as the number of hops increases, the superiority of Heaviest Path (or Stablest Path) over Lightest Path becomes more obvious. That is because, when the number of hops is large, the actual path by Heaviest Path (or Stablest Path) deviates to a larger extent from the one by Lightest Path, e.g. probably not only edge-disjoint but also even node-disjoint.

For the same number of hops (e.g. 9-hop), the results of C-shape graph are much worse than those of square-shape (standard deviation of PDE in 9-hop case is about 2 in square-shape graph whereas about 5 in C-shape one). This can be explained by the fact that the overall structure of square-shape map is more rigid. Among C-shape graph results, it's found that Heaviest Path and Stablest Path cannot provide much improvement, especially when the number of hops is large (e.g. 11-hop and 12-hop). The main reason is because shortest-paths of a long route in C-shape graph, *i.e.* between a top-right node and a bottom-right node, intend to follow along the inner circle of the C-shape graph. Such inner-circle paths normally have small CI, since they lie along the fringe of map. Besides, inner-circle paths have few alternatives and the three types of shortest-paths (Heaviest Path, Stablest Path, and Light Path) might be the same, hence their results are similar.

It shall be noted that, although both Heaviest Path and Stables Path give us better accuracy than Lightest Path does, the advantage is not very significant, especially in C-shape graph. This is partially attributed to the heuristics nature of our path selection method. In the following chapter, we will introduce a new path selection method that relies on rigorous analysis.

Chapter 4

CRLB-based Path Selection Algorithm

In this chapter, we propose and investigate another path selection method for Vector Addition, based on Cramer-Rao Lower Bound (CRLB) and some geometric principles.

4.1 CRLB in Anchored Network

CRLB is a lower bound on the minimum variance that an unbiased estimator can achieve [21], and it provides a performance benchmark for unbiased estimator to compare with. Applied to localization problem in wireless ad-hoc network, CRLB gives us the best accuracy that an unbiased position estimator is able to obtain. The CRLB of unknown parameter θ which is to be estimated is closely related to its measured value's Probability Density Function (PDF) $p(x; \theta)$, where x is random measurements. Intuitively, it makes sense since the smaller range those measured value disperses around actual value, the more accurate estimate we can obtain.

How much measured value will disperse around actual value could be described by the “sharpness” (or curvature) of measured value's PDF. PDF's curvature is quantified by the negative average of the second derivative of the natural logarithm of PDF on all possible measured value. Denote an unbiased estimator of θ by $\hat{\theta}$. Its variance will satisfy

$$\text{var}(\hat{\theta}) \geq \frac{1}{-E \left[\frac{\partial^2 \ln p(x; \theta)}{\partial \theta^2} \right]}$$

4.1 CRLB in Anchored Network

and this lower bound is CRLB.

Derivation of CRLB could be extended to a series of unknown parameters, i.e. a vector parameter $\theta = [\theta_1, \theta_2, \dots, \theta_p]$, provided that θ 's PDF is available. CRLB of corresponding estimator $\hat{\theta}$ is directly computed from Fisher Information Matrix (FIM), $I(\theta)$, whose dimension is $p \times p$ and which is defined as

$$[I(\theta)]_{ij} = -E \left[\frac{\partial^2 \ln p(x; \theta)}{\partial \theta_i \partial \theta_j} \right]$$

where $i, j \in [1, p]$. Then variance of the estimator of an element $\hat{\theta}_i$ satisfies

$$\text{var}(\hat{\theta}_i) \geq [I^{-1}(\theta)]_{ij}.$$

For the problem of localization in wireless ad-hoc network, nodes' coordinates $\theta = [x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n]$ (n is the number of nodes with unknown positions) are the parameters that are to be estimated, and we need to derive the PDF of θ 's measured values from the PDF of ranging measurement so that we are then able to find CRLB of position estimates. Patwari *et al.* [29] provide us a nice analysis on how to compute the PDF of position estimates as well as corresponding Fisher Information Matrix (FIM) F in a network with at least three anchors. Here we would like to iterate their analysis and results.

For coordinates parameter $\theta = [x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n]$, its FIM F 's dimension is $2n \times 2n$, and F can be further divided into three $n \times n$ submatrices F_{xx} , F_{xy} and F_{yy} , as shown below

$$F = \begin{pmatrix} F_{xx} & F_{xy} \\ F_{xy}^T & F_{yy} \end{pmatrix} = \begin{pmatrix} f_{x_1 x_1} & \cdots & f_{x_1 x_n} & f_{x_1 y_1} & \cdots & f_{x_1 y_n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ f_{x_n x_1} & \cdots & f_{x_n x_n} & f_{x_n y_1} & \cdots & f_{x_n y_n} \\ f_{y_1 x_1} & \cdots & f_{y_1 x_n} & f_{y_1 y_1} & \cdots & f_{y_1 y_n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ f_{y_n x_1} & \cdots & f_{y_n x_n} & f_{y_n y_1} & \cdots & f_{y_n y_n} \end{pmatrix}_{2n \times 2n}.$$

Assume that ranging errors follow zero-mean normal distribution as suggested

4.1 CRLB in Anchored Network

in [38]. Then these three submatrices can be computed as

$$\begin{aligned}
 [F_{xx}]_{k,l} &= \begin{cases} \frac{1}{\sigma^2} \cdot \sum_{i \in N(k)} \frac{(x_k - x_i)^2}{d_{ki}^2}, & k = l \\ -\frac{1}{\sigma^2} \cdot I_{N(k)}(l) \cdot \frac{(x_k - x_l)^2}{d_{kl}^2}, & k \neq l \end{cases} \\
 [F_{xy}]_{k,l} &= \begin{cases} \frac{1}{\sigma^2} \cdot \sum_{i \in N(k)} \frac{(x_k - x_i)(y_k - y_i)}{d_{ki}^2}, & k = l \\ -\frac{1}{\sigma^2} \cdot I_{N(k)}(l) \cdot \frac{(x_k - x_l)(y_k - y_l)}{d_{kl}^2}, & k \neq l \end{cases} \\
 [F_{yy}]_{k,l} &= \begin{cases} \frac{1}{\sigma^2} \cdot \sum_{i \in N(k)} \frac{(y_k - y_i)^2}{d_{ki}^2}, & k = l \\ -\frac{1}{\sigma^2} \cdot I_{N(k)}(l) \cdot \frac{(y_k - y_l)^2}{d_{kl}^2}, & k \neq l \end{cases}
 \end{aligned} \tag{4.1}$$

where σ^2 is variance of ranging errors, $N(k)$ is the set of node k 's neighbors, (x_i, y_i) is actual position of node i , d_{kl} is actual distance between node k and node l , and $I_{N(k)}(l)$ is equal to zero if $l \in N(k)$ otherwise zero.

Fig. 4.1: Visualization of CRLB in anchored network: N_0 , N_1 , and N_2 are anchors; transmission range is 20, and standard deviation of ranging error is 1; CRLB on unknown nodes' position estimates are described through 2- σ uncertainty ellipse. (A comprehensive Graphic User Interface to visualize CRLB could be found in [4].)

4.2 CRLB in Anchor-less Network

With FIM F available, CRLB can be easily obtained by taking F 's inverse matrix. The $2n$ diagonal elements of F^{-1} represent CRLB on these $2n$ coordinates parameters, respectively. Since we focus our work on 2-D case, we could visualize CRLB through $2\text{-}\sigma$ uncertainty ellipse that is widely used in the study of GPS. Fig. 4.1 illustrates an example on the visualization of CRLB in a network with three anchors. Since we assume ranging errors are Gaussian distributed, nodes' positions can be estimated by an unbiased estimator, i.e. each ellipse is centered at its corresponding true position. Clearly, CRLB of N_3 's position estimate variance is much smaller than that of N_6 . It intuitively makes sense since N_3 has direct ranging measurements with three anchors, while N_6 is only connected with unknown nodes (including N_3 itself). In other words, CRLB helps us gauge the "trustfulness" of each node's position estimation. Recall that such "trustfulness" could assist in making path selection decision in vector addition approach for locating remote nodes. That is the reason why we harness CRLB for designing our second proposed path selection method. However, before embarking to employ CRLB in choosing optimal path, we have to revise the procedure for computing CRLB in order to find CRLB in an anchor-less network or in the local map of an unknown node.

4.2 CRLB in Anchor-less Network

Before utilizing CRLB in making path selection decision, we shall find a way to compute the CRLB of each 1-hop neighbor's position estimate, described in an unknown node's local map. Unlike the scenario considered in [29], a node's local map does not contain any fixed anchor nodes, as each local map covers only 2 or 3 hops of nodes. Nevertheless, we can still make use of some fixed "coordinates parameters" to compute CRLB even though fixed anchor nodes do not exist.

Recall the LCS setup procedure discussed in section 2.1 as well as the LCS example illustrated in Fig. 2.1. It is not difficult to find that, among all coordinates in a LCS, there are three fixed coordinates: X-coordinate and Y-coordinate of the central node (the origin), and Y-coordinate of the node that determines positive X-axis of LCS. These three coordinate values are all fixed at zero, and the other coordinates are considered as normal parameters that are to be estimated. Suppose there are $(n+1)$ nodes in a LCS, and let N_0 denote the central node and N_1 the node determining positive

4.3 Path Selection based on CRLB

X-axis. Then there are totally $(2n - 1)$ coordinates parameters to be estimated, and rest three coordinates are fixed at zero:

$$x_0 = y_0 = y_1 = 0.$$

Following the structure of FIM discussed in previous section, we can easily deduce the structure of FIM in an anchor-less network (or in an LCS):

$$F = \begin{pmatrix} f_{x_1x_1} & f_{x_1x_2} & \cdots & f_{x_1x_n} & f_{x_1y_2} & \cdots & f_{x_1y_n} \\ f_{x_2x_1} & f_{x_2x_2} & \cdots & f_{x_2x_n} & f_{x_2y_2} & \cdots & f_{x_2y_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ f_{x_nx_1} & f_{x_nx_2} & \cdots & f_{x_nx_n} & f_{x_ny_2} & \cdots & f_{x_ny_n} \\ f_{y_2x_1} & f_{y_2x_2} & \cdots & f_{y_2x_n} & f_{y_2y_2} & \cdots & f_{y_2y_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ f_{y_nx_1} & f_{y_nx_2} & \cdots & f_{y_nx_n} & f_{y_ny_2} & \cdots & f_{y_ny_n} \end{pmatrix}_{(2n-1) \times (2n-1)}.$$

The formulas 4.1 used for computing each element of FIM in anchored network are still applicable for computing FIM's element in anchor-less network. Thus derivation of CRLB in an unknown node's local map is then straightforward. Similar with anchored network, CRLB in node's local map could also be visualized, as shown in Fig. 4.2. It shall be noted that, since Y-coordinate of N_1 is fixed at zero, N_1 is gliding along X-axis and its CRLB is a line segment instead of an ellipse. Compared with the CRLB of anchored network in Fig. 4.1, CRLB in anchor-less network is bigger as a result of larger number of parameters.

4.3 Path Selection based on CRLB

Recall that CRLB gives the performance lower bound on estimation variance of any unbiased estimator. Given a particular unbiased estimator, e.g. LSE minimization method, CRLB is also efficacious in telling which parameter has the smallest estimation variance among all parameters. In order words, the variance bound given by CRLB about each parameter could be used as metric to select the optimal path which has minimal total variance (sum of variance in X-axis and Y-axis). Here in this section, we would like to first illustrate how to predict the variance bound of a remote node's position estimate based on CRLB of each local map along a predefined path.

4.3 Path Selection based on CRLB

Fig. 4.2: Visualization of CRLB in N_0 's local map: N_1 's CRLB is a line segment rather than ellipse.

If assuming that each LCS along a path is aligned and fixed with a global coordinate system or with initial node N_0 's LCS, the variance of resulting summed vector can be expressed as

$$\begin{aligned} \text{var}(\overrightarrow{N_0 N_m}) &= \text{var}\left(\sum_{i=0}^{m-1} \overrightarrow{N_i N_{i+1}}\right) \\ &= \sum_{i=0}^{m-1} \text{var}(\overrightarrow{N_i N_{i+1}}) + 2 \sum_{i,j:i < j} \text{cov}(\overrightarrow{N_i N_{i+1}}, \overrightarrow{N_j N_{j+1}}). \end{aligned}$$

It shall be noted that the bound of each term in above formula can be found through the inverse of FIM, i.e. CRLB, and thus variance bound of remote node's position estimate can be predicted by this formula. However, it is impossible to achieve the assumption that all LCSs along a path are aligned and fixed. That is because LCS is defined by neighboring node whose position is unknown. Thus some additional factors have to be taken into consideration when predicting variance bound of remote node's position estimate.

When calculating CRLB in the local map of node N_j , we shall select N_{j-1} , which is the last hop node to initial node N_0 , as the positive X-axis. That is because a node may

4.3 Path Selection based on CRLB

Fig. 4.3: Selection of positive X-axis of LCS and rotation of LCS (after necessary reflection) when adding vectors' variance: the bound ellipses are computed in the local map of last hop node and rotated to fit global coordinate; CRLB after coordinate rotation can be calculated according to [21].

have different LCSs defined by different X-axis and the value of CRLB depends on the LCS where it is computed. Thus along a path from initial node to a remote node, we shall select last-hop node as X-axis to define LCS so that it makes sense to sum CRLB in two contiguous local maps. For example in Fig. 4.3, N_1 uses N_0 to define its LCS and compute the uncertainty ellipse of N_2 ; similarly, N_2 uses N_1 to define its LCS in which N_3 's uncertainty ellipse is computed. Besides, similar with when adding vectors in Fig. 4.3, each local map shall be aligned to the coordinate system of N_0 or to a global map when adding the variance of each vector. Although the sum of vectors' variance in X-axis and Y-axis directions doesn't change with respect to rotation of coordinate [20], alignment of LCSs with a common one can help decompose the variance of resulting vector, *i.e.* being able to know the variance's components in X-axis and Y-axis of a common coordinate. In Fig. 4.3, the CRLB values of N_2 and N_3 are transformed to fit the LCS of N_0 (N_1 's CRLB is already computed in N_0 's LCS). It shall also be noted that there are also many covariance terms in calculating the estimate variance of resulting vector, and most of them can be omitted if the two intermediate vectors are

4.3 Path Selection based on CRLB

too many hops away, *i.e.* little correlation between them. As long as two vectors are within the local map of some node N_j , the bound of the covariance between them can be found in the CRLB derived at node N_j 's local map.

Fig. 4.4: Impact of X-axis's vibration on variance of remote node's position estimate: α is standard deviation of $\overrightarrow{N_1 N_0}$'s vibration angle in N_0 's local map (which is the global one); β is standard deviation of $\overrightarrow{N_2 N_1}$'s vibration angle in N_1 's local map; γ is standard deviation of $\overrightarrow{N_3 N_2}$'s vibration angle in global map; the dash lines are Global Reference for the convenience to express these vibration angles.

Another interesting problem arising when we are predicting the variance of a remote node is the “vibration of local map's X-axis”, which generally increases the resulting variance. As illustrated in Fig. 4.4, the CRLB uncertainty ellipses of N_1 , N_2 , and N_3 are computed in the local maps of N_0 , N_1 and N_2 respectively. N_0 's local map uses its own LCS as the global coordinate system, but N_1 uses $\overrightarrow{N_1 N_0}$ and N_2 uses $\overrightarrow{N_2 N_1}$ as positive X-axis for their local maps. N_1 's variance in global map doesn't suffer from vibration of X-axis since N_0 's LCS is global and fixed. However, when we use N_1 's local map to find N_2 's variance bound, we have to take the vibration of $\overrightarrow{N_1 N_0}$, which is the positive X-axis, into consideration. The variance of $\overrightarrow{N_1 N_0}$'s vibration angle, as indicated by α in Fig. 4.4, depends on the mean value of $|\overrightarrow{N_1 N_0}|$ and the component of N_1 's variance in the direction perpendicular to $\overrightarrow{N_1 N_0}$. Let $\epsilon_{N_1, \perp \overrightarrow{N_1 N_0}}$ denote the

4.3 Path Selection based on CRLB

projection of N_1 's position estimate on the direction perpendicular to $\overrightarrow{N_1 N_0}$. Then α can be approximately expressed by

$$\alpha = \tan^{-1} \left(\frac{\epsilon_{N_1, \perp \overrightarrow{N_1 N_0}}}{E \left[\left| \overrightarrow{N_1 N_0} \right| \right]} \right).$$

Variance of α then can be found by *delta method*, which uses second-order Taylor expansions to approximate the variance of a function, and delta method is given by

$$\text{var}[f(x)] \approx [f'(E[x])]^2 \cdot \text{var}(x).$$

Then the variance of α is computed by

$$\begin{aligned} \text{var}(\alpha) &= \left(\frac{d}{d\epsilon_{N_1, \perp \overrightarrow{N_1 N_0}}} \left(\tan^{-1} \left(\frac{\epsilon_{N_1, \perp \overrightarrow{N_1 N_0}}}{E \left[\left| \overrightarrow{N_1 N_0} \right| \right]} \right) \right) \right) \bigg|_{\epsilon_{N_1, \perp \overrightarrow{N_1 N_0}}=0}^2 \cdot \text{var}(\epsilon_{N_1, \perp \overrightarrow{N_1 N_0}}) \\ &= \frac{1}{E^2 \left[\left| \overrightarrow{N_1 N_0} \right| \right]} \cdot \text{var}(\epsilon_{N_1, \perp \overrightarrow{N_1 N_0}}). \end{aligned}$$

With the variance of α available, the contribution of $\overrightarrow{N_1 N_0}$'s vibration on the increased variance of N_2 in global map is then equal to $E^2 \left[\left| \overrightarrow{N_1 N_0} \right| \right] \cdot \text{var}(\alpha)$. This kind of additional increase on N_2 's variance is graphically represented by the bold line near N_2' in Fig. 4.4. Similarly, when we use N_2 's local map to calculate N_3 's variance bound, vibration of $\overrightarrow{N_2 N_1}$ with respect to global coordinate also affects N_3 's variance bound in global map. The variance of $\overrightarrow{N_2 N_1}$'s vibration angle, γ , can be computed by

$$\text{var}(\gamma) = \text{var}(\alpha) + \text{var}(\beta),$$

assuming α and β are uncorrelated. In Fig. 4.4, the increased variance of N_3 in global map due to vibration of $\overrightarrow{N_2 N_1}$ is depicted as the bold line near N_3' .

Heretofore we have discussed the two main contributing factors of the variance of remote node: one is due to ranging error in a local map and another is due to vibration of X-axis. The variance bound on both of these two factors are closely related to CRLB in each LCSs. We could sum these two contributions up and approximately predict the resulting variance bound of a remote node along a certain path. Our objective is to find a path such that the resulting variance bound is minimal. This problem is similar with the routing problem in wireless ad-hoc network which could be solved by Dijkstra's

4.4 Performance Evaluation and Analysis

algorithm. Recall that there are some covariances terms that indeed contribute to the resulting variance. Thus in this case, we could not directly make use of Dijkstra’s algorithm to find the optimal path with minimal sum of variance. Instead, we could first find a few (e.g. 3-5) optimal paths [30] without considering covariance, and then from these paths we pick up the one with least resulting variance bound after adding covariance between two adjacent vectors. Other covariance terms are omitted owing to their small values.

4.4 Performance Evaluation and Analysis

4.4.1 CRLB in Local Maps with Different Map Sizes

Recall that it is computationally intensive to refine a map by LSE minimization with high node density and large input size. Therefore, we use the number of hops from a local node N_0 as the radius to define a local map so that LSE minimization is with moderate computational cost. Since we are interested in only the relative positions of N_0 ’s 1-hop neighbors, we would like to know how local map’s radius affect the variance of 1-hop neighbors’ position estimate. In this subsection, we analyze CRLB of 1-hop neighbors with different map sizes and investigate how the LSE minimization method performs accordingly.

The analysis is simulated in Matlab (version 7.1) and a large number of nodes are uniformly distributed in a square area. In order to obtain a fairly uniform distribution, we first generate a very dense network, and then gradually dilute the network by taking out those nodes that are too close to their 1-hop neighbors. The relationship between CRLB of 1-hop neighbors and the size of local map is plotted in Fig. 4.5. 1-hop neighbors’ CRLB when radius is 7-hop is used as a benchmark to evaluate 1-hop neighbors’ CRLB at smaller map sizes. It is clear that local map coverage of 2-hop greatly improves the CRLB of 1-hop neighbors’ position estimates compared to 1-hop coverage case. Larger than 2-hop radius does not provide significant improvement as compared to 2-hop radius case, since there is already no much space left for improvement. (We therefore use local map with 2-hop coverage in the analysis in the following section.) When the average connectivity of the network varies, the CRLB of 1-hop neighbors in dense network intends to converge to optimal bound a bit faster than that in sparse

4.4 Performance Evaluation and Analysis

network. This is because the geometric shape of local map in a dense network is more rigid than that in a sparse one.

Fig. 4.5: Average CRLB of 1-hop neighbors in local maps with different sizes: the horizontal axis is the number of hops from a local node (local map radius), whereas vertical axis represents the average CRLB of 1-hop neighbors, taking 1-hop neighbor's bound in a 7-hop-large map as benchmark bound; the ranging error here is Gaussian with standard deviation equal to 5% of transmission range.

4.4.2 Optimal Path Selection for Locating Remote Nodes

Similar like the performance analysis section presented in section 3.2, here two topologies are investigated to evaluate our CRLB-based path selection approach: square-shape map (quasi-straight optimal path) and U-shape map (curved optimal path, similar with C-shape map). Fig. 4.6 shows the samples of square-shape network map and U-shape map. We plot paths that are set up from different criteria (CRLB-based method, heaviest CI, and lightest CI) in order to have a good visual comparison. Local maps with 2-hop radius and nodes' exact positions are used to predict resulting variance bound and find the optimal path. Clearly, in the squared-shape topology, there is certain overlap between path from CRLB-based method and the path with heaviest CI. The reason is because the path from CRLB-based method and path with heaviest CI both have the tendency to walk through area with high node density. In U-shape topology, there is also certain overlap, but between paths with heaviest CI and lightest CI. The reason is because paths obtained according to CI are all restricted to the shortest paths from local node to remote node. There are few choices of shortest paths available in U-shape topology since they all intend to follow the inner-circle. However, CRLB-based method

4.4 Performance Evaluation and Analysis

doesn't suffer from the shortest-path constrain, and the derived path still intends to go through dense area where relative position estimates are more accurate (because of higher CRLB).

Fig. 4.6: Path setup in square-shape and U-shape topologies: the nodes encircled by square and triangle are initial node and remote node respectively; the arrow is the global X-axis used at the local node; solid line is the path from the CRLB-based method, dash line is the path with heaviest Confidence Index (CI), and dot line is the path with lightest CI; transmission range is 20, and ranging error is Gaussian with zero mean and standard deviation equal to 1.

4.4 Performance Evaluation and Analysis

With the three types of paths set up as above, we run “MDS with Refinement” method to refine each local map (2-hop coverage) along each path, and obtain an estimate of remote node’s position through vector addition. 200-run of simulation is performed, and in each run, we blur the ranging measurement by adding zero-mean Gaussian-distributed errors. Then we could get 200 position estimates of remote node and thus obtain the simulated variance. Since LSE minimization can achieve CRLB when ranging error is Gaussian distributed, it makes sense to compare predicted variance bound with the simulated variance. As illustrated in Fig. 4.7, the simulated variances of each path are consistent with the predicted variance bounds, although they cannot match exactly with the predicted values. The reason is mainly because of the incomplete inclusion of vectors’ covariance. The simulated variances of CRLB-based path and heaviest-CI path in square-shape topology are similar, and are both better than that of lightest-CI path. In U-shape topology, CRLB-based path outperforms both of heaviest-CI and lightest-CI paths. That is because CRLB-based path does not walk along the inner circle of map, and enjoy high accuracy by walking through dense region.

As another benchmark for performance comparison, we simulate the variance of target node’s position by directly refining the “big” map which consists of all 2-hop nodes along an arbitrary shortest-path. It is also the optimal variance bound that can be achieved using the method in [35], as shown by the ellipse with grey surface in Fig. 4.7. Note that refining “big” map is much more computational expensive than vector addition. In order to have a salient comparison, here we use the lightest-CI path as the arbitrary shortest path. In square-shape topology, the variance from “big” map is comparable with that from CRLB-based path, although the ellipse eccentricities are a bit different. However, the variance from CRLB-based path is better than that from the “big” map in U-shape topology. The reason is because the “big” map in this case resides along the inner circle of the U-shape map and its structure becomes slim and flexible.

Fig. 4.7: Predicted variance bound versus simulated variance in square-shape topology (left) and U-shape topology (right): the coordinate is the one used at starting local node; the rectangular boxes are predicted bounds, whereas ellipses are simulated variances; the rectangle and ellipse with solid perimeter are results of CRLB-based methods, whereas those with dash and dot perimeter are of heaviest-CI shortest-path and lightest-CI shortest-path respectively; the ellipse with grey surface is the optimal variance bound using method in [35].

4.4 Performance Evaluation and Analysis

4.4.3 Deviation of Optimal Path

It shall be noted that in the previous sections we make use of the *true positions* of nodes when computing CRLB, predicting variance bound of resulting vector and selecting the optimal path. Although the input of connectivity information is always correct for computing CRLB, in real case the true positions information is never available to us. We could not get the true value of each vector's direction either, and the deviation of vectors' direction affects the X-axis and Y-axis components of predicted bound but doesn't affect the sum of them. Nevertheless, the refinement process, e.g. LSE minimization, provides close estimates of those nodes' position in local map if ranging error is not large, so that an approximated CRLB can be computed. Consequently, the optimal path might partially deviate from the one obtained from exact CRLB. The degree of deviation is shown in Fig. 4.8 with different standard deviation of ranging error. Coverage of 3-hop doesn't reduce much deviation as compared to 2-hop case, since the improvement on CRLB in 3-hop case is not significant.

Fig. 4.8: Degree of path deviation with respect to different ranging errors: the square-shape map in Fig. 4.6 is used; degree of overlap refers to the average number of hop-overlap between true-CRLB-deduced path and approximate-CRLB-deduced path; ranging error is zero-mean Gaussian distributed with different standard deviations.

4.4.4 Communication and Computational Overheads of Vector Addition Scheme

As the proposed Vector Addition localization scheme deals with selecting the optimal path to locate a remote node, its communication overheads are similar with those

4.4 Performance Evaluation and Analysis

in the classic routing problems of wireless ad-hoc networks. In other words, we can adopt available approaches for routing to construct protocol that supports our proposed Vector Addition localization scheme.

Take the CRLB-based path selection method for example. The first step at each node is to collect ranging measurements within its local map as defined by radius of a few hops. This can be done using simple broadcasting messages with limited Time-to-Live (TTL). Note that, in this way, we implicitly assume that two nodes with ranging measurement are within communication range. With ranging measurements available, the following step at each node is to perform MDS-plus-refinement to generate local map. In the last step, each node broadcasts the details of its local map (both coordinates and connectivity information in LCS) to the whole network, so that other nodes have sufficient information to perform the Vector Addition scheme.

The CI-based path selection heuristics in Chapter 4 has similar communication overheads with CRLB-based method, and it is computationally less expensive. Both methods need perform the expensive LSE minimization when generating local map. When looking for optimal path, CI-based heuristics requires the initial node to simply compute all shortest-paths (in term of the number of hops to remote node) and picks up the optimal one (either Heaviest or Stablest Path). However, CRLB-based method involves the expensive computation of CRLB. Whereas each node can locally compute its adjacent edges' CI and broadcast it to others, their local CRLB results are useless for path selection because CRLB results are dependent on which edge is positive X-axis. As a result, when looking for CRLB-based optimal path using Dijkstra's algorithm, the initial node has to perform $O(n)$ calculations of CRLB where n is the number of nodes in the network.

Chapter 5

Fast Message Dissemination: Collision-prohibited Case

In this chapter, we start embarking on the second main part of our research work: how to efficiently disseminate local ranging information to the whole network so that previously-proposed cooperative localization scheme can be performed. There are basically two ways to define the “efficiency” of a particular message dissemination strategy in wireless ad-hoc network: how many nodes have to transmit in one dissemination process, and how much time the whole process takes. The first definition is about the minimization of energy consumption [25] whereas the second one is regarding the minimization of broadcast latency. We focus our work on the later definition, *i.e.* to minimize the worst dissemination/broadcast latency. In details, once a node in wireless ad-hoc network has a ranging measurement to one of its neighbors, our proposed message dissemination strategy enables the ranging measurement to reach all nodes in the network within minimal time. It shall be noted that such fast message dissemination problem can be found not only in cooperative localization but also in many other practical applications in wireless ad-hoc network, such as dissemination of link state message in MANET and dissemination of critical control message in wireless sensor network.

We consider time as discrete, *i.e.* we divide time into many timeslots, each of which is equal to message’s transmission time in air. Message is initiated from a particular node and disseminated into all nodes in the network, *i.e.* it is a single source broadcast operation. Our job is to find a broadcast scheduling scheme so as to minimize the num-

5.1 The Wave Expansion Approach

ber of timeslots that are required to complete this broadcast operation. Chlamtac *et al.* [9] have proved this problem is NP-hard, and thus we intend to look for approximate solutions.

5.1 The Wave Expansion Approach

In order to minimize broadcast latency, we adopt a simple greedy approach (Heuristic H1 in [9]). In details, scheduled broadcast operation actually resembles a wave propagation process, and the number of “wave cycle¹” (i.e. the number of timeslots) for completing broadcast operation can be minimized if in every cycle the “wave-front” nodes cover maximal number of new (or untouched/uninformed) nodes. This approach is also called the Wave Expansion Approach according to [10].

Under this approach, the problem of low-latency broadcast scheduling can be reduced to some simple combinatorial optimization problem. In details, from a set of nodes which have already obtained the message, we have to select a set of nodes for re-transmission so that the number of new nodes which receive the message for the first time is maximum. This problem can be easily represented by a bipartite graph $H = (V + U, E)$, as shown in Fig. 5.1. V is set of informed nodes each of which covers at least one uninformed node, and U is set of uninformed nodes each of which has connection to at least one informed node; $n = |V|$, $\mu = |U|$; edges between vertices of the same set are trivial and hence omitted.

It shall be noted that, due to the nature of wireless medium, collision will occur at node u_j if two or more neighbors of u_j transmit concurrently. Thus based on whether collision is allowed or not, we are able to divide above-mentioned combinatorial optimization problem into two cases: collision is strictly prohibited, and collision is allowed. Clearly, regarding the maximum number of successful receptions in U , the collision-free case is less greedy than the collision case. We propose better approximate solutions for both cases and discuss them in this chapter and Chapter 6, respectively.

In literature, there have been some efforts devoted to this interesting combinatorial optimization problem. For collision-free case, two representative solutions are the Phase Assignment Algorithm in [9] and the 2-step Algorithm in [14]. Both of them

¹One wave cycle consists of several concurrent re-transmissions; wave cycle initiated from source is a special case with only one transmission.

5.2 Our Proposed Solution for Collision-free Case

Fig. 5.1: Bipartite graph representation

do not follow the wave expansion approach. Instead, the Phase Assignment Algorithm starts from an arbitrary neighbor of source node and gradually assigns re-transmitting timeslots according to some simple rules for collision avoidance. The 2-step Algorithm first finds a broadcast tree based on the construction of dominating set, and then in the second step it deploys similar phase assignment method to schedule the re-transmission of on-tree nodes. For collision-allowed case, the only available work, to the best of our knowledge, is the Spokesmen Election Algorithm (SEA) proposed in [10]. Details of SEA is to be discussed in Chapter 6. It should be noted that the authors of [10] named the collision-allowed case as Spokesmen Election Problem, which we believe is also applicable to collision-free case. Thus in the following part of this thesis, we will call the combinatorial optimization problem in Fig. 5.1 Generic Spokesmen Election Problem, and the nodes selected for re-transmission in set V are called spokesmen nodes.

5.2 Our Proposed Solution for Collision-free Case

As mentioned earlier, we adopt a greedy approach and try to make each broadcast wave cycle cover maximum number of untouched nodes without introducing collision. Referring to the sample network in Fig. 5.2 (a), after the broadcast wave cycle from S finishes, potential forwarders that are considered to start a new wave cycle include A , B and C , which can cover 2, 2, and 1 untouched nodes, respectively. Owing to collision of wireless radio, A and B could induce collision at D if they both re-transmit

5.2 Our Proposed Solution for Collision-free Case

immediately after receiving a message from S (i.e. at the same time instant). It is the same with B and C , whose concurrent transmissions induce collision at F . Therefore, rebroadcasts of A and B , as well as B and C , cannot be scheduled at the same time instant.

Fig. 5.2: Construction of weighted conflict graph: (a) Sample network with S as source; (b) Weighted conflict graph when building the second wave cycle (first wave cycle is the one initiated by S); value in parenthesis is weight.

Because of potential collision, there exists “conflict” between A and B as well as between B and C , when we try to select forwarders from them. This situation can be transformed into a conflict graph with weighted vertices, as depicted in Fig. 5.2 (b). In this type of conflict graph, a node’s weight refers to the number of untouched nodes that the node is able to cover, and an edge between two nodes implies that they have conflict. The objective to find a set of spokesmen nodes to cover maximum number of untouched nodes for a particular wave cycle is now shifted to find a set of un-adjacent vertices that have maximum total weight in the conflict graph, i.e. to solve the Maximum Weight Independent Set (MWIS) problem in graph theory. Intuitively, set $\{A, C\}$ is the optimal solution.

MWIS problem is NP-hard, and its optimal solution can only be found by “brute-force” (BF) search throughout all combinations of vertices. Some approximate algorithms have been designed to efficiently solve this problem in polynomial time, e.g. the GWMIN2 algorithm in [32]. GWMIN2 selects the vertex with maximum weight ratio in the subgraph induced by that vertex and its neighbors, removes it and its neighbors from graph, and iterates on remaining graph until no vertex left.

After finishing selecting spokesmen for a particular wave cycle, the next step is to proceed to construct a set of new potential forwarders for next wave cycle. New

5.2 Our Proposed Solution for Collision-free Case

potential forwarders consist of nodes that are just covered by previous wave cycle as well as nodes that are left unscheduled in previous set of potential forwarders. Nodes that do not have untouched neighbors can be taken out from potential forwarders set, since their weights are zero. The algorithm iterates and stops when there is no untouched node left in the network. Its pseudo-code is illustrated in Fig. 5.3. The running time of building conflict graph is bounded by $O(n^3)$ whereas the running time of GWMIN2 is bounded by $O(n^2)$, thus the proposed algorithm with GWMIN2 implemented has running time bounded by $O(n^3)$ where n is the number of nodes.

BroadcastScheduling ($G=(V,E), s$):

```

1. PotentialForwarders = {s}
2. VertexStatus(s) = 'being_considered', and VertexStatus(v) = 'untouched' for other  $v \in V$ 
3. timeInstant = 0
4. while PotentialForwarders  $\neq \phi$  // each iteration constructs one broadcast wave cycle
5.     find conflict graph  $G_{\text{conflict}}$  for  $v \in \text{PotentialForwarders}$ , and their weight  $W(v)$ 
6.     find MWIS of above  $G_{\text{conflict}}$ 
7.     NewlyCovered =  $\phi$ 
8.     for each  $v \in \text{MWIS}$ 
9.         Schedule(v) = timeInstant // scheduling transmission time instant
10.        VertexStatus(v) = 'scheduled'
11.        for each  $v' \in \{v' \mid \text{VertexStatus}(v') = \text{'untouched'}, (v, v') \in E\}$ 
12.            NewlyCovered = NewlyCovered  $\cup \{v'\}$ 
13.            VertexStatus(v') = 'newly_covered'
14.        endfor
15.    endfor
16.    timeInstant ++ // time is discrete
17.    TempSet = (PotentialForwarders - MWIS)  $\cup$  NewlyCovered
18.    PotentialForwarders =  $\phi$  // empty itself to build a new set for next iteration
19.    for each  $v \in \text{TempSet}$ 
20.        if there exists  $v'$ , such that  $(v, v') \in E$  and  $\text{VertexStatus}(v') = \text{'untouched'}$ 
21.            PotentialForwarders = PotentialForwarders  $\cup \{v\}$ 
22.            VertexStatus(v) = 'being_considered'
23.        else
24.            VertexStatus(v) = 'covered'
25.        endif
26.    endfor
27. endwhile

```

Fig. 5.3: Pseudo-code of proposed algorithm.

5.3 Numerical Results

5.3 Numerical Results

The proposed algorithm is evaluated in Matlab, with brute-force (BF) and GWMIN2 available for calculating MWIS. Performance metrics include the maximum delay time (i.e. maximum number of timeslots) for a broadcast operation to complete (as denoted by $D\text{-max}$) and the number of transmitting nodes (including source). If we let D_i denote the depth (or the number of hops) between source and a node i , the network depth as viewed from source is equal to $\text{Max}(D_i)$. Clearly, network depth represents an upper bound of the best performance of $D\text{-max}$, and thus it makes sense to evaluate the difference between $D\text{-max}$ and $\text{Max}(D_i)$. $\text{Max}(D_i)$ can be easily found through Breadth First Search (BFS). The 2-step Algorithm in [14] is also simulated in order to assess the merit of our solution.

Fig. 5.4: Performance result of $D\text{-max}$ with respect to network depth, summarized using boxplot: bold line segment refers to median, max whisker length is 1.5 times of interquartile range, '+' refers to outlier, and the number sitting above each boxplot is the mean value; result of BF in 100-node scenario is not available owing to its extremely long simulation time.

5.3 Numerical Results

In simulation, nodes are randomly deployed to form a connected graph in a 100x100 square area with transmission range equal to 20. We perform three scenarios by setting the number of nodes to 50, 75 and 100, and these scenarios have average degree of 5.2, 7.7 and 10.4, respectively. 500 runs are executed for each scenario. The most bottom left node of each network is selected as source for each run, in order to obtain a large network depth and consistent wave propagation pattern.

Fig. 5.5: Performance result of the number of transmissions, summarized using boxplot as described in Fig. 5.4.

Fig. 5.4 presents the results of $D-max$ with respect to network depth, i.e. $D-max$ minus $Max(D_i)$. It is clear that our proposed algorithm, implemented with either BF or GWMIN2 method for computing MWIS, outperforms the 2-step algorithm in this performance metric. Because of its approximation for selecting MWIS, GWMIN2 is not able to cover as many untouched nodes as BF does during each wave cycle. Hence, GWMIN2 suffers a higher $D-max$ result than BF, but the difference is quite small.

In contrast, BF requires a slightly larger number of transmissions to complete a

5.3 Numerical Results

broadcast operation than GWMIN2 does, as depicted in Fig. 5.5. During each wave cycle, although BF achieves an optimal coverage of untouched nodes through enumerating all forwarder sets, it could not guarantee that the average number of untouched nodes covered by a single transmission is also optimal. Therefore, BF is not able to produce an optimal performance of the number of transmissions, which is crucial in view of power consumption. On the other hand, GWMIN2 strikes a balance between the two metrics' performance, and gives us a smaller number of transmissions than BF, albeit not significantly.

As the number of nodes increases and network becomes denser, simulation results show that the gap between $D-max$ and network depth would become exacerbated. That is because, during each wave cycle, more potential untouched nodes are prohibited from being covered in dense network as a result of higher probability of collision. Similarly, the number of transmissions will also increase as network becomes denser, but at a rate much lower than that of the increasing number of total nodes. That is attributed in large part to the increased average number of untouched nodes covered by a single transmission.

Chapter 6

Fast Message Dissemination: Collision-allowed Case

In this chapter, we carry on previous discussion on the generic Spokesmen Election (SE) Problem and start investigating the collision-allowed case. Here we still refer to the bipartite graph representation of our problem in Fig. 5.1.

It has been proven in [10] that collision-allowed SE problem is NP-hard, and its optimal solution can only be found through enumerating all combination of nodes in V , i.e. by Brute-Force (BF) search. The work in [10] also provides the only available work in literature so far on collision-allowed Spokesmen Election Problem, which is called the Spokesmen Election Algorithm (SEA). SEA consists of two phases. The first step of SEA is to determine the optimal size of spokesmen set S . In details, SEA computes the average number of successful receptions in U for a particular size k of S ($1 \leq k \leq n$). Among n possible values of k , the one generating maximum successful receptions is selected as the optimal size of S . With the size of S fixed, the second step of SEA consists of k iterations, during each of which a new spokesmen node is selected based on similar “maximum average number of successful receptions” rule. The drawback of SEA is that the actual optimal size of S is not necessarily equal to k , and thus the resulting selection of spokesmen nodes in the second step is for sure not optimal. The lower bound of SEA’s performance is defined by $p = |U|/\ln|V|$, i.e. at least p percentage of nodes in U can have successful receptions in each broadcast wave cycle.

6.1 Problem Formulation and Illustration

We adopt a completely different approach by formulating our problem into a maximum stable set problem and easily solve it using available simple approximate methods.

6.1 Problem Formulation and Illustration

6.1.1 Problem Formulation

Let a solution of collision-allowed SE problem be denoted by a binary vector or a set of binary variables $\{x_1, x_2 \dots x_n\}$, where $n = |V|$, such that $x_i = 1$ means node $v_i \in V$ is selected as spokesmen node. For each pair of node $u_j \in U$ and node v_i , we let c_{ji} denote the connectivity between u_j and v_i (bidirectional link is assumed). Clearly, the bipartite graph in Fig. 5.1 can be expressed by a binary $\mu \times n$ matrix $C = [c_{ji}]$, where $\mu = |U|$. We also let binary variable y_j denote the status of u_j 's reception (i.e. 1 means successful). Whenever there is collision at node u_j , we have $y_j = 0$. Instead, $y_j = 1$ if and only if there is just one $c_{ji} \cdot x_i$ term equal to 1 among all n terms related to u_j . In term of mathematical equation, we get

$$y_j = \begin{cases} 1 & , \text{ if } \sum_{i=1}^n c_{ji} \cdot x_i = 1 \\ 0 & , \text{ otherwise} \end{cases} \quad (6.1)$$

Then, collision-allowed SE problem could be modeled as maximizing $\sum_{j=1}^{\mu} y_j$.

The form of y_j in (6.1) makes it difficult to find y_j 's maximal sum through conventional optimization techniques, and hence we shall revise it into a single algebraic format. By observing that c_{ji} , x_i , and y_j are all binary, we construct truth table of y_j and then derive its Boolean algebraic expression in term of only x_i :

$$y_j = \sum_{i:c_{ji}=1} (x_i \cdot \prod_{k \neq i: c_{jk}=1} \overline{x_k}) \quad (6.2)$$

The expression of (6.2) guarantees that y_j is equal to 1 only when with respect to node u_j there is a single connected node in V transmitting; otherwise if there is no or more than one connected node in V transmitting, y_j becomes zero. Now the objective function that we are going to maximize is

$$f(x_1, x_2 \dots x_n) = \sum_{j=1}^{\mu} y_j = \sum_{j=1}^{\mu} \sum_{i:c_{ji}=1} (x_i \cdot \prod_{k \neq i: c_{jk}=1} \overline{x_k}) \quad (6.3)$$

6.1 Problem Formulation and Illustration

6.1.2 An Example for Illustration

We present an example in Fig. 6.1 to illustrate above-mentioned formulation of collision-allowed SE problem. $V = \{v_1, v_2, v_3\}$ ($n = 3$) is set of informed nodes, whereas $U = \{u_1, u_2 \dots u_5\}$ ($\mu = 5$) is set of uninformed nodes. We shall select a set of nodes from V as spokesmen nodes, so that the number of correct receptions in U is maximal.

Fig. 6.1: An example of typical collision-allowed SE problem.

Based on the connectivity matrix

$$C = [c_{ji}]_{5 \times 3} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad (6.4)$$

and the expression in (6.2), we can write y_j in term of x_i for which $c_{ji} = 1$:

$$\begin{cases} y_1 = x_1 \\ y_2 = x_1 \\ y_3 = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot x_3 \\ y_4 = x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 \\ y_5 = x_3 \end{cases} \quad (6.5)$$

Then the solution of this typical collision-allowed SE problem is the binary value of

6.2 Posiform Maximization

$\{x_1, x_2, x_3\}$ that maximizes

$$\begin{aligned} f(x_1, x_2, x_3) &= \sum_{j=1}^5 y_j \\ &= x_1 + x_1 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \\ &\quad + \overline{x_1} \cdot \overline{x_2} \cdot x_3 + x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 + x_3 \end{aligned} \quad (6.6)$$

It shall be noted that in the expression of (6.5), whose overall format resembles the connectivity matrix C in (6.4), each non-zero term is corresponding to a *TRUE* entry in C , i.e. to an edge in E . Let t_{ji} denote the i -th term in y_j 's expression in (6.5) (term of zero is also counted in order to keep format consistency), and let e_{ji} denote the edge between vertices u_j and v_i . We say that there is correspondence among t_{ji} , e_{ji} , and c_{ji} . For example, the second term of y_4 , $t_{42} = x_2 \cdot \overline{x_3}$, corresponds to edge e_{42} between u_4 and v_2 . This kind of correspondence will help us visualize the relationship between collision-allowed SE problem and its corresponding algebraic maximization problem of (6.3), whose solution is to be discussed in following section.

6.2 Posiform Maximization

6.2.1 Basic Concepts

The expression of (6.3) is a pseudo-Boolean function[6], which is real-value and whose variables are binary. Furthermore, the function in (6.3) happens to be a special representation of pseudo-Boolean function, called *posiform*, whose general expression is

$$f(x_1, x_2 \dots x_n) = \sum_{i \in N} (w_i \cdot T_i) \quad (6.7)$$

where w_i is positive coefficient, $T_i = \prod_{j \in A_i} x_j \cdot \prod_{k \in B_i} \overline{x_k}$, and $A_i \cap B_i = \phi$.

Posiform maximization problem could be solved through its corresponding conflict graph, G_f [6]. In G_f , a vertex with weight of w_i is associated to T_i of (6.7), and an edge exists between the two vertices corresponding to T_i and T'_i if in T_i there is a variable x_j for which $\overline{x_j}$ can be found in T'_i , i.e. there is conflict between T_i and T'_i . Clearly, in this case, T_i and T'_i could not be *TRUE* or 1 at the same time. Therefore, maximization of posiform f is equivalent to finding G_f 's *maximum weighted stable set*.

6.2 Posiform Maximization

w_i in (6.7) is always equal to 1 for the type of function in (6.3), if we do not sum up those common terms, e.g. the two x_1 terms in (6.6). The reason why we do not combine common terms is twofold: one is because it is easier to visualize the correspondence between terms of f and edges of E ; another reason is because all vertices in G_f then have unit weight and we need only find G_f 's *stability number* (defined as the size of the largest stable set) rather than dealing with more difficult *maximum weighted stable set*.

6.2.2 Constructing Conflict Graph

We now return to the typical example in Fig. 6.1 to illustrate how to construct conflict graph G_f for collision-allowed SE problem. Recall that posiform f of the example is in (6.6). Since f 's eight terms come from the eight edges of H , we simply have eight unweighted vertices in G_f . As mentioned before, an edge exists between two vertices of G_f if there is conflict between these two vertices' corresponding terms in f . The decomposed format of f in (6.5) is more lucid than f 's normal format in (6.6) for us to find conflicts among those terms in f . Based on simple logic, we formulate two rules that determine conflicts among f 's terms:

1. For t_{ji} and $t_{j'i'}$ where $i \neq i'$, i.e. for two terms that have the same row index, there is conflict between them; in other words, terms at the same row induce a clique in G_f .
2. For t_{ji} and $t_{j'i'}$ where $j \neq j'$ and $i \neq i'$, conflict exists if either $c_{ji'} = 1$ or $c_{j'i} = 1$. (Note that there is no conflict between two terms having the same column index.)

Since there is one-to-one correspondence between t_{ji} and e_{ji} , above two rules could be rewritten with respect to edges of H :

1. There is conflict between two edges that have the same endpoint in U , e.g. between e_{31} and e_{32} .
2. There is conflict between two edges each of which contributes a distinct endpoint to a third edge, e.g. between e_{21} and e_{32} . (There is no conflict between two edges that have a common endpoint in V , e.g. between e_{11} and e_{21} .)

Fig. 6.2 depicts the resulting conflict graph G_f .

6.3 Maximum Stable Set of Conflict Graph

Fig. 6.2: Constructing G_f : each vertex of G_f corresponds to an edge of H and also a *TRUE* entry of connectivity matrix C ; graph on the left follows C 's format.

6.3 Maximum Stable Set of Conflict Graph

6.3.1 Greedy Algorithm

Let $\alpha(G)$ denote the stability number of graph G . Our next step is to find G_f 's maximum stable set and compute $\alpha(G_f)$. It is well-known that finding maximum stable set is NP-hard, and we have to turn to approximation algorithms for efficient solutions. Hochbaum summarizes a number of such approximation algorithms in [17]. For here, we would like to mention only the Greedy algorithm, which is the most primitive one.

There are two types of Greedy algorithm to find approximate maximum stable set: Minimum-degree Greedy (*Greedy-min*) and Maximum-degree Greedy (*Greedy-max*). Greedy-min selects the vertex of minimum degree as a vertex in stable set, removes that vertex as well as its adjacent vertices from the graph, and iterates on remaining graph until no vertex left. Greedy-max operates in an inverse way: it deletes the vertex of maximum degree until no edges left, and the remaining vertices constitute the approximate solution. For example when applied to G_f in Fig 6.2, both of these two algorithms generate stable set of $\{e_{11}, e_{21}, e_{43}, e_{53}\}$ and $\alpha(G_f) = 4$ (it happens to be the optimal solution). It implies that four nodes of U have correct receptions and spokesmen nodes can be selected by solving $t_{11} = t_{21} = t_{43} = t_{53} = 1$. It turns out that $x_1 = x_3 = 1, x_2 = 0$, and thus the spokesmen nodes are v_1 and v_3 . The four nodes

6.3 Maximum Stable Set of Conflict Graph

having correct receptions are u_1, u_2, u_4 , and u_5 .

According to [15], both Greedy-min and Greedy-max are able to achieve the Turán Bound on stability number, which states that

$$\alpha(G) \geq \frac{|V|}{\delta + 1} \quad (6.8)$$

where δ is the average vertex degree of G and $|V|$ is the number of vertices. Let R denote the set of nodes having correct receptions. Then, the Greedy algorithm guarantees that

$$|R| \geq \frac{|V_{G_f}|}{\delta_{G_f} + 1} = \frac{|E|}{\delta_{G_f} + 1} = \frac{\delta_U}{\delta_{G_f} + 1} \cdot \mu \quad (6.9)$$

where δ_{G_f} is the average vertex degree in G_f and $\delta_U = |E|/\mu$ refers to the average degree of U 's vertices in H . In other words, regarding the performance of Greedy algorithm in solving collision-allowed SE problem, $p = \delta_U/(\delta_{G_f} + 1)$ defines a lower bound on the fraction of nodes in U that have correct receptions. By fixing the sizes of V, U and E , we want to know which structure of H gives the smallest p (or the largest δ_{G_f} since δ_U is also fixed), in order to probe the worst-case performance.

6.3.2 Performance Lower Bound

Fig. 6.3: Overlap of *TRUE* entries in connectivity matrix C : χ_{jk} is used to analyze inter-row conflict; *TRUE* entries need not be contiguous.

Let κ denote the fixed size of E (i.e. fixed number of vertices in G_f) and φ denote the value of total degree (i.e. twice of the number of edges) in G_f . Then $\delta_{G_f} = \varphi/\kappa$. φ can be found according to the “conflict rules” in Section III-B. With G_f 's vertices aligned in connectivity matrix C 's format, e.g. the left graph in Fig. 6.2, we are able to analyze G_f 's total degree in a row-by-row manner. According to “Rule 1”, vertices

6.3 Maximum Stable Set of Conflict Graph

in G_f having the same row index forms a clique, e.g. among e_{31} , e_{32} , and e_{33} in Fig. 6.2. Let d_j denote the degree of a node u_j in H . Then such “self-conflict at row u_j ” contributes $d_j(d_j - 1)$ degree to G_f . Meanwhile, according to “Rule 2”, vertices at row u_j have conflicts with vertices at row u_k if there is overlap of *TRUE* entries χ_{jk} between row u_j and u_k , as exemplified in Fig. 6.3. Such “inter-row conflict owing to overlap” adds $(d_j - \chi_{jk})\chi_{jk} + \chi_{jk}(\chi_{jk} - 1) + \chi_{jk}(d_k - \chi_{jk}) = \chi_{jk}(d_j + d_k - 1) - \chi_{jk}^2$ degree to vertices at row u_j . Let φ_j denote the total degree of vertices at row u_j . Then we obtain

$$\varphi_j = d_j(d_j - 1) + \sum_{k \neq j} [\chi_{jk}(d_j + d_k - 1) - \chi_{jk}^2] \quad (6.10)$$

Therefore, G_f 's total degree is given by

$$\begin{aligned} \varphi &= \sum_{j=1}^{\mu} \varphi_j \\ &= \sum_{j=1}^{\mu} d_j^2 - \kappa + 2 \sum_{j=1}^{\mu} \sum_{k > j} [\chi_{jk}(d_j + d_k - 1) - \chi_{jk}^2] \end{aligned} \quad (6.11)$$

Given φ 's expression in (6.11), we investigate which structure of connectivity matrix C (or structure of H) generates the maximal value of φ . Recall that the degree contributed by “inter-row conflict” is characterized by $\chi_{jk}(d_j + d_k - 1) - \chi_{jk}^2$, whose value reaches maximal when $\chi_{jk} = (d_j + d_k - 1)/2$. Therefore, for any set of $\{d_1, d_2 \dots d_{\mu}\}$, φ is maximal when all χ_{jk} satisfy $\chi_{jk} = (d_j + d_k - 1)/2$. Note that χ_{jk} , d_j and d_k are all integer and $\chi_{jk} \leq \min(d_j, d_k)$. Hence the condition of $\chi_{jk} = (d_j + d_k - 1)/2$ can be revised to $\chi_{jk} = \min(d_j, d_k)$, i.e. maximal overlap of *TRUE* entries between any pair of rows. Assuming that the elements of $\{d_1, d_2 \dots d_{\mu}\}$ follow a non-increasing order, we obtain $\chi_{jk} = d_k$. Then φ in (6.11) becomes

$$\begin{aligned} \varphi &= \sum_{j=1}^{\mu} d_j^2 - \kappa + 2 \sum_{j=1}^{\mu} \sum_{k > j} [d_k(d_j + d_k - 1) - d_k^2] \\ &= \kappa^2 - \kappa - 2 \sum_{j=1}^{\mu} \sum_{k > j} d_k \end{aligned} \quad (6.12)$$

Now we shall find the minimal value of $\sum_{j=1}^{\mu} \sum_{k > j} d_k$. In other words, given a set of *TRUE* entries of size κ , we shall distribute them into each row of connectivity matrix C in such a way that $\sum_{j=1}^{\mu} \sum_{k > j} d_k$ reaches its minimum. Expand $\sum_{j=1}^{\mu} \sum_{k > j} d_k$:

6.3 Maximum Stable Set of Conflict Graph

	v_1	v_2	...	v_{n-1}	v_n
u_1	1	1	...	1	1
...			...		
u_{t-1}	1	1	...	1	1
u_t	1	...	1		
u_{t+1}	1	$d_{t+1} = l$			
...	...				
u_μ	1	$d_\mu = l$			

(a)

	v_1	...	v_{d_l}	...	v_n
u_1	1	...	1		
u_2	1				
...	...				
u_t	1				
u_{t+1}			1		
...				...	
u_μ					1

(b)

Fig. 6.4: Structure of connectivity matrix C that generates maximal φ : (a) when $\kappa \geq \mu + n - 1$; (b) when $\kappa < \mu + n - 1$.

$$\begin{aligned}
 \sum_{j=1}^{\mu} \sum_{k>j} d_k &= (d_2 + d_3 + \dots + d_\mu) + (d_3 + d_4 + \dots + d_\mu) \\
 &\quad + \dots + (d_{\mu-1} + d_\mu) + d_\mu \\
 &= d_2 + 2d_3 + 3d_4 + \dots + (\mu - 1)d_\mu
 \end{aligned} \tag{6.13}$$

From (6.13), we can infer that in order to obtain its minimal value, we shall cram from row d_1 to row d_μ in-order and at the same time maintain maximum overlap between each pair of rows. Note that each $d_j \geq 1$. Hence, the structure of C that generates minimal value of $\sum_{j=1}^{\mu} \sum_{k>j} d_k$ follows the format of Fig. 6.4(a), and we get

$$\begin{aligned}
 \left(\sum_{j=1}^{\mu} \sum_{k>j} d_k \right)_{min} &= n + 2n + \dots + (t - 2)n + (t - 1)d_t \\
 &\quad + t + (t + 1) + (t + 2) + \dots + (\mu - 1) \\
 &= \frac{1}{2}n(t - 1)(t - 2) + (t - 1)d_t + \frac{1}{2}(t + \mu - 1)(\mu - t)
 \end{aligned} \tag{6.14}$$

where $t = \lceil (\kappa - \mu) / (n - 1) \rceil$. Then according to (6.12), the maximal value of φ can be easily computed.

It shall be noted that the derivation from (6.12)-(6.14) is based on the assumption that the maximal overlap between any pair of rows is achievable. When κ is too small,

6.4 Numerical Results

the “maximal overlap” assumption could not hold. The threshold condition is that the first row is full whereas other rows have just one *TRUE* entry each, i.e. $\kappa = \mu + n - 1$. When $\kappa < \mu + n - 1$, we can only affirm that $\varphi_{max} < n(n-1) + 2(\mu-1)(n-1)$ and $n(n-1) + 2(\mu-1)(n-1)$ is the maximum of φ when $\kappa = \mu + n - 1$. In addition, we conjecture that φ reaches maximum when C is in the format of Fig. 6.4(b), and

$$\varphi_{max} = 2(t-1)(d_1-1) \quad (6.15)$$

where $t = \kappa - n + 1$.

6.4 Numerical Results

We evaluate our proposed solution of collision-allowed SE problem and show its superiority over SEA in low-latency broadcasting operation. The performance of Brute-Force (BF) search is also simulated in order to gauge how much our proposed solution is close to optimal. Among the two greedy algorithms, we select Greedy-max for computing approximate solution of G_f ’s maximum stable set (the result of Greedy-min is not as good as that of Greedy-max and thus is not presented here). The simulation setup is similar like the one for collision-free case. Nodes are randomly deployed to form a connected graph in an 80x80 area with transmission range equal to 20. By varying the number of deployed nodes from 40 to 70 with step of 10, we have totally four scenarios with different average node connectivities (or average vertex degrees). 500 runs are performed for each scenario and the average node connectivities of these four scenarios are 6.10, 7.61, 9.22, and 10.76, respectively. The most bottom left node in network is selected as source for each run, in order to obtain large network depth and consistent wave propagation pattern.

Performance metrics include maximum delay time (i.e. maximum number of wave cycles, as denoted by $D-max$) and number of transmissions (including source), for a broadcasting operation to complete. $D-max$ is evaluated with respect to network depth, which represents a bound on the best performance of $D-max$, as depicted in Fig. 6.5. Clearly, our proposed method (with Greedy-max) generates much lower $D-max$ than SEA in all scenarios. Owing to its approximation, our proposed solution may not be able to cover as many uninformed nodes as BF does in each wave cycle, and thus its resulting $D-max$ is a bit worse than that of BF. Fig. 6.6 illustrates the results of

6.4 Numerical Results

the number of transmissions. The advantage of our proposed solution over SEA is conspicuous, as the number of transmission is improved almost by half. SEA intends to overestimate the size of spokesmen set in its first step and thus suffers large number of transmissions.

Fig. 6.5: Performance results of the difference between $D\text{-max}$ and network depth, summarized in boxplot: bold line segment refers to median, max whisker length is 1.5 times of interquartile range, 'x' refers to outlier, and the number sitting above each boxplot is the mean value; results of BF in 60-node and 70-node scenarios are not available owing to their extremely long simulation time.

Note that the number of transmissions by SEA sometimes exceeds the number of total nodes. In other words, during certain wave cycle, SEA may adopt an inappropriate size of spokesmen set in such a way that resulting number of uninformed nodes with correct receptions is smaller than the size of spokesmen set.

Besides, our proposed method requires less number of transmission than BF albeit not significantly. The reason is because, although BF generates maximum number of correct receptions among uninformed nodes, it could not guarantee an optimal ratio between the number of correct receptions and the size of spokesmen set.

6.4 Numerical Results

Fig. 6.6: Performance results of the number of transmissions, summarized using boxplot as described in Fig. 6.5 except here median is not bold.

Chapter 7

Conclusions and Recommendations

From Chapter 2 to Chapter 4, we adopt a simple Vector Addition approach for locating remote nodes in wireless ad-hoc network. In order to adapt the proposed scheme to anchor-less wireless ad-hoc network, we formulate a series of procedures to set up Local Coordinate System (LCS) at each unknown node and manipulate coordinate system alignments among different unknown nodes' LCS. For each unknown node's local map, we adopt the "MDS with Refinement" approach to obtain accurate position estimates of 1-hop neighboring nodes in that local map. Multi-Dimensional Scaling (MDS) provides a rough and initial position estimates for refinement, whereas refinement using least square error (LSE) minimization achieves minimum variance estimation when ranging error is Gaussian distributed. Through the combination of "MDS with Refinement" and Vector Addition, we hope to strike a balance between computational cost and positioning accuracy for locating remote unknown nodes.

Most of our research work is focused on optimal path selection for Vector Addition. Inspired by the correlation between system equation's over-determinacy and positioning accuracy, we first propose a simple heuristics to quantify the "reliability" of each edge using a parameter called Confidence Index (CI). Numerical results show that CI is able to guide the vector addition path to walk through dense area and hence obtain high accuracy in term of end-to-end distance estimate. As CI-based approach suffers the short-path constraint, we proposed another path selection method based on Cramer-Rao Lower Bound (CRLB). We first extend available computation method of CRLB from anchored network map to anchor-less network map, *i.e.* to each unknown node's

local map. Then we analyze how to make use of CRLB to predict the variance bound of locating remote node by vector addition. Vibration of X-axis, which is another major contributing factor of variance bound, is also investigated and incorporated into the prediction procedure. Numerical results verify that CRLB-based approach indeed has better tendency than CI-based approach to walk through dense network area. What is more important, the accuracy of CRLB-based approach is comparable to that of previous method in [35] which is computationally expensive, and sometimes even better when network topology is irregular.

Since we have proved that our proposed CRLB-based approach indeed performs well in term of positioning accuracy, the future work is to further investigate how much advantage our proposed method could bring in term of resource consumptions (both communication cost and computation cost). Furthermore, in the theoretical aspect, we would like to analyze the relationship between information flow (message exchange) and positioning accuracy, so that we could better understand their tradeoff.

In Chapter 5 and Chapter 6, we focus on how to quickly disseminate local ranging measurement message to the whole network, whose problem nature is quite different from the first part. We adopt a simple yet efficient Wave Expansion Approach and formulate the problem as a generic Spokesmen Election (SE) problem, which has two sub-types: collision-free and collision-allowed. For collision-free case, the problem is formulated as a Maximum Weight Independent Set problem, and available technique does a good job in finding better approximate solution. For collision-allowed SE problem, we formulate it into a Maximum Stable Set problem by utilizing the concept of Posiform Maximization. Again, simple greedy techniques are able to generate much better result than available work in literature. Besides, we also perform a lower bound analysis on the number of successful reception with our proposed method.

We believe that there are still many unknown issues on generic SE problem especially in the collision-allowed case. Firstly, the performance bound analysis should be extended to whole broadcast process so that a upper bound on the number of wave cycles can be obtained. Secondly, the greedy approximate algorithm is not able to generate stability number result with constant approximation ratio. For example, Johnson [18] gave us a circumstance where greedy algorithm fails to produce “acceptable” stability number result, and our solution for collision-allowed SE problem might fall into

that pitfall as well. Therefore, it is desirable to prove whether a constant approximation ratio for collision-allowed SE problem is achievable or not and, if it really exists, to construct such approximation algorithm. Thirdly, we are intrigued to find that, given a particular set of U , what kind of bipartite graph H has the smallest optimal result. Last but not least, it is interesting to investigate how to optimize broadcasting latency and the number of transmissions at the same time.

Appendix A

Author's Publications

Here is a list of author's publications:

- W. Wang and B. H. Soong, "A Distributed Heuristics of Localization in Wireless Sensor Network," in *Proc. 2nd IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2006)*, Wuhan, China, Sep. 2006.
- W. Wang and B. H. Soong, "Locating remote nodes in anchor-free wireless sensor networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC'07)*, Hong Kong SAR, China, Mar. 2007, pp. 3353-3357.
- W. Wang and B. H. Soong, "Collision-free and Low-latency Scheduling Algorithm for Broadcast Operation in Wireless Ad Hoc Networks," *IEEE Communication Letters*, vol. 11, no. 10, Oct. 2007, pp. 793-795.
- W. Wang and B. H. Soong, "Spokesmen Election Problem Revisited," in *Proc. IEEE Vehicular Technology Conference, Spring (VTC-Spring'08)*, Singapore, May 2008, pp. 1732-1736.

References

- [1] Specification of crossbow imote2 platform. [Online]. Available: <http://www.xbow.com> 5
- [2] A. A. Ahmed, H. Shi, and Y. Shang, "SHARP: A new approach to relative localization in wireless sensor network," in *Proc. IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)*, Columbus, OH, Jun. 2005, pp. 892–898. 15
- [3] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communication Magazine*, vol. 40, no. 8, pp. 102–114, 2002. 4
- [4] J. Ash. Sensor network localization explorer. [Online]. Available: <http://www.ece.osu.edu/~randy/localization> 25
- [5] P. Bergamo and G. Mazzini, "Localization in sensor networks with fading and mobility," in *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'02)*, Lisbon, Portugal, Sep. 2002, pp. 750–754. 4
- [6] E. Boros and P. L. Hammer, "Pseudo-boolean optimization," *Discrete Applied Mathematics*, vol. 123, no. 1-3, pp. 155–225, 2002. 51
- [7] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad-hoc networks," in *Proc. IEEE Hawaii International Conference on System Science (HICSS-34)*, Maui, HI, Jan. 2001. 9, 10, 11, 16
- [8] J. C. Chen, K. Yao, and R. E. Hudson, "Source localization and beamforming," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 30–39, 2002. 5

REFERENCES

-
- [9] I. Chlamtac and S. Kutten, “On broadcasting in radio networks: Problem analysis and protocol design,” *IEEE Transactions on Communications*, vol. 33, no. 12, pp. 1240–1246, 1985. [41](#)
 - [10] I. Chlamtac and O. Weinstein, “The wave expansion approach to broadcasting in multihop radio networks,” *IEEE Transaction on Communications*, vol. 39, no. 3, pp. 426–433, 1991. [41](#), [42](#), [48](#)
 - [11] L. M. Feeney, B. Ahlgren, and A. Westerlund, “Spontaneous networking: an application-orientated approach to ad hoc networking,” *IEEE Communication Magazine*, vol. 39, no. 6, pp. 176–181, 2001. [1](#)
 - [12] J. F. Figueroa and J. S. Lamancusa, “A method for accurate detection of time of arrival: Analysis and design of an ultrasonic ranging system,” *the Journal of the Acoustical Society of America*, vol. 91, no. 1, pp. 486–494, 1992. [4](#)
 - [13] F. Galton, “Classification of portraits,” *Nature*, vol. 76, no. 1981, pp. 617–618, 1907. [8](#)
 - [14] R. Gandhi, S. Parthasarathy, and A. Mishra, “Minimizing broadcast latency and redundancy in ad hoc network,” in *Proc. 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc’03)*, Annapolis, MD, Jun. 2003, pp. 222–232. [41](#), [45](#)
 - [15] M. M. Halldorsson and J. Radhakrishnan, “Greedy is good: Approximating independent sets in sparse and bounded-degree graph,” *Algorithmica*, vol. 18, no. 1, pp. 145–163, 1997. [54](#)
 - [16] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, “Energy-efficient surveillance system using wireless sensor networks,” in *International Conference on Mobile System, Applications and Services (MobiSys’04)*, Boston, MA, Jun. 2004, pp. 270–283. [2](#)
 - [17] D. S. Hochbaum, *Approximation Algorithm for NP-hard Problems*. Boston, MA: PWS, 1997. [53](#)

REFERENCES

-
- [18] D. S. Johnson, "Approximation algorithms for combinatorial problems," in *Proceedings of the fifth annual ACM Symposium on Theory of Computing*, Austin, TX, Apr. 1973, pp. 38–49. [61](#)
 - [19] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *International Conference of Architectural Support for Programming Languages and Operating Systems (ASPLOS'02)*, San Jose, CA, Oct. 2002, pp. 96–107. [2](#)
 - [20] E. D. Kaplan, *Understanding GPS, Principles and Applications*. MA: Artech House Publishers, 1996. [29](#)
 - [21] S. M. Kay, *Fundamentals of Statistical Signal Processing (volume 1): Estimation Theory*. Upper Saddle River, NJ: Prentice Hall, 1993. [13](#), [23](#), [29](#)
 - [22] J. Kumagai and S. Cherry, "Sensors and sensibility," *IEEE Spectrum*, vol. 41, no. 7, pp. 22–28, 2004. [2](#)
 - [23] Z. N. Low, J. H. Cheong, C. L. Law, W. T. Ng, and Y. J. Lee, "Pulse detection algorithm for line-of-sight (LOS) UWB ranging applications," *IEEE Antennas and Wireless Propagation Letters*, vol. 4, 2005. [4](#)
 - [24] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics (SIAM)*, vol. 11, no. 2, pp. 431–441, 1963. [13](#)
 - [25] M. Min, O. Prokopyev, and M. Pardalos, "Optimal solutions to minimum total energy broadcasting problem in wireless ad hoc networks," *Journal of Combinatorial Optimization*, vol. 11, 2006. [40](#)
 - [26] E. D. N. B. Priyantha, H. Balakrishnan and S. Teller, "Anchor-free distributed localization in sensor networks," MIT Laboratory for Computer Science, Tech. Rep. 892, Apr. 2003. [12](#), [13](#), [15](#)
 - [27] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proc. IEEE Global Telecommunications Conference (GlobeCom'01)*, San Antonio, TX, Nov. 2001, pp. 2926–2931. [11](#)

REFERENCES

-
- [28] N. Patwari, J. N. Ash, S. Kyperountas, A. O. H. III, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005. [12](#)
 - [29] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor network," *IEEE Transaction on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, 2003. [24](#), [26](#)
 - [30] M. Pollack, "The k-th best route through a network," *Operations Research*, vol. 9, no. 4, pp. 578–580, 1961. [32](#)
 - [31] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *International Conference on Mobile Computing and Networking (MobiCom'00)*, Boston, MA, Aug. 2000, pp. 32–43. [4](#)
 - [32] S. Sakai, M. Togasaki, and K. Yamazaki, "A note on greedy algorithm for the maximum weighted independent set problem," *Discrete Applied Mathematics*, vol. 126, no. 2-3, pp. 313–322, 2003. [43](#)
 - [33] C. Savarese, K. Langendoen, and J. Rabaey, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *USENIX Technical Annual Conference*, Monterey, CA, Jun. 2002, pp. 317–328. [11](#)
 - [34] A. Savvides, H. Park, and M. Srivastava, "The bits and flops of the N-hop multilateration primitive for node localization problems," in *International Workshop on Wireless Sensor Networks and Application (WSNA'02)*, Atlanta, GA, Sep. 2002, pp. 112–121. [11](#)
 - [35] Y. Shang, J. Meng, and H. Shi, "A new algorithm for relative localization in wireless sensor networks," in *Proc. IEEE International Parallel and Distributed Processing Symposium (IPDPS'04)*, Santa Fe, NM, Apr. 2004. [36](#), [37](#), [61](#)
 - [36] Y. Shang and W. Ruml, "Improved MDS-based localization," in *Proc. Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*, Hong Kong SAR, China, Mar. 2004, pp. 2640–2651. [14](#), [15](#)

REFERENCES

- [37] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, “Localization from connectivity in sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961–974, 2004. [11](#), [13](#), [14](#), [20](#)
- [38] K. Whitehouse and D. Culler, “Calibration as parameter estimation in sensor networks,” in *International Workshop on Wireless Sensor Networks and Application (WSNA ’02)*, Atlanta, GA, Sep. 2002, pp. 59–67. [20](#), [25](#)
- [39] Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed energy conservation for ad hoc routing,” in *International Conference on Mobile Computing and Networking (MobiCom’01)*, Rome, Italy, Jul. 2001, pp. 70–84. [5](#)
- [40] M. Zorzi and R. R. Rao, “Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance,” *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 349–365, 2003. [5](#)