Algorithmic Aspects of Communication and Localization in Wireless Sensor Networks

zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Markus Völker

aus Aschaffenburg

Tag der mündlichen Prüfung:	14. Mai 2012
Erster Gutachter:	Frau Prof. Dr. Dorothea Wagner
Zweiter Gutachter:	Herr Prof. Dr. Roger Wattenhofer

Acknowledgments

During my time as a PhD student, I was supported by many people who either directly or indirectly contributed to my work and to this thesis. At this point I want to take the opportunity to thank all these people and to express my sincere gratitude.

First of all, I want to thank my advisor Dorothea Wagner, for giving me the possibility to work in her group, for the constant support, and for her very friendly and respectful leadership, which significantly contributed to the pleasant working atmosphere in our group. Additionally, I want to thank all the others who are involved in the organization of the DFG research training group 1194 on self-organizing sensor-actuator networks, for all the effort they put into the project and for many interesting discussions, and Roger Wattenhofer, for willingly taking over the task of reviewing this thesis.

I also want to thank all my current and former colleagues in our working group and in the research training group on sensor networks, for supporting my work in many ways and for always being there when I needed work-related input or a brief distraction from work. A big thank-you for all the fruitful discussions and for the good collaborations goes to all co-authors of my publications: Reinhard Bauer, Sergey Bereg, Felix Brandt, Andreas Cardeneo, Tobias Gädeke, Danny Holten, Bastian Katz, Klaus D. Müller-Glaser, Martin Nöllenburg, Dennis Schieferdecker, Johannes Schmid, Dorothea Wagner, Alexander Wolff, and Yuanyi Zhang. I especially want to thank Alexander Wolff for his support during my time as a student, which certainly also influenced my work as a PhD student.

Furthermore, I am very thankful to the German Research Foundation (DFG) for financing my work, to Lilian Beckert, Elke Sauer, and Bernd Giesinger for taking care of all problems concerning administration and computer infrastructure, and to Andreas Gemsa, Tanja Hartmann, Thomas Pajor, Ignaz Rutter, and Dennis Schieferdecker for proof-reading parts of this thesis.

Last but not least, I also want to sincerely thank all the people who supported me outside of work with their encouragement and patience, especially my parents and Tatiane.

Deutsche Zusammenfassung (German Summary)

Unter drahtlosen Sensornetzen (DSN) versteht man Netze aus einer Vielzahl kleiner Sensorknoten, die ihre Umwelt mittels Sensoren überwachen und drahtlos miteinander kommunizieren können. Die einzelnen Sensorknoten sind dabei vollwertige, wenngleich leistungsschwache Kleinstrechner. Die stetig voranschreitende Miniaturisierung von Hardwarekomponenten und der damit einhergehende Preisverfall haben dazu geführt, dass Sensornetze bereits heute zur großflächigen Überwachung von Gebieten eingesetzt werden, beispielsweise zur Detektion von Waldbränden oder zur automatischen Bewässerung. Es ist davon auszugehen, dass künftig immer größere und leistungsfähigere Sensornetze verfügbar sein werden.

Die vorliegende Arbeit, die im Rahmen des Graduiertenkollegs 1194 *Selbstorganisierende Sensor-Aktor-Netzwerke* entstanden ist, behandelt algorithmische Fragestellungen zur Kommunikation und zur kommunikationsbasierten Lokalisierung in drahtlosen Sensornetzen. Die Arbeit gliedert sich in zwei Teile. Im Bereich der Kommunikation steht das *Scheduling*, die zeitliche Planung von drahtlosen Übertragungen, im Fokus. Der Lokalisierungsteil beschäftigt sich mit der automatischen Erkennung topologischer Eigenschaften drathloser Netze unter ausschließlicher Nutzung von Nachbarschaftseigenschaften im zugrundeliegenden Kommunikationsgraphen, der schritthaltenden Lokalisierung mobiler Sensorknoten anhand von Signalstärkemessungen und der indirekten Lokalisierung statischer Sensornetze mithilfe eines mobilen Sensorknotens. Es folgt eine knappe Übersicht der behandelten Fragestellungen sowie der erzielten Resultate.

Kommunikation in drahtlosen Netzen

Dieser Teil der Arbeit beschäftigt sich mit der effizienten und ressourcensparenden Kommunikation in drahtlosen Sensornetzen. Hierbei wird das physikalisch motivierte *Signal-To-Interference-Plus-Noise-Ratio (SINR) Modell* zu Grunde gelegt, bei dem man vereinfachend davon ausgeht, dass Übertragungen genau dann erfolgreich sind, wenn das Verhältnis zwischen empfangener Signalstärke und der Summe der auftretenden Interferenzen einen bestimmten Grenzwert überschreitet. Verglichen mit graphbasierten Kommunikationsmodellen, die in der Algorithmik bis vor wenigen Jahren fast ausschließlich verwendet wurden, erlaubt das SINR-Modell eine wesentlich genauere Modellierung tatsächlich auftretender Effekte. Diese realistischere Modellierung wird jedoch mit einer höheren Komplexität beim Entwurf und der Analyse von Algorithmen erkauft. Folgende Fragestellungen werden behandelt:

Dynamische Sendeleistungskontrolle

Das Problem der *Sendeleistungskontrolle* besteht darin, für eine gegebene Menge drahtloser Übertragung minimale Sendeleistungen zu bestimmen, so dass unter Annahme des SINR-Modells Fehlübertragungen durch Interferenzeffekte ausgeschlossen werden können. Die Sendeleistungskontrolle ist ein zentraler Bestandteil zahlreicher Scheduling-Algorithmen. Die gebräuchlichen Ansätze zur Lösung des Problems können zwar zu einer gegebenen Menge drahtloser Übertragungen effizient optimale Sendeleistungen bestimmen, sie sind aber nicht darauf ausgelegt dynamisch auf kleinere Änderungen im Netzwerk zu reagieren. Gerade solche kleineren Änderungen sind aber im Zusammenhang mit Schedulingalgorithmen besonders interessant, da man effizient testen möchte, welche Auswirkungen die Aktivierung bestimmter Übertragungen auf die optimalen Sendeleistungen der anderen Sender hätte.

Die in der vorliegenden Arbeit vorgeschlagenen Datenstrukturen zur Sendeleistungskontrolle nehmen sich dieses Problems an. Neben der effizienten Berechnung optimaler Sendeleistungen zu einer gegebenen Menge von Sendern erlauben sie insbesondere, ausgehend von einer Teillösung sehr effizient auf kleinere Änderungen in der Eingabe zu reagieren.

Zeitliche Planung drahtloser Übertragungen

Die Idee beim Scheduling drahtloser Übertragungen besteht darin, durch zeitliche Trennung stark miteinander interferierender Übertragungen das Auftreten von Fehlübertragungen zu vermeiden. Hierdurch erhofft man sich sowohl höhere Datendurchsätze, als auch Energieeinsparungen durch das Vermeiden wiederholter Übertragungsversuche.

Aufbauend auf den zuvor beschriebenen Datenstrukturen zur dynamischen Sendeleistungskontrolle wurden neue Verfahren zum Scheduling entwickelt. Diese Verfahren grenzen sich von existierenden Ansätzen dadurch ab, dass sie zu aktivierende Übertragungen danach auswählen, welche Effekte deren Aktivierung auf die optimalen Sendeleistungen der anderen aktiven Sender hätte. Mit Hilfe der neuen Methoden zur Bestimmung optimaler Sendeleistungen ist dies effizient möglich, und im simulationsbasierten Vergleich mit mehreren existierenden Ansätzen hat sich gezeigt, dass durch diese Strategie Verbesserungen sowohl bezüglich des erreichbaren Datendurchsatzes, als auch bezüglich der benötigten Sendeleistungen, möglich sind.

Scheduling im geometrischen $\mathrm{SINR}_{\mathrm{G}}$ Modell

Das SINR-Modell selbst macht keinerlei Aussagen darüber, wie sich die Signalstärke in Abhängigkeit von der Distanz zwischen Sender und Empfänger verhält. Aus theoretischer Sicht ist es daher sehr schwierig, die Güte von Scheduling-Algorithmen analytisch zu untersuchen. Aus diesem Grund wurde vor einigen Jahren das geometrische $\mathrm{SINR}_{\mathrm{G}}$ Modell eingeführt, bei dem man zusätzlich vereinfachend annimmt, dass der Abfall der Signalstärke über die Distanz durch ein Potenzgesetz beschrieben werden kann. Unter dieser zusätzlichen Annahme ist es möglich, beispielsweise Approximationsgarantien für Algorithmen zu beweisen.

Im Zusammenhang mit dem Scheduling im $\mathrm{SINR}_{\mathrm{G}}$ Modell sind noch einige Komplexitätsfragen ungelöst. In der vorliegenden Arbeit wird gezeigt, dass das Scheduling mit Sendeleistungskontrolle im $\mathrm{SINR}_{\mathrm{G}}$ Modell NP-schwer ist, sofern die zulässigen Sendeleistungen beschränkt sind oder nur aus einer Menge zur Verfügung stehender Sendeleistungen gewählt werden können.

Lokalisierung in drahtlosen Sensornetzen

Zusammenhangsbasierte Randerkennung

Bei der zusammenhangsbasierten Randerkennung betrachtet man ein sehr großes Sensornetz und möchte herausfinden, welche der Knoten sich in der Nähe des äußeren Randes des Netzes oder in der Nähe eines größeren Lochs, also eines Bereiches ohne Sensorknoten im Inneren des Netzes, befinden. Hierzu wird angenommen, dass man über keinerlei Informationen bezüglich der Positionen der Sensorknoten oder der Winkel zwischen Sensorknoten verfügt. Die einzige Information, auf die man zurückgreifen kann, ist das Wissen darüber, welche Sensorknoten sich zueinander in Kommunikationsreichweite befinden.

Auch für dieses Problem existieren bereits mehrere Lösungsansätze. Der Fokus bei dem in dieser Arbeit entwickelten Verfahren liegt darauf, mit möglichst wenig Kommunikation und Berechnungsaufwand auszukommen, um das Verfahren auch für schwache Sensorknoten und dynamische Szenarios einsetzbar zu machen. Mit Hilfe des entwickelten

Ansatzes kann jeder Knoten selbständig anhand von Zusammenhangsinformationen aus seiner lokalen 2-Hop Nachbarschaft entscheiden, ob er sich potenziell in der Nähe einer Begrenzung des Netzes befindet. Hierzu wird ausgenutzt, dass sich Randknoten von inneren Knoten im Bezug auf statistische Eigenschaften bezüglich der Längen von Zyklen in der 2-Hop Nachbarschaft, welche bestimmte Eigenschaften aufweisen, unterscheiden. Anhand von Simulationen wurde der beschriebene Ansatz mit mehreren etablierten Verfahren zur Randerkennung verglichen. Dabei hat sich herausgestellt, dass er im Vergleich zu ähnlich gut klassifizierenden Ansätzen sehr effizient ist und insbesondere relativ stabil bezüglich inhomogener Verteilungen der Sensorknoten.

Signalstärkebasiertes Tracking

Beim signalstärkebasierten Tracking geht es um die schritthaltende Lokalisierung einer Person oder eines mobilen Gerätes anhand von Signalstärkemessungen. Die Idee ist, dass das mobile Gerät, beispielsweise ein Handy, die Signalstärken aus Paketen auswertet, die es von Ankerknoten mit bekannten Positionen empfängt, und daraus die Distanz zu den Ankerknoten schätzt. Das beschriebene Lokalisierungsproblem wird seit Jahren intensiv erforscht. Häufig verwendete Ansätze sind Varianten des Kalmanfilters, Partikelfilter und das Fingerprinting, bei dem man in einer Trainingsphase für eine Vielzahl von Referenzpositionen die empfangenen Signalstärken misst und später zur Lokalisierung die wahrscheinlichste Referenzposition ermittelt.

In der vorliegenden Arbeit wird die Anwendung von kräftebasierten Verfahren auf das signalstärkebasierte Tracking untersucht. Kräftebasierte Verfahren werden seit Jahrzehnten erfolgreich im Bereich der Visualisierung von Graphen und der Einbettung von Netzwerken eingesetzt. Man modelliert dabei gewünschte Distanzen zwischen Knoten durch Federn, die zwischen die Knoten gespannt sind. Die Ideallängen der Federn entsprechen dabei den Wunschdistanzen. Ist die aktuelle Distanz zwischen den Knoten geringer als die Wunschdistanz, so wirken abstoßende Kräfte, andernfalls wirken anziehende Kräfte. Die Kräfte sind umso größer, je größer die Abweichung von der Wunschdistanz ist. Eine gute Einbettung kann dann dadurch gefunden werden, dass man iterativ ein Kräftegleichgewicht herstellt.

Die Einbettung eines ganzen Netzes ist relativ aufwändig. Glücklicherweise ist man beim Tracking in der komfortablen Situation, dass in jedem Zeitschritt nur ein einzelner Knoten eingebettet werden muss. Daher kann man erwarten, dass die Positionsbestimmung effizient möglich ist. In der vorliegenden Arbeit wird untersucht, wie sich Wissen über die Signalausbreitung in Gebäuden sinnvoll in geeignete Definitionen für Federkräfte übertragen lässt. Des Weiteren wird analysiert, wie sich Bewegungsmodelle durch Einführung zusätzlicher Kräfte realisieren lassen.

Die Leistungsfähigkeit der vorgeschlagenen kräftebasierten Trackingverfahren wird anhand von Echtweltdaten und Simulationen im Vergleich mit erweiterten Kalmanfiltern (EKFs) untersucht. Die Echtweltdaten stammen hierbei aus Messungen in einem Sensornetz mit 60 Knoten und sind ebenfalls im Rahmen des Graduiertenkollegs 1194 in einem anderen Teilprojekt entstanden.

Bei den Untersuchungen hat sich gezeigt, dass der Einsatz kräftebasierter Verfahren zum Tracking vielversprechend ist. Die Laufzeitkomplexität pro Zeitschritt ist linear in der Anzahl der verwendeten Messungen, und da zur Bestimmung resultierender Kräfte lediglich Vektoren addiert werden müssen ist das Verfahren sehr effizient. Darüber hinaus gibt es nahezu keine Einschränkungen bezüglich der Definition weiterer Kräfte, wodurch man sehr einfach zusätzliches Wissen im Verfahren integrieren kann. So kann man beispielsweise auf den Einsatz von Linearisierungen verzichten, die bei anderen Ansätzen wie dem erweiterten Kalmanfilter nötig sind.

Indirekte Lokalisierung drahtloser Sensornetze

Zahlreiche Anwendungen in Sensornetzen benötigen Wissen über die Positionen der einzelnen Sensorknoten. Je nach Einsatzszenario existieren hierfür bereits zahlreiche Methoden, die es ermöglichen, dass das Sensornetz sich nach der Ausbringung selbständig lokalisiert.

Die vorliegende Arbeit beschäftigt sich mit der indirekten Lokalisierung von Sensornetzen. Im Vordergrund steht ein Szenario, bei dem eine größerer Anzahl von Sensorknoten in einem Gebäude ausgebracht wird. Es wird davon ausgegangen, dass die einzelnen Sensorknoten keinerlei Information über ihren Aufenthaltsort besitzen. Zur Lokalisierung des Sensornetzes sollen ein oder mehrere mobile Sensorknoten eingesetzt werden, welche regelmäßig Pakete von den stationären Sensorknoten erhalten und anhand der empfangenen Signalstärken Distanzen zu den stationären Knoten schätzen. Ausgehend von einer Vielzahl solcher Distanzschätzungen soll anschließend das Netzwerk lokalisiert werden.

Das in dieser Arbeit untersuchte Verfahren grenzt sich von den meisten existierenden Ansätzen dadurch ab, dass weder Informationen über exakte oder ungefähre Positionen einzelner Sensorknoten zur Verfügung stehen, noch über die absolute oder relative Bewegung der mobilen Sensorknoten. Insbesondere die relativ großen und teils systematischen Fehler, die sich zwangsweise aus der signalstärkebasierten Distanzschätzung in Innenräumen ergeben, spielen in den meisten existierenden Arbeiten zur indirekten Lokalisierung keine Rolle.

Als Einsatzszenario für den beschriebenen Lokalisierungsansatz wird die Ausbringung eines Sensornetzes in einem Gebäude zum anschließenden Tracking mobiler Sensorknoten näher untersucht.

Contents

Chapter 1

Introduction

Wireless Sensor Networks (WSN) consist of small autonomous devices, which are spatially distributed and use sensors to cooperatively monitor physical or environmental conditions. Each sensor node can be regarded as an independent small computer, which is equipped with sensors and communication hardware. The ongoing miniaturization of sensor nodes and the availability of cheaper and cheaper hardware make it possible to use sensor networks for large-scale monitoring tasks. Possible applications range from environmental monitoring over industrial machinery surveillance to home automation [ZJ09]. It is hoped that sometime soon the sensor nodes will become small and cheap enough that thousands of them can be scattered in the environment to work cooperatively. This concept became known as smart dust [KKP99], because the devices are intended to be only the size of dust particles. It is believed that in the near future, sensor networks will play a very important role in many areas of everyday life. Thus, it is no surprise that there has been an enormous amount of research on sensor networks over the last years.

The constraints concerning both size and costs of sensor nodes put high demands on the used hardware and software. Accordingly, many applications for wireless sensor networks introduce new and interesting challenges to the design of algorithms. Sensor nodes are usually operated by battery and it is not always possible to replace batteries when sensor nodes run out of power. This means that lifetime and operability of sensor networks are limited by the available energy, and *energy-efficiency* is one of the major goals in the design of algorithms for wireless sensor networks. Another aspect that distinguishes wireless sensor networks from conventional wireless networks is that in many scenarios both computational power and available memory are significantly limited. This means that one is especially interested in *resource-efficient* algorithms. The large dimensions of the intended sensor networks in terms of node number makes it also necessary that algorithms for sensor networks scale very well with network size. In this context, the focus lies on *distributed* algorithms in which the workload is spread. Due to the high number of involved devices and the energy limitations, one also has to deal with failures of devices. This can for example be achieved by introducing a certain degree of *redundancy*.

In many applications it is also necessary that sensor nodes know their absolute or relative location to identify the position where certain measurements were taken. Such information is for example necessary to locate outbreaks of forest fires or cracks in pipelines. There are many possibilities how position estimates can be obtained. If the network is rather small, it might be possible to initialize the position of each node manually. For outdoor scenarios, systems such as the Global Positioning System (GPS) [HWLC97] or Galileo [HWBL$^+$08] can be used to locate sensor nodes with a precision of a few meters. However, in the context of sensor networks there are also many applications where location discovery is still a challenging problem. Manual localization is ruled out as soon as many nodes are involved, if nodes are mobile, or if nodes are distributed in some random and unpredictable fashion, e.g., by throwing them out of a plane to distribute them efficiently over a vast area. The problem with the use of GPS or similar systems is that they only work outdoors and they also require that the sensor nodes are equipped with additional hardware. This makes the nodes larger and more expensive, which might be prohibitive in some applications. For this reasons, location discovery is one of the fundamental research topics in the context of wireless sensor networks.

There exists an extensive amount of literature concerning various aspects of wireless sensor networks. More detailed introductions to wireless sensor networks and their applications are presented in [ASSC02, ZJ09]. Overviews with focus on algorithmic aspects can be found in [WW07, Bou09].

1.1 Overview and Contributions

This thesis deals with several aspects concerning communication and localization in wireless networks. Roughly speaking, in the communication part of this thesis the focus is on reliable and energy-efficient communication in wireless networks, whereas the localization part deals with aspects concerning location discovery in wireless sensor networks. Despite this separation into two parts, both topics are to some extent intertwined. The location discovery, as considered in this thesis, is based on communication between nodes, and the communication between nodes relies significantly on the properties of signal propagation in wireless networks, which is extensively analyzed in the localization part. We now give a brief outline of this thesis and an overview of the main contributions.

Chapter 2: Preliminaries

This chapters presents an introduction to general notations, models, and techniques that build the foundations of this thesis. It covers mathematical foundations, methods to compute embeddings of graphs, and some information on information processing using Kalman filters. Additionally, an introduction to the modeling of radio propagation and interference is given. Several existing models are described and compared with respect to their benefits and shortcomings. The chapter concludes with a brief survey on localization in wireless sensor networks, which covers the estimation of distances and angles between nodes, the localization of static nodes, the tracking of mobile nodes, and the localization of whole sensor networks.

Part I: Communication

Chapter 3: Dynamic Power Control

The problem of power control considers the computation of minimum transmission powers for a given set of wireless transmissions. In this chapter, we present new data structures for the computation of optimum transmission powers. In contrast to existing approaches, the proposed data structures are optimized for efficient dynamic updates of optimum transmission powers. Especially when there are only slight changes in the input, e.g., when single additional transmissions are activated or deactivated, the presented data structures allow for significantly more efficient updates of optimum transmission powers. Three slightly different data structures are presented, which differ in the supported operations and the corresponding complexities.

Chapter 4: Energy-Efficient Scheduling

We introduce new heuristics for the joint problem of transmission scheduling and power control. In contrast to existing approaches, the presented heuristics aim at minimizing the transmission powers that are necessary to execute a given set of wireless transmissions within a given timespan. For this purpose, they use the full knowledge about resulting optimum transmission powers to decide which of the transmissions are activated concurrently. In order to compute power changes efficiently, the algorithms make extensive use of the data structures that are presented in Chapter 3.

In a simulation-based comparison with existing approaches, it turns out that the new scheduling heuristics outperform the other approaches with respect to both throughput and power-efficiency, i.e., they yield Pareto-superior schedules. Additionally, they can be used to find a good tradeoff between throughput and required transmission powers.

Chapter 5: Complexity of Scheduling with Power Control in SINR_G

The geometric SINR_G model is a theoretical model for signal propagation and interference in wireless networks. The model is frequently used for the design of scheduling algorithms with provable worst-case guarantees. In the context of the SINR_G model, some complexity questions are still unresolved. We present an NP-hardness proof for the problem of scheduling with power control in the geometric SINR_G model. The proof applies to situations where upper and lower bounds on the available transmission powers exist. This also includes situations where the transmission powers can be chosen from a discrete set of allowed power levels.

Part II: Localization

Chapter 6: Connectivity-based Detection of Network Boundaries

Several applications in wireless sensor networks require knowledge about boundaries and holes of the network. We present a new distributed algorithm for connectivity-based boundary recognition. The main benefit of the new algorithm in contrast to existing approaches is that it requires very little information—only connectivity information of a local 2-hop neighborhood—to decide very reliably whether a node is close to a hole or boundary. The algorithm is compared in extensive simulations to several existing approaches. Despite its low computational cost, the algorithm produces in all considered settings surprisingly few misclassifications.

Chapter 7: RSS-based Localization: Preliminaries

The remaining chapters of this thesis deal with localization based on received signal strengths (RSS). To evaluate the proposed approaches, both data from real-world experiments and simulations are used. This chapter starts with a description of the sensor network experiments and the used hardware. Subsequently, some preliminary experiments concerning signal propagation and attenuation effects of walls are analyzed. The gained insights are finally used to design meaningful simulations, and a direct comparison between the experiments and the simulations is presented.

Chapter 8: RSS-based Position Estimation

This chapter deals with the localization of stationary devices based on received signal strengths. In the considered problem, signal strengths of wireless broadcasts are analyzed to estimate distances to beacon nodes with known positions. This information is then used to localize nodes with unknown positions. In preparation for Chapter 9, which introduces force-directed methods for the RSS-based tracking of mobile devices, the focus of this chapter is on the design of adequate force-definitions for the representation of RSS-based distance estimates in force-directed localization approaches. To evaluate the achievable localization accuracies, a comparison with several other approaches for RSS-based position estimation is presented based on real-world measurements.

Chapter 9: Force-Directed Tracking

Force-directed methods are very popular in the areas of graph drawing and network embedding. Additionally, there also exist several force-directed approaches to find embeddings of wireless sensor networks based on connectivity information or received signal strengths. In this chapter, we study the use of force-directed methods for the tracking of pedestrians in indoor environments based on RSS measurements and movement information. Different possibilities to represent knowledge from movement models or inertial data by additional kinds of forces are discussed. Additionally, similarities and differences between the proposed force-directed tracking approach and extended Kalman filters are examined. The performance of the force-directed tracking approach is then evaluated based on real-world measurements and additional simulations.

Chapter 10: Indirect Network Localization

The automatic localization of wireless networks is one of the fundamental topics in research on wireless sensor networks. Several approaches exist for this problem. Some assume that positions of some anchor nodes are known, others use communication between static nodes to estimate pairwise distances and to compute an embedding based on this information.

In this chapter, we present an approach for the indirect localization of wireless sensor networks. The presented approach is an extension of the well-known MDS-MAP embedding algorithm [SRZF03], but instead of using communication between static nodes to estimate distances, we use a mobile node to estimate pairwise distances between static nodes indirectly. In contrast to most existing indirect approaches, we assume that neither information about anchor node positions is available, nor about the movement of the mobile node. Furthermore, we put special focus on the additional difficulties that arise in connection with the high uncertainties of distance estimates based on RSS-measurements in indoor environments. To evaluate the performance of the presented approach, we use simulations and real-world experiments in an indoor scenario. As an example application, the RSS-based tracking based on the computed node positions is studied.

Chapter 11: Conclusion

This chapter presents a brief summary of the main contributions of this thesis and concludes the thesis.

This work was supported by the German Research Foundation (DFG) within the Research Training Group GRK 1194 “Self-organizing Sensor-Actuator Networks”. Parts of this work have been published in [KVW08, VKW09, KVW10, SVG$^+$10, SVW11a, SVW11b, VSG$^+$12]. The chapters on wireless communication in the SINR model describe joint work with Bastian Katz and Dorothea Wagner, the chapter on connectivity-based boundary recognition describes joint work with Dennis Schieferdecker and Dorothea Wagner, and the chapters on RSS-based localization in wireless sensor networks describe joint work with Johannes Schmid, Tobias Gädeke, Klaus D. Müller-Glaser and Dorothea Wagner.

Chapter 2

Preliminaries

This chapter introduces basic notations, models, and techniques that will be assumed as known in the remainder of this thesis. The chapter starts with some terminology in the context of graphs and networks, followed by basics of algorithm analysis and complexity theory. Subsequently, some approaches to compute embeddings of graphs and networks are presented. In the area of information processing, the basic concepts of Kalman filters and extended Kalman filters are described. The chapter concludes with an overview of the modeling of radio propagation and interference in wireless networks and a brief survey on localization in wireless sensor networks.

2.1 Mathematical Foundations

2.1.1 Graphs and Networks

In this section, we introduce some basic terminology in the context of graphs and networks. The presented definitions follow the usual definitions of graphs and networks. For more detailed information on graph theory and related questions, we refer to the standard literature on this topic, e.g., [Jun08, Die10].

An *undirected graph* or *graph* is a pair $G = (V, E)$, where V is a finite set of *vertices* and $E \subseteq V \times V$ is a set of *unordered* pairs of vertices, called *edges*. An edge between $u \in V$ and $v \in V$ is denoted by $\{u, v\}$. Two vertices that are connected by an edge are called *adjacent* or *neighbors*. Pairwise non-adjacent vertices or edges are called *independent*. A set of vertices or edges is called *independent* if no two of its elements are adjacent. In contrast to an *undirected graph*, a *directed graph* or *digraph* $G = (V, A)$ consists of a finite set of vertices V and a set $A \subseteq V \times V$ of *ordered* pairs of vertices, called *arcs*. An arc from $u \in V$ to $v \in V$ is denoted by (u, v).

In many situations it is useful to assign to each edge $e = \{u, v\} \in E$ a *weight* $w(e)$. For example, in the context of sensor networks, this weight could be the distance between sensor nodes u and v, or the power required for a transmission between u and v. Such a graph G with weighted edges is referred to as a *weighted graph*.

A *path* P in a graph $G = (V, E)$ is a sequence $(v_1, v_2, \ldots, v_k)$ of distinct vertices $v_1, \ldots, v_k \in V$ such that $\{v_i, v_{i+1}\} \in E$ for $1 \leq i < k$. Similarly, a *directed path* in a digraph $G = (V, A)$ is a sequence $(v_1, v_2, \ldots, v_k)$ of distinct vertices $v_1, \ldots, v_k \in V$ such that $(v_i, v_{i+1}) \in A$ for $1 \leq i < k$. The number of edges of a path is called the *length* of the path. If $P = (v_1, v_2, \ldots, v_k)$ is a path and $k \geq 3$, then $C = (v_1, v_2, \ldots, v_k, v_1)$ is called a *cycle*. The *length* of a cycle is its number of edges.

Given an unweighted graph $G = (V, E)$, for each pair $(u, v) \in V \times V$, the *hop-distance* or *distance* between u and v is defined as the number of edges of a shortest path in G from u to v.

The set of *neighbors* of a vertex v in a graph $G = (V, E)$ is denoted by $N_G(v)$, or briefly by N_v. We will call the set of neighbors of a vertex v also the *neighborhood* of v. The *degree* $d_G(v) = d(v)$ of a vertex v is the number $|E(v)|$ of edges incident to

v. For undirected graphs, this is equal to the number of neighbors of v. The number $d(G) := \sum_{v \in V} d(v)/|V| = 2|E|/|V|$ is called the *average degree* of G.

We call a non-empty graph G *connected* if any two of its vertices are linked by a path in G. G is called *k-connected* (for $k \in \mathbb{N}$) if $|G| > k$ and any two of its vertices can be joined by k independent paths. A maximal connected subgraph of an undirected graph G is called a *connected component* of G.

Given a graph $G = (V, E)$, a d-dimensional *embedding* $p : V \to \mathrm{I\!R}^d$ assigns each vertex $v \in V$ coordinates $p(v)$ in a d-dimensional space.

A graph $G = (V, E)$ is called *unit disk graph (UDG)* if and only if there is an embedding of the nodes in the plane such that there exists an edge $\{u, v\}$ between node u and node v if and only if the Euclidean distance dist(u,v) between u and v is less than or equal to 1.

Given some parameter $d \in [0, 1]$, a graph $G = (V, E)$ is a *d-quasi unit disk graph (d-QUDG)* if and only if there is an embedding of the nodes in the plane such that dist(u,v) $\leq d \Rightarrow \{u, v\} \in E$ and $\mathrm{dist}(u, v) \geq 1 \Rightarrow \{u, v\} \notin E$.

2.1.2 Algorithms and Complexity

Time and space complexity. In the design of algorithms, one is often interested in worst-case and average-case running times. The *worst-case running time* specifies the longest running time for any input of size n. Knowing the worst-case running time gives a guarantee that an algorithm will never take any longer. This makes the worst-case running time particularly interesting from a theoretical point of view.

Analogously, the *average-case running time* specifies the running time for an average input for the considered problem. In order to compute average running times, a *probabilistic analysis* has to be applied. However, it often is not clear what constitutes an average input for a problem, thus the applicability of the average-case analysis is limited [CLRS09].

In addition to the time complexity of an algorithm, one is usually also interested in the *space complexity*, which is basically the number of memory cells that an algorithm needs. In many applications, there is a tradeoff between time complexity and space complexity, i.e., sometimes it is possible to save computation time by using additional memory.

Asymptotic notation. As usual in the theoretical analysis of algorithms, we present time and space complexities in terms of asymptotic bounds. To describe the asymptotical behavior, we use the common *Landau notation* [Knu76]: For two functions $f, g : \mathbb{N} \to \mathbb{R}$, $f \in O(g)$ if and only if there exist constants $c \in \mathbb{R}$ and $n_0 \in \mathbb{N}$ such that $f(n) \leq cg(n)$ for all $n \geq n_0$. Similarly, $f(n) \in \Omega(g)$ if and only if there exist constants $c \in \mathbb{R}_+$ and $n_0 \in \mathbb{N}$ such that $f(n) \geq cg(n)$ for all $n \geq n_0$. Finally, for $g : \mathbb{N} \to \mathbb{R}$, $\Theta(g)$ is given as $\Theta(g) = O(g) \cap \Omega(g)$.

The classes P, NP, and NPC. In this section, we give a very brief introduction to three complexity classes that are frequently used to categorize problems according to their hardness. The following presentation is on a rather high level of abstraction. For further information and more rigorous definitions we refer to [GJ79, Weg05, CLRS09].

The first class, P, consists of all problems that are solvable in polynomial-time, i.e., problems that can be solved in time $O(n^k)$, with n being the size of the input and k being some arbitrary but fixed constant. Such problems are usually considered to be *tractable*, and an algorithm whose worst-case time complexity is polynomially bounded in the size of the input is referred to as *efficient*. This does not necessarily mean that the problem really can be solved efficiently in practice, as even an "efficient" algorithm with time complexity $O(n^k)$ can only solve very small instances if k is large.

The complexity class NP consists of the problems that are *verifiable* in polynomial time. In this context, verifiable means that if we are given a solution to the problem, we can check in polynomial time whether the solution is correct.

It clearly holds that $\mathrm{P} \subseteq \mathrm{NP}$, but whether $\mathrm{P} = \mathrm{NP}$ or $\mathrm{P} \neq \mathrm{NP}$ is still unknown. In fact, this fundamental question constitutes one of the most important open problems in the field of theoretical computer science. The reason for this importance will become clear soon, but first we have to introduce some further terminology.

Most problems that are considered in computer science are *optimization problems*, in which each feasible solution has some value, e.g., the length of a computed path in a shortest-path algorithm, and the goal is to find a feasible solution with optimum value. In contrast to optimization problems, a *decision problem* is a problem that only allows the answers *yes* or *no*. Usually it is possible to reformulate optimization problems to decision problems by imposing a bound on the value that is optimized. For example, instead of asking for the shortest path, it is possible to ask whether there exists a path whose length is shorter than or equal to some given value z.

Apparently, an algorithm that solves an optimization problem can also be used to solve the corresponding decision problem, as, to decide whether the answer is *yes* or *no*, the value of an optimum solution simply has to be compared with the bound that is imposed by the corresponding decision problem. Thus, showing that a decision problem is *hard* also implies that the corresponding optimization problem is *hard*.

Let now Π_1 and Π_2 be two decision problems, and let I_1 be an instance of Π_1, where an *instance* I of a problem is simply a valid input to the problem. A *polynomial-time reduction* $\leq_P$ from Π_1 to Π_2 is a polynomial-time algorithm that computes for each instance I_1 of Π_1 an instance I_2 of Π_2 such that I_1 is a *yes*-instance of Π_1 if and only if I_2 is also a *yes*-instance of Π_2. This means, if there exists a polynomial-time reduction from Π_1 to Π_2 and if there exists a polynomial-time algorithm that solves Π_2, then the combination of both implies a polynomial-time algorithm that solves Π_1. Thus, in this case, Π_2 is at least as hard as Π_1.

A problem Π_1 is said to be *NP-hard* if every problem $\Pi_2 \in \mathrm{NP}$ can be reduced to Π_1 in polynomial time, i.e., $\Pi_2 \leq_P \Pi_1$ for all $\Pi_2 \in \mathrm{NP}$. Furthermore, a problem Π_1 is *NP-complete* if it is NP-hard and $\Pi_1 \in \mathrm{NP}$. According to this definition, every problem in NP can be reduced in polynomial time to any NP-complete problem, making NP-complete problems in some sense the “hardest” problems in NP. If only one NP-complete problem could be solved in polynomial time, then this would directly imply $\mathrm{P} = \mathrm{NP}$. The class that contains all NP-complete problems is denoted by NPC.

There is one problem left, though. One has to show for one first problem that all problems in NP are polynomial-time reducible to that problem. This breakthrough was achieved by Cook [Coo71] for the Sat problem, the problem to decide whether a given Boolean formula in Chomsky normal form (CNF) is satisfiable. Based on this major result, many other problems have been proven to be NP-complete over the last decades. An extensive list of many fundamental problems that are NP-complete can for example be found in the book on intractability by Garey and Johnson [GJ79].

2.2 Embedding of Graphs and Networks

2.2.1 Multidimensional Scaling

Multidimensional scaling (MDS) refers to a family of techniques, which use a matrix $\Delta \in \mathbb{R}^{n \times n}$ of dissimilarities or distances between n items to compute an embedding of the items in d-dimensional space, such that the distances in the computed embedding correspond as closely as possible to the given distances. This is, given a set V of items and distances δ_{uv} between all pairs $\{u, v\} \in V \times V$ of items, the objective of MDS is to find for each $v \in V$ a position $p_v \in \mathbb{R}^d$ such that $||p_u - p_v|| \approx \delta_{uv}$ for all pairs $\{u, v\} \in V \times V$. There exist many different types of MDS techniques, e.g., metric and nonmetric MDS, replicated MDS, weighted MDS, deterministic and probabilistic MDS. In this work, we use the *classical MDS* method, which has been proposed by Torgerson [Tor65]. In classical MDS, proximities are treated as distances in an Euclidean space. A detailed description of multidimensional scaling, including several variants of MDS, can be found in [CC01].

2.2.2 Force-Directed Embedding

Force-directed embedding approaches, also known as *spring embedders*, are very popular in the areas of graph drawing and network embedding. Originally, spring embedders have been used in the context of VLSI design [QB79]. The introduction to the area of graph drawing is usually accounted to an early work by Eades [Ead84]. The basic idea behind force-directed approaches is to model graph vertices by steel rings and to replace the edges by springs in order to form some kind of mechanical system. Starting from some initial layout, the vertices are then released so that the spring forces move the system to a state of minimal energy. One of the reasons why spring embedders are so popular is the enormous flexibility they offer. There are virtually no restrictions on the kinds of forces that can be used to model the springs. Accordingly, force-based methods can be adjusted to various applications and objectives.

As one example for a force-directed embedding method, we give in the following a brief overview of the well-known Fruchterman-Reingold spring embedder [FR91]. For further details on spring embedders and graph drawing in general, we refer to the book on graph drawing by Kaufmann and Wagner [KW01].

Fruchterman-Reingold algorithm. Given an undirected graph $G = (V, E)$, the algorithm of Fruchterman and Reingold tries to find an embedding of G such that nodes u and v which are connected, i.e., $\{u, v\} \in E$, are located close together, and unconnected nodes are separated from each other.

For every node $v \in V$, let $p_v = (x_v, y_v)$ be the position of node v in a given embedding $\mathcal{E}$. Additionally, let $d_{uv} = ||p_u - p_v||$ be the Euclidean distance between u and v in $\mathcal{E}$. The unit vector $\overrightarrow{p_u p_v}$ from p_u to p_v is given by

$$\overrightarrow{p_u p_v} = \frac{(p_v - p_u)}{||p_v - p_u||} \quad . \tag{2.1}$$

In order to achieve that connected nodes are pulled together, an attracting force f_{attr} is introduced for all pairs u, v of nodes, for which $\{u, v\} \in E$. In dependence of the positions p_u and p_v of u and v, this force is given by

$$f_{\text{attr}}(p_u, p_v) = \frac{||p_u - p_v||^2}{k} \cdot \overrightarrow{p_v p_u} \qquad \forall \{u, v\} \in E \quad . \tag{2.2}$$

Moreover, to prevent that nodes are pulled too close together, and to achieve that unconnected nodes are separated spatially, an additional repelling force f_{rep} is introduced between all pairs $u, v \in V$ of nodes:

$$f_{\text{rep}}(p_u, p_v) = \frac{k^2}{||p_u - p_v||} \cdot \overrightarrow{p_u p_v} \qquad \forall u, v \in V \tag{2.3}$$

In both force definitions, $k = C\sqrt{\frac{\text{area}}{|V|}}$ is a constant that controls the distances in the computed embedding in dependence of the area that is available for the drawing and the number of network nodes.

To compute an embedding of the network, the algorithm first is initialized with some initial embedding $\mathcal{E}_{\text{init}}$, e.g., by distributing the nodes randomly in the intended area. In general, this initial embedding is under tension, meaning that when summing up all forces that act on a node $u \in V$, the resulting force vector points in some direction. The Fruchterman-Reingold algorithm now iteratively tries to find an embedding that is in force equilibrium. In each step, for all nodes the forces acting on the node are summed up and the node is moved a small distance into the direction where the force vector points. This is repeated until either a certain number of iterations is reached or the lengths of all force vectors fall below some predefined threshold. To determine the movement distances, the Fruchterman-Reingold algorithm uses a cooling scheme. While the movement distances are rather large in the beginning, they become smaller over time, thus resulting in a convergence of the approach.

2.2.3 MDS-MAP

MDS-MAP [SRZF03] is a well-known technique to compute embeddings of wireless sensor networks. Unlike many other embedding approaches, MDS-MAP works even when no anchor nodes with known positions are available. The computation of the embedding is based on classical multidimensional scaling (MDS) [Tor65]. Accordingly, as classical MDS requires pairwise distances between all nodes, the first step of MDS-MAP consists in finding appropriate estimates for the node distances. Usually it is assumed that only connectivity information is given, i.e., the knowledge which of the nodes are able to communicate with each other. For such nodes that can communicate directly, a distance of 1 is assumed. Distances between the remaining pairs of nodes are approximated using shortest-path computations. If better distance estimates are available, e.g., by means of distance estimates based on received signal strengths, this information can easily be used instead of the hop-distances.

Given the estimates for the distances between all pairs of nodes, MDS-MAP computes an embedding using classical MDS. If additional knowledge about the positions of some anchor nodes is available, this information can be used in a subsequent step to map the relative coordinates computed by MDS to absolute coordinates. The time complexity of MDS-MAP is in $O(n^3)$, with n being the number of nodes.

Over the last years, several modifications and extensions of MDS-MAP have been proposed, which mostly aim at the improvement of the computed embeddings. An overview of some of these approaches will be given in Section 10.1.1.

2.3 Information Processing

2.3.1 Kalman Filter

The Kalman filter was proposed by R.E. Kalman [Kal60] as a recursive solution to the discrete-data linear filtering problem. Since then, the Kalman filter has become one of the most frequently used tools in the area of autonomous and assisted navigation. As we will use an extension of the Kalman filter later on as a reference approach for RSS-based tracking in wireless networks, we will give a very brief overview of the main concepts in this section. This presentation of the Kalman filter is based on [WB01] and [May79]. Additional information on the Kalman filter, as well as mathematical derivations, can be found in many textbooks on stochastic estimation, e.g., in [May79].

The Kalman filter can be used to estimate the state $x \in \mathbb{R}^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, \tag{2.4}$$

where A is an $n \times n$ matrix that relates the state x_{k-1} at time step $k-1$ to the current state x_k, in the absence of changes in direction and speed and neglecting process noise, B is an $n \times l$ matrix which relates optional control input $u \in \mathbb{R}^l$ to x_k, and w_k is a random variable that represents the process noise at time k. The state estimation x_k at time k is then improved using a measurement $z \in \mathbb{R}^m$ with

$$z_k = Hx_k + v_k, \tag{2.5}$$

where the $m \times n$ matrix H relates the state x_k to the measurement z_k and v_k represents the measurement noise. It is also assumed that the random variables w_k and v_k are independent of each other, normally distributed and zero-mean, i.e., $p(w) \sim N(0, Q)$ and $p(v) \sim N(0, R)$. The *process noise covariance* Q and *measurement noise covariance* R are assumed to be constant.

In order to estimate the state of the process, a form of feedback control is used. The Kalman filter consists of two groups of equations: *time update* and *measurement update* equations. The time update equations use the current state estimate $\hat{x}_{k-1}$ and error estimate P_{k-1} and project them forward in time to get *a priori* estimates $\hat{x}_k^-$ and P_k^- for the next time step:

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_{k-1} & (2.6) \\ P_k^- &= AP_{k-1}A^T + Q & (2.7) \end{aligned}$$

The measurement update is then used to incorporate new measurements z_k into the a priori estimate $\hat{x}_k^-$, in order to obtain an improved *a posteriori* estimate $\hat{x}_k$ and corresponding error covariance P_k. The used equations are as follows:

$$\begin{aligned} K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} & (2.8) \\ \hat{x}_k &= \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) & (2.9) \\ P_k &= (I - K_k H)P_k^- & (2.10) \end{aligned}$$

K_k denotes the *Kalman gain* of step k and defines the weighting of the predicted position estimate $\hat{x}_k^-$ relative to the measurement z_k for the *a posteriori* estimate $\hat{x}_k$. Roughly speaking, the higher the measurement uncertainty R_k, the stronger is the weighting of $\hat{x}_k^-$, and the higher the uncertainty of prediction $\hat{x}_k^-$, the stronger is the weighting of z_k. One can show that K_k is defined such that it minimizes the *a posteriori* error covariance. For a detailed derivation of this equations, we refer to [WB01].

One nice feature of the Kalman filter is its recursive nature. In each step, only the *a posteriori* estimates of the last time step are used to predict the new *a priori* estimates. This allows for very efficient implementations of the Kalman filter, as it does not require to keep all data in memory.

2.3.2 Extended Kalman Filter (EKF)

As mentioned in the previous section, the Kalman filter offers an optimal solution to the problem of estimating the state of a *linear* discrete-time controlled process. Unfortunately, processes of interest and measurement relationships are often not linear. To deal with such situations, an *extended Kalman filter* (EKF) can be used. The principle behind the EKF is that it linearizes about the current mean and covariance, using the partial derivates of the process and measurement functions. In contrast to the Kalman filter, the EKF thus can deal with situations where the process state x_k is given by a *non-linear* stochastic difference equation

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \quad , \tag{2.11}$$

in which the *non-linear* function f relates the state of time step $k-1$ to the state of the current time step, and where the state x_k at time k is related to the measurement z_k by an arbitrary non-linear function h

$$z_k \quad = \quad h(x_k, v_k) \quad . \tag{2.12}$$

Again, u_k represents the control input, w_k represents the process noise, and v_k represents the measurement noise.

The derivation of the time and measurement update equations of the extended Kalman filter is beyond the scope of this brief introduction. For further information, we refer to [May79, WB01].

Unfortunately, the EKF also has a significant disadvantage in comparison to the Kalman filter: The distributions of the random variables are no longer normal after application of the non-linear transformations. Accordingly, the EKF is only an *ad hoc* estimator, which approximates Bayes' rule by linearization [WB01]. Thus, unlike the Kalman filter for linear systems, the EKF cannot provide guarantees on the optimality of the computed estimates.

2.4 Radio Propagation and Interference

Radio propagation in wireless networks is influenced by a multitude of effects, most of which are highly unpredictable. The mutual cause of all random effects is the reflection, diffraction, and scattering of electromagnetic waves on all kinds of obstacles and surroundings. Multi-path propagation of radio waves results in interferences, which result in rapid fluctuations of received signal strengths even over very short distances of only few wavelengths. In consequence, modeling the radio channel is very difficult, and there exists a variety of radio models for all kinds of scenarios.

Most existing models for radio propagation are statistical models, which try to predict both the average decay of signal strength over distance and the variability of the signal strength over short distances. Models that mainly focus on the mean decay of signal strength over distance are usually called *large-scale propagation models*, those that consider the fluctuations over short distances or time intervals are called *small-scale propagation models*.

In addition to the statistical models, there also exists a variety of deterministic models for signal decay and interference. The use of such deterministic models is motivated by the fact that they allow for mathematical worst-case analyses and theoretical comparisons of algorithms that are influenced by radio propagation. In particular for worst-case analyses, which play an important role in the theoretical analysis of algorithms, statistical models are inconvenient. The worst-case usually would just mean that the signal does not reach the receiver with sufficient power to be decoded. For this reason, deterministic propagation models are frequently preferred in the algorithmic and theoretical computer science communities.

As radio propagation, and particularly the modeling of radio propagation, plays an important role for several problems that are considered in this thesis, we give in the following a brief overview of some possibilities to model radio propagation and interference. However, as the modeling of radio propagation is a very extensive and diverse field, this overview can only scratch the surface. For a more detailed introduction to statistical models for radio propagation and to the physical background of radio propagation, we refer to the book on wireless communication by Rappaport [Rap01]. Details on deterministic and graph-based interference models for the design of algorithms with worst-case guarantees can be found in the survey on algorithmic models for sensor networks by Schmid *et al.* [SW06] and in the book on algorithms for sensor and ad-hoc networks by Wagner and Wattenhofer [WW07].

2.4.1 Modeling of Path Loss

Path loss describes the signal attenuation of a radio signal on the way from its sender to the receiver. This includes both the signal decay with distance and attenuation effects at obstacles. Depending on the scenario, different path loss models can be used. The following brief overview is based on the book on wireless communication by Rappaport [Rap01].

Friis free space model. Let P_t denote the power transmitted by the sender and $P_r(d)$ denote the power with which the signal is received by the receiver in distance d from the receiver. In the simple case of free space propagation in the far-field of the transmitting antenna, the received power $P_r(d)$ in dependence of the distance d between sender and receiver can be described with *Friis free space equation* [Rap01], which in its simplest form states that

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2} \quad , \tag{2.13}$$

where G_t is the additional antenna gain of the transmitter, G_r is the antenna gain of the receiver, and λ is the wavelength of the radio wave in meters. Usually, the path loss PL is defined as the difference between the transmitted power and the received power, measured in decibels (dB). The path loss may or may not include the effects of antenna gains. If antenna gains are not considered, it is assumed that the antennas have unity gain. For Friis free space model, assuming unity gain for the antennas gives

$$PL_{\mathrm{dB}}(d) = 10 \log_{10} \frac{P_t}{P_r} = -20 \log_{10} \left(\frac{\lambda}{4\pi d} \right) \quad . \tag{2.14}$$

Even for line-of-sight communication over short distances, the received signal strength depends on a multitude of effects, e.g., the characteristic of the antennas. For this reason, a usual approach is to measure the received signal strength (RSS) $P_r(d_0)$ at some reference distance d_0, and then to relate the distance $P_r(d)$ for distances $d > d_0$ to $P_r(d_0)$. For free space propagation, this gives

$$P_r(d) = P_r(d_0)\left(\frac{d_0}{d}\right)^2 \quad . \tag{2.15}$$

Log-distance path loss model. In practical applications, one usually has to deal with additional effects that are caused by all kinds of obstacles. This is particularly true for indoor signal propagation, where walls typically result in significant signal attenuation. A famous model to subsume all additional effect that are caused by obstacles is the *log-distance path loss model.* The log-distance model is inspired by the observation that even under realistic conditions, the average received signal strength in dependence of the distance between sender and receiver can be described very well by a power law, i.e.,

$$P_r(d) \propto \frac{P_t}{d^\alpha} \quad . \tag{2.16}$$

The constant α in Equation 2.16 is the so-called *path loss exponent*, which defines how fast the signal strength decreases with distance. For unobstructed outdoor communication, α is usually assumed to be about 2. In buildings, walls and obstacles result in additional attenuation, which usually results in higher values of α in the range of 3 to 5. However, walls can also have positive effects on the received signal strengths. For line-of-sight transmissions in buildings, even α-values lower than 2 are possible. For example, [Rap01] states that for this situation typical values of α are in the range of 1.6 to 1.8.

Using path loss exponent α and the path loss $PL_{\mathrm{dB}}(d_0)$ at some reference distance d_0, the path loss in the log-distance path loss model is given as

$$PL_{\mathrm{dB}}(d) = PL_{\mathrm{dB}}(d_0) + 10\alpha\log_{10}\left(\frac{d}{d_0}\right) \quad . \tag{2.17}$$

Log-normal shadowing. Equation 2.17 only considers the average large-scale decay of signal strength. However, received signal strengths also fluctuate very intensely over very short distances, caused by multi-path propagations and self-interferences. It has been shown in various measurements that received signal strengths for a given sender-receiver-distance are normally distributed around the average value when measured in dB [Ber87]. To model this, Equation 2.17 can be extended by a zero-mean Gaussian random variable X_σ, giving

$$PL_{\mathrm{dB}}(d) = PL_{\mathrm{dB}}(d_0) + 10\alpha\log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \quad . \tag{2.18}$$

Modeling of indoor path loss. Radio propagation in indoor environments is characterized by a particularly high degree of signal strength fluctuation. Various kinds of obstacles result in an enormous amount of reflection, diffraction, and scattering of radio waves. This is aggravated by the fact that different kinds of materials have very different effects on signal propagation.

To model the characteristics of indoor propagation, various models have been proposed. Very popular is the use of the log-distance model, usually extended by the Gaussian random variable X_σ of the log-normal shadowing (cf. Equation 2.18). Appropriate values for the path loss exponent α and the variance of X_σ have been determined in experimental studies for various types of buildings. An early study, considering several scenarios and radio signals of different frequencies, is for example given in [ARY94].

The log-distance model subsumes various kinds of effects in two quantities, the path loss exponent α and the standard deviation of the random variable X_σ, which models the fluctuations of the signal strength for a given distance. However, especially in indoor scenarios, there are several effects on received signal strengths that can be predicted to some degree. For example, the penetration of walls and floors causes significant signal attenuation. These so-called *partition losses* have been analyzed for various kinds of obstacles and materials, e.g., in [Rap91]. A study with focus on the attenuation of floors was for example presented in [SR92]. In [SR92], it was also shown that the standard deviation of the noise in the log-distance path loss model can be significantly reduced by modeling effects such as the penetration of floors separately. In the presented *attenuation factor model*, the path loss was modeled as

$$PL_{\mathrm{dB}}(d) = PL_{\mathrm{dB}}(d_0) + 10\,\alpha^{\mathrm{SF}} \log_{10}\left(\frac{d}{d_0}\right) + \gamma^{\mathrm{FAF}} \quad , \tag{2.19}$$

where α^{SF} denotes the path loss exponent for measurements at the same floor and γ^{FAF} is an additional floor attenuation factor (measured in dB), which models the signal attenuation that is expected in consequence of the type and number of penetrated floors.

2.4.2 Modeling of Communication and Interference

Whether or not a specific transmission is successful depends on several factors. In particular, the signal that reaches the receiver must be strong enough to be decoded, and there must not be too much interference from concurrent transmission. In order to analyze algorithms that are influenced by signal propagation and interference, well defined conditions for successful reception are necessary. This section gives a brief overview of models that are frequently used to describe communication in wireless networks.

Unit disk graph models. Wireless receivers usually have a limited *sensitivity*, meaning that they are only able to decode signals that exceed some hardware-dependent threshold. As the signal strength of wireless transmissions decreases with distance, this means that signals that were transmitted with some fixed power can only be received up to some distance $d_{\max}$. Beyond that distance, the signal is too weak to be decoded. In analogy to graph-theoretic unit disk graphs (cf. Section 2.1.1), this simple observation is modeled in *unit disk graph communication models* (UDG models), which assume that two wireless nodes can communicate with each other if and only if their distance does not exceed $d_{\max}$.

Quasi unit disk graph models. Due to multi-path propagation and attenuation effects, received signal strengths are subject to high fluctuations. In consequence, knowing the distance between a sender-receiver pair is not sufficient to decide whether a transmission is successful or not. However, for small distances one usually can be pretty sure that the received signal strength is sufficiently high for successful reception, and for very large distances it is also relatively certain that the transmission has to fail. This is modeled in *quasi unit disk graph communication models* (QUDG models) by assuming that all transmissions up to some distance d_1 are possible, all transmissions exceeding a sender-receiver distance d_2 definitely fail, and for all other transmissions there is only a certain possibility that they succeed.

Graph-based interference models. Even when the signal strength at the receiver exceeds the receiver sensitivity threshold, this is no guarantee that the signal can be decoded. Additionally, the received signal strength has to exceed both the background noise and the interferences that are caused by concurrent transmission of nearby senders by a certain amount. One possibility to model this additional condition are *conflict graph models*. A *conflict graph* defines for each pair of links, i.e., sender-receiver pairs that could communicate with each other if they would not have to share the communication channel, whether it is possible that both links are active concurrently without failures. Several models to decide for pairs of links whether concurrent transmission is possible or not have been proposed. A famous example is the *protocol model* [GK00], which assumes two conditions for successful transmission from some sender s to a receiver r: First, the distance between s and r must be smaller than the maximum transmission range R. Second, the distance between any other sender s_2, which is active concurrently, and r must exceed $(1+\Delta)R$ for an appropriate constant Δ. An overview on further graph-based interference models is presented in [WW07].

SINR model. Graph-based interference models have two fundamental problems: they oversimplify the distance dependence of received signal strengths and they are not able to model that interferences from several concurrent transmissions can sum up. As a result, many effects that occur in reality are not reproducible by graph-based interference models [MWW06]. These problems are tackled by the physically motivated *signal-to-interference-plus-noise-ratio model* (SINR model), which assume that a transmission is successful if and only if the ratio of the received signal strength to the sum of all interferences plus the background noise exceeds some hardware-dependent SINR threshold β. In its most general form, the SINR model makes no assumptions concerning the signal decay. Instead, it allows arbitrary link gains for all pairs of senders and receivers. A formal introduction of the SINR model will be given in Section 3.2.

Geometric SINR$_\mathbf{G}$ model. The SINR model makes no assumptions concerning signal propagation. This makes it hard to give worst-case guarantees for algorithms that are based on the SINR model. To alleviate this problem, frequently an extended version of the model is used, the *physical model* [GK00], also called *geometric SINR$_G$ model* [GOW07]. In the SINR$_\mathrm{G}$ model, it is additionally assumed that the signal decay with distance can be fully described by a simple power law. In analogy to the log-distance path loss model (cf. Section 2.4.1), the SINR$_\mathrm{G}$ model uses a path loss exponent α to define how fast the signal decays with distance. Based on the assumptions of the SINR$_\mathrm{G}$ model, several approximation algorithms for the scheduling of wireless transmissions have been proposed recently. A formal introduction of the geometric SINR$_\mathrm{G}$ model will be given in Section 4.2.

Generalized physical model. Like the unit disk graph model, the SINR$_\mathrm{G}$ model is unable to model the random nature of received signal strengths. Each distance is assigned a well-defined signal strength. To allow for fluctuations in signal strength while preserving the possibility to provide worst-case guarantees, a generalization of the SINR$_\mathrm{G}$ model has been introduced by Moscibroda *et al.* in [MWZ06]. The proposed *generalized physical model* assumes that the signal strength P_r that reaches a node r from some sender s can deviate up to a constant factor θ from the theoretical value as given by the SINR$_\mathrm{G}$ model, i.e.,

$$\frac{1}{\theta}\frac{P_t}{d(s,r)^\alpha} \leq P_r \leq \theta\frac{P_t}{d(s,r)^\alpha} \quad , \tag{2.20}$$

with P_t being the used transmission power, $d(s,r)$ being the distance between sender s and receiver r, and α being the corresponding path loss exponent.

2.5 Localization in Wireless Sensor Networks

Location discovery is a very ample problem, which has been studied in various different contexts including robotics, ad hoc networks, military, aviation and space travel. Today, most outdoor localization problems have been solved with the upcoming of global navigation satellite systems such as GPS [HWLC97] or Galileo [HWBL$^+$08]. However, especially for sensor networks consisting of many devices and for indoor scenarios, cheap and reliable localization still poses a considerable challenge.

In this section, we give a brief overview on research concerning different aspects of localization in wireless sensor networks. The different approaches differ widely in their requirements and achievable localization accuracies. Like it is often the case for wireless sensor networks, there is no all-round approach that works optimally for all situations. Instead, the best localization approach usually strongly depends on the considered application.

The intention of this brief survey is to give a quick overview on different areas of research in the context of localization in wireless networks and to show how the research presented in this work fits into a bigger context. For topics that are covered in this thesis, additional information on related work will be given in the corresponding chapters. For more detailed surveys on localization in wireless networks, we refer to [MFA07, Bou09, WGD10].

2.5.1 Estimation of Distances and Angles

Several approaches for the communication-based estimation of distances and angles in wireless networks have been presented. They differ widely both in achievable accuracies and in the requirements on the hardware, and it strongly depends on the considered application which of the approaches is suited best.

Connectivity: Connectivity information, i.e., the knowledge which nodes can communicate with each other, offers a very simple and cheap possibility to estimate distances between wireless nodes. If two nodes are able to communicate, one can assume that they are within one communication range of each other. Not surprisingly, this distance estimate is relatively inaccurate.

Received Signal Strength (RSS): Most modern communication hardware allows to measure the signal strength of received wireless packets using a *received signal strength indication* (RSSI). As the signal strength on average decreases with the distance between sender and receiver, one can use this information for a more accurate distance estimation than it would be possible using only connectivity information. In reality, the received signal strength is additionally influenced by a multitude of effects, e.g., obstacles, interferences, and antenna characteristics. Accordingly, RSS-based distance estimation is rather inaccurate, with an average error in the range of some meters [SHS01]. However, as RSS-based distance estimation is available on many modern devices without additional cost, it offers a viable possibility to estimate distances in applications that do not require high accuracies. In Section 7.3, we present a study of RSS-based distance estimation based on experiments in a wireless sensor network.

Time of Arrival (ToA): Another way to estimate distances between nodes is to measure the time a signal needs to propagate from one node to another. Using the propagation speed of the signal in the considered medium, one can easily determine the distance between the nodes [HWLC97]. Depending on the used kind of signal, extremely precise clocks are necessary, and it is also necessary that the clocks of the nodes are precisely synchronized.

Time Difference of Arrival (TDoA): Similar to the time of arrival method, one can use time differences to estimate distances between nodes. Basically, there are two possibilities: One can either use a single signal from one node and measure the differences of the arrival times at different other nodes, or one node sends signals with different propagation velocities at the same time, e.g., radio and ultrasound like in the Cricket indoor location system [PCB00] and the SpiderBat ranging platform [OSW10, OSW11], and the difference of the arrival time is then used to estimate the distance. Using TDoA approaches, accuracies in the range of few centimeters are possible [SHS01, OSW11].

Angle of Arrival (AoA): Using antenna arrays or directive antennas, it is also possible to estimate the angle of arrival of wireless signals [NN03]. For example, one can measure arrival times at different receivers and use this information in combination with knowledge about the positions of the receivers to estimate the angle to the sender in relation to reference points or an electronic compass [PMBT01]. A recent approach for angle of arrival measurement is the SpiderBat system [OSW10, OSW11], where an array of ultrasound receivers is used to measure angles with an accuracy in the range of few centimeters.

For a more detailed overview on possibilities for distance and angle estimation, we refer to [Bou09].

2.5.2 Position Estimation

The approaches that are described in Section 2.5.1 allow it to estimate distances and angles between nodes. In this section, we give a brief overview on methods that are frequently used to estimate node positions based on this kind of information.

Center of Reference Nodes: A very simple approach to estimate the position of a wireless node is to put it in the center of the reference nodes that are within communication range [HHB^{+}03]. From a computational point of view, this approach is very efficient, but, not surprisingly, the achievable accuracy is rather low and many reference nodes are required to achieve reasonable localization results.

Trilateration: If distances to three or more reference nodes with known positions (also called *anchor nodes*) are available, the position can be computed as the intersection of circles around the reference nodes with radii that correspond to the respective distances. In reality, usually both the distance estimates and the positions of the reference nodes are inaccurate. To estimate the node position despite this inaccuracies, one can solve the (possibly over-determined) system of equations that is induced by the distance estimates using a least squares approach [TBC11].

Triangulation: In contrast to trilateration, triangulation allows to estimate the position of a device based on knowledge about angles to at least three reference nodes with known positions. The position is then computed using simple trigonometrical relationships [NN03].

Probabilistic Approaches: Probabilistic approaches [SR04b, PS07] aim at estimating the probability that the device is located at certain positions in space. Instead of computing a single position estimate, probabilistic approaches consider many points in space and compute the likelihood, given the distance estimates and their uncertainties, that the device is at these positions. If the application requires a single position estimate, for example the most likely position can be returned.

Fingerprinting: Fingerprinting works by directly associating location-dependent characteristics, such as received signal strengths, with each location. The corresponding characteristics have to be collected in an initial training phase and are stored in a fingerprint database [YYAS03]. To locate a mobile device, the received signal strengths at the unknown location are compared to the entries in the database, and the location whose RSS fingerprint is closest to the signal strengths that were received is returned as position estimate. The achievable accuracy depends on the number of locations that are sampled in the fingerprint database. If the database contains data from many positions, fingerprinting offers good accuracies compared to other RSS-based approaches. However, the required initial training phase is also relatively complex. Recent examples for fingerprinting are presented in [AC09, PCC+10, BOGVB10].

Global Navigation Satellite Systems (GNNS): In outdoor scenarios, global navigation satellite systems [HWBL+08] such as the global positioning system (GPS) offer an efficient possibility to estimate node positions within an accuracy of few meters. Devices with GPS receivers can use ToA information to estimate their distances to four or more GPS satellites with known positions. The position of the device, i.e., its latitude, longitude and altitude, can then be computed using trilateration. While GPS probably is the optimal solution for many scenarios, this technique also has some disadvantages, which disallow its use in several situations: As special GPS receivers are necessary, cost, size, and energy consumption of the sensor nodes are increased, which is particularly undesirable for sensor networks. Additionally, GPS is only applicable in scenarios where the nodes have unobstructed sight to the satellites, making this technique unavailable for indoor localization.

2.5.3 Tracking

In the tracking problem in wireless sensor networks, the position of a mobile device has to be estimated and followed over some period of time. In addition to the information that is available in the static position estimation problem, in the tracking problem one often has additional knowledge from movement models or inertial sensors, which allows an improved localization. Tracking has been studied intensely in various fields of science, using all kinds of means to estimate distances and positions. In the area of wireless sensor networks, the position estimation is mostly based on connectivity information or measurements of received signal strengths. The following approaches are frequently used for tracking in wireless networks:

Static Position Estimation: The simplest possibility to realize tracking in wireless networks is to use one of the approaches for position estimation that are described in Section 2.5.2 repeatedly. Especially fingerprinting is frequently used without using additional information from movement models or inertial data. However, if movement knowledge is available, utilizing this information can result in significantly improved localization results.

Kalman Filters: Among the most frequently used approaches for tracking in wireless networks are variants of Kalman filters such as extended Kalman filters (EKFs) [WB01] or unscented Kalman filters (UKFs) [JUDW95]. Kalman filters themselves are only applicable to linear systems. Accordingly, they cannot be applied directly to the considered localization and tracking problems, as these are non-linear due to the involved distance estimates. EKFs and UKFs are extensions of standard Kalman filters, which approximate the non-linear system by a linear system using either analytical (EKF) or statistical (UKF) techniques. More detailed information on Kalman filters and EKFs is presented in Section 2.3. Examples for the use of EKFs and UKFs for tracking in wireless networks can be found in [PW09, SKL10, SBN$^+$11].

Particle Filters: Particle filters are simulation-based model estimation techniques, which are usually used to estimate Bayesian models. In particle filters, probability distributions are represented by a set of randomly chosen weighted samples. If the number of samples is sufficiently high, particle filters can be designed such that they approach the optimal Bayesian estimate. In particular, particle filters are not restricted to the modeling of Gaussian distributions, but they are fully capable of modeling multimodal probability distributions. However, to achieve this improved estimation, particle filters also have significantly higher computational costs than for example EKFs. Like EKFs, particle filters are frequently used for the tracking of mobile devices in wireless networks, e.g., in [MNR$^+$06, WLS$^+$07].

2.5.4 Network Localization

Most of the approaches in the previous sections assume the availability of reference nodes with known positions. However, in many possible applications of wireless sensor networks, especially those that involve hundreds or thousands of sensor nodes, it cannot be expected that all positions are initialized manually or that all devices have the capabilities to determine their own position autonomously using techniques such as GPS. As knowledge about node positions is very crucial for most applications in sensor networks, various approaches have been proposed to infer absolute or relative node positions from information about connectivity or from the locations of some reference nodes with known positions.

Boundary Recognition: For some applications, it is sufficient to know whether a node lies within the network or near a hole or boundary of the network. This knowledge can then for instance be used to aid efficient routing within the network or for load balancing near holes. Possibly applications of boundary detection and related work are described in Chapter 6, which deals explicitly with boundary recognition based on connectivity information.

Network Localization with Anchor Nodes: In some scenarios, it is assumed that a subset of the wireless devices, so-called *anchor nodes*, know their position. Several approaches have been proposed to use this information to infer the positions of nodes with unknown positions. For example, in the Ad Hoc Positioning System (APS) [NN01] nodes determine their hop-distances to anchor nodes with known positions. Based on the hop distances, trilateration is used to compute the position estimates. The Recursive Position Estimation (RPE) [ACZ01] approach works similar, but additionally regular nodes become anchor nodes as soon as they have an own position estimate. In [SR04b], a mobile beacon node with known position is used to initialize the positions of static sensor nodes using a probabilistic approach.

Anchor-free Network Localization: There also exist approaches that allow the estimation of node positions in wireless sensor networks without any reference nodes. A famous example for such *anchor-free* approaches is MDS-MAP [SRZF03], where distances between nodes are estimated by the corresponding hop distances and multidimensional scaling is then used to approximate the underlying embedding of the network. An extension of MDS-MAP has been presented in [SR04a], where parts of the network are embedded independently from each other and the single maps are finally combined to an embedding of the whole network. Approaches for the force-directed embedding of wireless sensor networks are for example presented in [EFI$^+$10, CKKK10].

Simultaneous Localization and Mapping (SLAM): *Simultaneous localization and mapping* (SLAM) describes a class of approaches, where a mobile device with unknown position is used to map the environment. Usually it is assumed that either information about the movement of the mobile device is available, for example by means of odometry data, or knowledge about the positions of some reference points. In SLAM, the position of the mobile device and the positions of the reference points are then estimated alternately. An example for the use of SLAM in sensor networks is given in [HMS01], where mobile robots are localized simultaneously with a sensor network.

Part I

Communication

Chapter 3

Dynamic Power Control

In this chapter, we present a new approach for the dynamic computation of optimum transmission powers. Unlike existing methods for the power control problem, the presented data structure is optimized for the efficient prediction of power changes when there are only small changes in the network. Three variants of the data structure are discussed, which differ in the supported operations and the corresponding complexities.

3.1 Introduction

Efficient communication is one of the very fundamental topics in the area of wireless networks. When nearby nodes communicate in the same frequency range, they have to share the same medium. The consequence is interference between concurrent transmissions, which can result in worse transmission rates or even in the failure of transmissions. The amount of interference is influenced by many factors, e.g., distances to interfering senders, environmental conditions, and used transmission powers. In the context of wireless communication, it is usually assumed that, for a transmission to be successful, the received signal strength has to exceed the interfering signals at the receiver by a certain amount. This leads to interesting optimization problems: By increasing the transmission power, a sender can increase the likelihood that the receiver is able to decode the signal. However, this also increases the interference at other receivers that are nearby, thus potentially causing the corresponding transmissions to fail. Accordingly, in order to ensure that all transmissions are successful, some tradeoff has to be found.

Finding such a tradeoff is the goal of the power control problem: Given a set of wireless transmission requests, one has to compute minimum transmission powers such that all transmissions are successful. Power control is a central building block in many algorithms that deal with communication in wireless networks. For example, in the area of topology control, power control is used to limit the transmission range of senders with the goal to avoid interferences in the network [MWZ06]. Similarly, power control is frequently used in combination with scheduling approaches for wireless transmissions. By reducing the transmission powers for transmissions over short distances, one can avoid unnecessary interference and thus increase the overall throughput in the network.

3.1.1 Related Work

Solving the power control problem, i.e., finding minimum transmission powers for a set of wireless transmissions, is a common subproblem of many algorithms that deal with wireless communication. Accordingly, the problem has been studied for a long time, and different solution approaches have been proposed.

Originally, power control has been studied in the context of channelized cellular systems [Zan92b, Zan92a, FM93, GVG94]. The focus of power control in this early work was the minimization of the outage probability due to co-channel interference. Power control was used to defer single transmissions in order to achieve a certain signal-to-noise ratio.

In an early work by Zander [Zan92b], which was based on prior work for satellite communication by Aein [Aei73], it was shown that under the assumption that the noise at the receiver can be neglected, the largest possible signal-to-interference ratio (SIR) β^* for a set of transmissions can be computed using an eigenvalue decomposition. In this case, β^* is related to spectral properties of a normalized variant Z of the underlying *link gain matrix* (the link gain matrix defines for each pair of nodes how much the signal is attenuated on the way from the sender to the receiver). With λ^* being the largest real eigenvalue of Z, β^* is given as

$$\beta^* = \frac{1}{\lambda^* - 1} \quad . \tag{3.1}$$

The power vector P^* that achieves β^* is then given by the eigenvector of Z corresponding to the eigenvalue λ^*. This power control approach was then used in several heuristics to find a maximal subset $S \subseteq \mathcal{S}$ of a set $\mathcal{S}$ of transmissions, such that all transmissions of S concurrently exceed a given SIR β [Zan92b, Zan92a, FM93].

Another possibility to tackle the power control problem is to express it by a linear program (LP) and then to use an LP solver to compute optimum transmission powers [WCRP05]. This approach also works when background noise is considered. The corresponding formulation of the power control problem as a linear program is given in Section 3.3.

A further approach, which is used by most recent heuristics for the combined problem of scheduling and power control, is to start with small transmission powers and then to iteratively increase the powers until either all transmissions become feasible, or one of the senders exceeds its maximum transmission power. This can be done in such a way that the resulting transmission powers converge to the optimum [FM93, Yat95, EE04]. This iterative power control approach can also easily be used in a distributed setting. In order to achieve faster convergence, one usually uses in each iteration transmission powers that are by a slight factor α (e.g., $\alpha = 1.05$) higher than the powers that are necessary to cope with the current interference. With this optimization, the approach usually converges after few iterations [EE04]. However, this also results in transmission powers that are slightly higher than the optimum powers. If applied in a centralized setting, an iteration of the algorithm takes $O(n^2)$ time, thus a nearly optimal power assignment can also be found in $O(n^2)$ time.

Most related work on power control considers the joint problem of scheduling with power control. Further information on related work in this area is given in Chapter 4, which deals with energy-efficient scheduling with power control.

3.1.2 Overview

If one has to compute a power assignment only once for a given set of links, or if the powers are computed in a distributed way, the existing iterative power control approach is very efficient. However, in the combined problem of scheduling and power control, such power assignments have to be computed over and over until the final set of links and powers is found. Between the computations, there usually are only very small changes to the set of active transmissions, e.g., single transmissions are activated or deactivated. Additionally, as we will see later, one can achieve very good schedules by basing the decision which transmissions to activate on the effects the decision has on the transmission powers of other active links. To utilize this observation, a scheduling algorithm has to frequently predict how transmission powers change when additional transmissions are activated.

In this work, we examine how such incremental changes can be computed more efficiently than by just recomputing all transmission powers from scratch. For this purpose, we present in Section 3.4 a data structure that is optimized for the incremental activation of links and the efficient prediction of power changes when further transmissions are activated.

In many real-world applications, it might also be necessary to react efficiently to various other kinds of changes in the network. For example, if nodes move, link gains between senders and receivers might change. Additionally, it may happen that new nodes enter the network, which makes additional updates necessary that can not easily be performed using the aforementioned data structure. For this reason, we demonstrate in Section 3.5 how the data structure can be extended to allow additional operations such as the deactivation of links or the update of transmission powers when the ambient noise changes.

In order to enable the efficient prediction of power changes for the activation or deactivation of single transmissions, one has to maintain some information even for transmissions that are not active. This results in additional computational effort, which is more than compensated when the prediction functionality is used frequently. However, if such predictions are rare or completely unnecessary, one does not have to maintain this additional information. In Section 3.6, we show how the presented data structure can be adjusted to this situation, and which improvements this enables for the time complexities of some operations.

3.2 Problem Definition

Scenario and terminology. We consider a wireless network $\mathcal{N}$. In this network, we are given a set $\{l_1, l_2, \dots, l_n\}$ of links between nodes of the network. Each *link* $l_i = (s_i, r_i)$ is defined by a *sender* s_i and a *receiver* r_i. It is assumed that the distance between s_i and r_i is sufficiently small to allow communication between both nodes. A link l_i is called *active* if s_i transmits data to r_i. We denote the *transmission power* that s_i uses by P_i. In the following, we assume that there exists an upper bound $P_{\max}$ on the available transmission powers, i.e., $0 \leq P_i \leq P_{\max}$ for all senders s_i. It further holds that $P_i > 0$ if and only if link l_i is active. If s_i transmits with power P_i, then the signal is received by r_i with lower power P_{ii}. This weakening of signal strength is caused by path losses on the way from the sender to the receiver, e.g., by free-space path losses or by attenuation of the signal due to walls and obstacles (for further information on path losses see Section 2.4). The *link gain* γ_{ii} between sender s_i and receiver r_i subsumes these losses and defines how much the signal strength decreases on the way from sender s_i to receiver r_i. When s_i sends with power P_i, then the signal is received by r_i with power $P_{ii} = \gamma_{ii} P_i$, where γ_{ii} depends on the distance between s_i and r_i, as well as on environmental conditions and obstacles. If s_i transmits data to r_i, then the signal of s_i also reaches other receivers r_j that are nearby. In this case, the signal causes *interference* at the affected receivers. The amount of interference depends on the link gain between sender s_i and receiver r_j and on the transmission power P_i of s_i, i.e., $P_{ij} = \gamma_{ij} P_i$. In addition to this interference due to concurrent transmissions, every receiver r_i experiences some *background noise* η_i. In the following, tuples $t_i = (s_i, r_i, P_i)$ are also called *transmissions*. Each transmission is defined by a sender-receiver-pair (s_i, r_i) and the transmission power P_i that s_i uses.

Feasibility of transmissions. In order for a receiver r_i to be able to decode the signal of a transmission from s_i, the signal strength at the receiver must be sufficiently stronger than the sum of the interferences plus the ambient noise. This condition is expressed by the so-called *signal-to-interference-plus-noise ratio* (SINR) condition:

$$\mathrm{SINR}(r_i) = \frac{P_i \gamma_{ii}}{\sum_{j \neq i} P_j \gamma_{ji} + \eta_i} \geq \beta_i \tag{3.2}$$

The SINR condition states that, for a transmission to be *feasible*, the ratio of the signal to the sum over all interferences plus background noise has to exceed some threshold $\beta_i > 1$. Analogously, a set $\{t_1, t_2, \ldots, t_k\}$ of transmissions is said to be feasible if SINR condition (3.2) is fulfilled for every receiver r_i, $1 \leq i \leq k$. The resulting interference model, the *SINR model*, is physically motivated and it is assumed that it reflects reality reasonably well [WW07]. Further information on the SINR model and its relation to other interference models can be found in Section 2.4.1.

Power control. In the *power control problem*, one is given a set L of links, and link gains γ_{ij} for all pairs (s_i, r_j) of senders s_i and receivers r_j. If it is possible to find for all links $l_i \in L$ transmission powers $P_i \in (0, P_{\max}]$ such that the resulting transmission set is feasible, the task is to compute such powers. Otherwise, it has to be indicated that the link set is infeasible. In order to extend the lifetime of the network, the optimization goal is to find *minimum* such transmission powers. As we will see in Section 3.3, the notion of minimum transmission powers is well-defined and independent of the metric that is applied.

3.3 Mathematical Background

Given n active transmissions, the necessary condition for successful transmission is given by the SINR inequalities (3.2). For every link l_i, one can easily rearrange the corresponding inequality to obtain a sufficient condition for the transmission power P_i of s_i:

$$P_i \quad \geq \quad \sum_{j \neq i} P_j \frac{\beta_i \gamma_{ji}}{\gamma_{ii}} + \eta_i \frac{\beta_i}{\gamma_{ii}} \quad , 1 \leq i \leq n \quad . \tag{3.3}$$

This condition can directly be transferred into an LP for the minimization of the sum of transmission powers:

$$\begin{aligned} \text{min!} \quad & \sum_{1 \leq i \leq n} P_i \\ \text{s.t.} \quad & P_i \geq \sum_{j \neq i} P_j \frac{\beta_i \gamma_{ji}}{\gamma_{ii}} + \eta_i \frac{\beta_i}{\gamma_{ii}} \quad \text{for } 1 \leq i \leq n \\ & P_i \geq 0 \quad \text{and} \quad P_i \leq P_{\max} \quad \text{for } 1 \leq i \leq n \ . \end{aligned} \tag{3.4}$$

In this form, we also observe that minimizing the sum of powers yields the same result as minimizing any function $f(P_1, \ldots, P_n)$ that is (not necessarily strongly) monotonically increasing in the P_i. Such functions include the maximum, the minimum, or any norm. Explicitly, if the problem is feasible, for any such function f, the optimum solution $p^\star := (P_1^\star, \ldots, P_n^\star)^T$ is given by setting $P_i^\star$ to the minimum value of P_i in any feasible solution $p = (P_1, \ldots, P_n)^T$. This is well-defined since the value of any P_i in any feasible solution is bounded from below and the set of feasible solutions is closed. Assume $P^\star$ is not feasible (if it is, it is obviously optimal), i.e., assume that Equation (3.4) is violated for some i. By construction, there is a feasible solution $p^i := (P_1^i, \ldots, P_n^i)^T$ with $P_i^\star := P_i^i$. For all $j \neq i$, we have $P_j^\star \leq P_j^i$ by construction and hence Equation (3.4) holds for i, in contradiction with the assumption.

Next, we show that for the optimal solution $p^\star$, Equation (3.4) is tight, since otherwise, some $P_i^\star$ could be reduced without losing feasibility. That is, finding powers minimizing any monotonically increasing function f subject to $0 \leq P_i \leq P_{\max}$ and

$$P_i \quad = \quad \sum_{j \neq i} P_j \frac{\beta_i \gamma_{ji}}{\gamma_{ii}} + \eta_i \frac{\beta_i}{\gamma_{ii}} \quad , 1 \leq i \leq n \tag{3.5}$$

is equivalent to the problem stated above. Note that in this work we assume non-zero noise, i.e., $\eta_i > 0$ for all links l_i. Writing Equation (3.5) as

$$\mathcal{A} \cdot p \quad = \quad b \tag{3.6}$$

with

$$\mathcal{A} = \begin{pmatrix} 1 & -\frac{\beta_1\gamma_{21}}{\gamma_{11}} & \dots & -\frac{\beta_1\gamma_{n1}}{\gamma_{11}} \\ -\frac{\beta_2\gamma_{12}}{\gamma_{22}} & 1 & \dots & -\frac{\beta_2\gamma_{n2}}{\gamma_{22}} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\beta_n\gamma_{1n}}{\gamma_{nn}} & -\frac{\beta_n\gamma_{2n}}{\gamma_{nn}} & \dots & 1 \end{pmatrix}, \quad p = \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{pmatrix}, \quad b = \begin{pmatrix} \beta_1\eta_1/\gamma_{11} \\ \beta_2\eta_2/\gamma_{22} \\ \vdots \\ \beta_n\eta_n/\gamma_{nn} \end{pmatrix},$$

we obtain the following result, which justifies dropping the optimization criterion:

Proposition. If there is any feasible solution to the power control problem, Equation (3.6) has a unique solution, which is the optimum $p^\star$.

Proof. Assume that the power control problem has a feasible solution. Then $p^\star$ is an optimum solution to the power control problem and also a solution to Equation (3.6). Assume another solution p' to Equation (3.6). Then A is not regular and $A \cdot \bar{p} = 0$ for $\bar{p} = p^\star - p'$. Then $A \cdot (p^\star - \lambda \bar{p}) = b$ for all $\lambda \in \mathbb{R}$, and taking any non-zero entry $\bar{P}_i$ and a sufficiently small $\varepsilon \in \mathbb{R}^+$, we get a feasible solution $p^s := p^\star - (\varepsilon / \bar{P}_i)\bar{p}$ with $P_i^s = P_i^\star - \varepsilon$ and $P_j^s \geq 0$ for all $1 \leq i \leq n$, which contradicts with the optimality of $p^\star$. □

Hence, from now on, for a given set of links, we are only interested in finding any solution $p = (P_1, \dots, P_n)^T$ to Equation (3.6). If all P_i are nonnegative and smaller than or equal to $P_{\max}$, this solution is the optimum (and unique). If there is no solution or any solution with some $P_i < 0$ or $P_i > P_{\max}$—which is always the case if there is more than one solution—, we can conclude that the problem is infeasible.

Solving equation system (3.6) in a naive way would not give a real advantage over existing approaches for power control, as practical analytical methods for solving equation systems have almost cubic time complexity. However, we will show in the following how the equation system can be solved stepwise (similar to a Gaussian elimination on parts of the equation system), thus providing an efficient method for incremental activation of transmissions, and how partial solutions of the equation system can be used to react efficiently to changes that can occur in dynamic networks.

3.4 Data Structure for Incremental Activation

We start with a data structure that is optimized for algorithms which activate transmissions iteratively and need an efficient prediction of resulting optimum transmission powers. The proposed data structure consists of a matrix (two-dimensional array) A and a vector (one-dimensional array) b. The combined matrix $[A|b]$ represents the current state of equation system (3.6). For every link l_i, it contains one row that represents the SINR equation of l_i. Entry a_{ij} of matrix A gives the coefficient of power P_j in the current representation of the SINR equation of link l_i. Rows that correspond to inactive links are irrelevant, as the corresponding SINR constraints do not have to be fulfilled. Thus, they do not have to be stored explicitly in memory (they are identical to the corresponding rows of equation system (3.6)), and we will also omit them in our illustrations.

For a set S of active links, we denote the corresponding arrays by A^S and b^S. In the beginning, when $S = \emptyset$, $A^\emptyset$ and $b^\emptyset$ are initialized with the initial values of $\mathcal{A}$ and b in equation system (3.6). Every time a link l_i is activated, matrix $[A^S|b^S]$ is modified such that one can easily extract optimum transmission powers for set $S' = S \cup \{l_i\}$ from the resulting matrix $[A^{S'}|b^{S'}]$. In the following sections, we will see how several useful operations can be realized by such matrix transformations. All transformations guarantee that the underlying solution of the equation system is preserved. In particular, it will always hold that an optimum solution to equation system

$$\sum_{j=1}^{n} a_{ij}^S P_j = b_i^S \quad \forall l_i \in S\,, \tag{3.7}$$

with $P_j = 0$ for all inactive links l_j, gives optimum transmission powers for all active links. The matrix transformations are designed such that after every operation the following invariants hold (cf. Figure 3.1a):

$$\begin{aligned} a_{ii}^S &= 1 \quad \forall l_i \in S & (3.8)\\ a_{ij}^S &= 0 \quad \forall (l_i, l_j) \in S \times S, i \neq j\ . & (3.9) \end{aligned}$$

We will see in the following sections how this structure enables efficient computation and updating of optimum transmission powers.

3.4.1 Extracting Optimum Powers

As a first convenient consequence of invariants (3.7), (3.8) and (3.9), one can infer the unique optimum power assignment for a set S of active transmissions directly from the corresponding matrix $[A^S|b^S]$. The power P_i of an active link l_i is defined by the corresponding equation in (3.7). According to invariants (3.8) and (3.9), $a_{ii}^S = 1$ and $a_{ij}^S = 0$ for all other active links $l_j \in S \setminus \{l_i\}$. Additionally, we know that the transmission powers of inactive links are zero. With this knowledge, equation system (3.7) can be simplified to

$$P_i = b_i^S \qquad \forall l_i \in S\,. \tag{3.10}$$

and we get that if we guarantee that the matrix operations that result in matrix $[A^S|b^S]$ do not alter the solution of the underlying equation system (3.6), then an optimum power assignment for link set S is simply given by

$$P_i = \begin{cases} b_i^S, & \text{if } l_i \in S \\ 0, & \text{otherwise} \end{cases}\,. \tag{3.11}$$

Thus, given $[A^S|b^S]$ for a set S of active links, the optimum power $P_i^\star$ for an active link $l_i \in S$ can be extracted in $O(1)$ time.

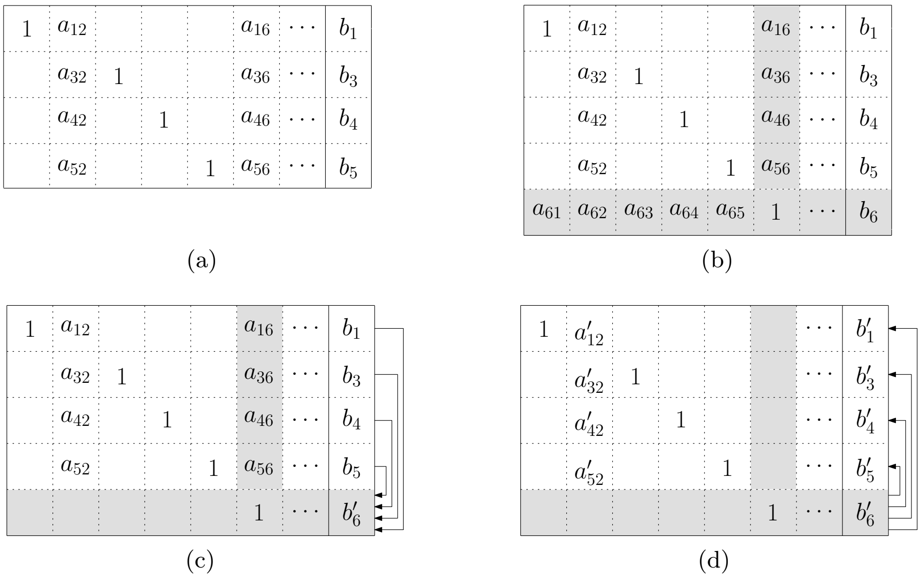

1 a_{12} a_{16} ⋯ b_1
a_{32} 1 a_{36} ⋯ b_3
a_{42} 1 a_{46} ⋯ b_4
a_{52} 1 a_{56} ⋯ b_5
(a)
a_{61} a_{62} a_{63} a_{64} a_{65} 1 ⋯ b_6
(b)
1 ⋯ b'_6
(c)
1 a'_{12} ⋯ b'_1
a'_{32} 1 ⋯ b'_3
a'_{42} 1 ⋯ b'_4
a'_{52} 1 ⋯ b'_5
1 ⋯ b'_6
(d)

Figure 3.1: Matrix $[A|b]$. (a) $[A^S|b^S]$ for $S = \{l_1, l_3, l_4, l_5\}$. (b) When link l_6 is activated, invariants (3.8) and (3.9) are possibly violated for the corresponding row and column. (c) Situation after Steps A and Step B of the activation. (d) After Step C, all invariants are fulfilled again.

3.4.2 Activating a Transmission

Let T be the set of inactive links. We now examine how we can activate an inactive link $l_i \in T$, yielding $S' = S \cup \{l_i\}$ and $T' = T \setminus \{l_i\}$. For now, we assume that there exists a feasible power assignment for S'. To preserve the invariants of our data structure, row i of matrix $[A^S|b^S]$ becomes important. We start by initializing the row with the SINR constraint of link l_i. At this point, all values of row i can be non-zero (cf. Fig. 3.1b). In order to restore invariant (3.9) for row i, we have to produce $a^{S'}_{ij} = 0$ for all $l_j \in S$. As $a^S_{jj} = 1$ at this point, this can be achieved by subtracting row j multiplied by a^S_{ij} from row i for all $l_j \in S$ (Step A):

$$a^\star_{ik} = a^S_{ik} - \sum_{l_j \in S} a^S_{ij} a^S_{jk} \quad \forall l_k \in S \cup T \qquad\qquad b^\star_i = b^S_i - \sum_{l_j \in S} a^S_{ij} b^S_j \tag{3.12}$$

Next, to fulfill invariant (3.8), we divide row i by $a^\star_{ii}$ to get $a^{S'}_{ii} = 1$ (Step B, cf. Fig 3.1c):

$$a^{S'}_{ik} = a^\star_{ik} / a^\star_{ii} \quad \forall l_k \in S \cup T \qquad\qquad b^{S'}_i = b^\star_i / a^\star_{ii} \tag{3.13}$$

Finally, for every l_j in S, we subtract row i multiplied by a^S_{ji} from row j, resulting in $a^{S'}_{ji} = 0$, thus restoring invariant (3.9) for row j (Step C, cf. Fig 3.1d):

$$a^{S'}_{jk} = a^S_{jk} - a^S_{ji} a^{S'}_{ik} \quad \forall l_j \in S,\, l_k \in S \cup T \qquad\qquad b^{S'}_j = b^S_j - a^S_{ji} b^S_i \quad \forall l_j \in S \tag{3.14}$$

Note that the described operations do not alter the solution of the underlying equation system.

The asymptotical time complexity of the whole activation update is in $O(k\,n)$, with $k = |S|$ and $n = |S \cup T|$. This is quite expensive, as we also updated columns of inactive transmissions. As we will see in Section 3.4.3, this allows us to predict very efficiently how the optimum powers change when a single transmission is activated. If such one-step-ahead predictions are frequent compared to actual activations of links, as it is the case for some iterative scheduling algorithms, the additional effort to update columns of inactive links pays of. However, if such predictions are not necessary, one can realize the activation of transmissions in time $O(k^2)$, as we will show in Section 3.6.

So far we did not deal with the case that a set of links cannot be scheduled concurrently when trying to activate an additional link l_i. With the considerations above, closer inspection of the equation system reveals that we can identify an event that indicates and proves infeasibility:

Proposition. Let S be a feasible set of active transmissions with corresponding matrix $[A^S|b^S]$ and l_i an inactive transmission. The power control problem has a feasible solution for $S' = S \cup \{l_i\}$ if and only if the matrix operations in Step A yield an entry $a_{ii}^{S'} > 0$ and the operations in Step B and Step C yield only entries $b_j^{S'} \leq P_{\max}$ for $l_j \in S \cup \{l_i\}$.

Proof. Making an induction over the cardinality of S, we can assume that during the computation of $[A^S|b^S]$ the case of an $a_{ii}^S \leq 0$ never occurred in Step A (this trivially holds for $S = \emptyset$). Hence, during all preceding matrix operations it is an invariant that (a) entries $a_{ii} > 0$, (b) entries $b_i > 0$, (c) entries $a_{ij} \leq 0$ for $i \neq j$. This can be seen by induction over the matrix operations: the invariants hold for the initial matrix $[A^\emptyset|b^\emptyset]$, and if they hold, Step A effectively adds positive multiples of other rows to the changed row. When adding a multiple of row j to row i, the only possible violation of the invariants is exactly the case that a_{ii} becomes non-positive, which, by the outer induction, did not happen so far.

Then, in Step B, we multiply row i by a positive factor (not violating any invariant), and in Step C, we effectively add a positive multiple of row i to the other rows, which by construction cannot violate any invariant either.

If we perform Step A for a new link $l_i \notin S$, and we get $a_{ii} > 0$, then again all invariants hold during all update operations and the relevant blocks of the matrix $[A^{S'}|b^{S'}]$ encode a feasible (and hence optimal) solution for the links in $S' = S \cup \{l_i\}$, given by the $P_i = b_i^{S'}$, which is feasible if and only if all these powers additionally are below $P_{\max}$. If, on the other hand, in Step A a matrix entry $a_{ii}^{S'} \leq 0$ occurs, then there cannot be a solution to the corresponding subproblem with all $P_i \geq 0$, since we still have $b_i^{S'} > 0$, but all $a_{ij}^{S'} \leq 0$. □

3.4.3 Prediction for Activation

In some applications it is necessary to efficiently predict the effects that the activation of a single link l_i would have, without actually updating the full data structure. Using matrix $[A^S|b^S]$, it is possible to compute the corresponding changes very efficiently. By looking at $[A^{S'}|b^{S'}]$ for the resulting set $S' = S \cup \{l_i\}$ of active transmissions, one can see that the optimum power P'_i of link l_i in S' is

$$P'_i = b_i^{S'} = \frac{b_i^S - \sum_{l_j \in S} a_{ij}^S b_j^S}{1 - \sum_{l_j \in S} a_{ij}^S a_{ji}^S} \quad . \tag{3.15}$$

Thus, given $[A^S|b^S]$, we can compute P'_i in $O(k) = O(|S|)$ time. In similar fashion, with $b_i^{S'}$ known, for every link $l_j \in S$ we can compute the new optimum power P'_j in constant time with

$$P'_j = b_j^{S'} = b_j^S - a_{ji}^S b_i^{S'} \qquad \forall l_j \in S \quad . \tag{3.16}$$

This means, given matrix $[A^S|b^S]$ for some feasible link set S, we can compute optimum powers for all transmissions in set $S \cup \{l_i\}$ in $O(k)$ time. This efficiency can only be achieved because of two reasons: First, invariant (3.9) allows us to produce $a_{ij} = 0$ for all $l_j \in S$ with only $O(k)$ operations, whereas we would need $O(k^2)$ operations if we would not have the diagonal matrix structure for the sub-matrix corresponding to the activated transmissions. Second, we precomputed the a_{ji} values during the previous activation steps. Otherwise, we would have to compute them now in $O(k^2)$ time, as described in Section 3.6.

3.4.4 Space Complexity

For a set of n links, the space complexity of the presented data structure is in $O(n^2)$. This is asymptotically optimal, as the gain matrix in the input already has $\Theta(n^2)$ entries. For scheduling approaches where several time slots are filled in parallel, such as the one described in Section 4.3.2, it might be necessary to use several power control data structures, one for each time slot. Let m be the number of slots, n_i be the number of active links in time slot i, and n the overall number of considered links. Then, one only needs $O(n_i\, n)$ matrix entries to store the contents of the data structure of slot i, as only rows of active transmissions have to be stored. Assuming without loss of generality that no link has to be scheduled more than once, one gets with $\sum_{k=1}^{m} n_i \leq n$ that the overall space consumption of the m data structures is still in $O(n^2)$ and thus optimal.

3.5 Data Structure for Dynamic Power Control

In order to allow additional operations, such as the efficient deactivation of single links or power updates when link gains between nodes change, one has to keep track of the operations that resulted in the current state of matrix $[A^S|b^S]$. In this section, we examine how this can be achieved efficiently and what kinds of additional operations this enables.

To keep track of performed operations, we introduce an additional matrix C. Similar to A^S, the state of the matrix corresponding to a set S of active transmissions is denoted by C^S. Matrix C is defined such that:

$$a_{ij}^S = \sum_{k=1}^{n} c_{ik}^S a_{kj}^\emptyset \ ,1 \leq i,j \leq n \qquad\qquad b_i^S = \sum_{k=1}^{n} c_{ik}^S b_k^\emptyset \ ,1 \leq i \leq n \tag{3.17}$$

Descriptively, every row i of matrix $[A^S|b^S]$ is a linear combination of the original SINR constraints of active links, and c_{ij}^S gives the corresponding factor of the SINR constraint of link l_j. In the beginning, when $S = \emptyset$, matrix C is initialized with

$$c_{ij}^\emptyset = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} . \tag{3.18}$$

For C^S, only rows and columns that correspond to links in S (active links) are of interest. Accordingly, only those rows and columns have to be stored explicitly in memory.

3.5.1 Extracting and Predicting Optimum Powers

Given matrix $[A^S|b^S]$, the extraction of optimum transmission powers for active transmissions and the prediction of optimum powers if an additional transmission is activated do not alter matrix $[A^S|b^S]$. Thus, both operations can be performed exactly as described in Sections 3.4.1 and 3.4.3.

3.5.2 Activating a Transmission

If a transmission is activated, the necessary changes for matrix $[A^S|b^S]$ are the same as the ones described in Section 3.4.2. Additionally, matrix C has to be updated so that the invariants of (3.17) remain satisfied. This can be achieved by performing the same matrix operations to C^S that are performed to $[A^S|b^S]$.

In Step A, row j of $[A^S|b^S]$ multiplied by a^S_{ij} was subtracted from row i for all $l_j \in S$. This translates to:

$$c^\star_{ik} = 0 - \sum_{l_j \in S} a^S_{ij} c^S_{jk} \quad \forall l_k \in S, \qquad\qquad c^\star_{ii} = 1 \tag{3.19}$$

Next, in Step B, row i of $[A^S|b^S]$ was divided by $a^\star_{ii}$, which means for C:

$$c^{S'}_{ik} = c^\star_{ik} / a^\star_{ii} \quad \forall l_k \in S \cup \{l_i\} \tag{3.20}$$

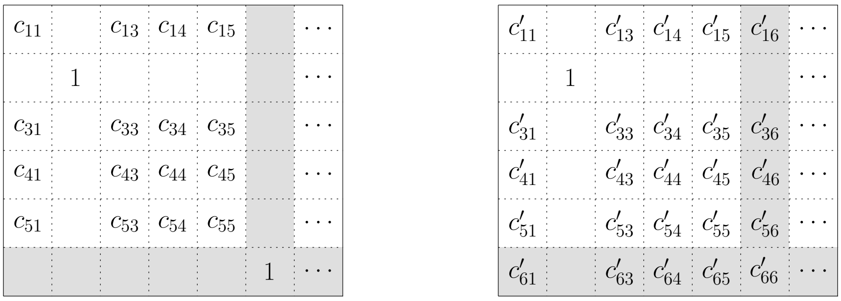
Finally, in Step C, for every l_j in S, row i multiplied by a^S_{ji} was subtracted from row j, which gives:

$$c^{S'}_{jk} = c^S_{jk} - a^S_{ji} c^{S'}_{ik} \quad \forall l_j \in S,\, l_k \in S \tag{3.21}$$

After these operations, the invariants of (3.17) are restored.

Figure 3.2 sketches the situation before and after the activation of some link l_6. Prior to the activation, the row of l_6 in C contains only a single entry, $c_{66} = 1$, as the SINR constraint of l_6 is in its initial state. Similarly, c_{66} is also the only non-zero element in the column of l_6, as the SINR constraint of l_6 has not yet been subtracted from any other SINR constraint. After the activation of l_6, the rows of all other active transmissions have possibly been subtracted from the row of l_6, and subsequently the row of l_6 has been subtracted from the rows of all other active transmissions. Note that it is not necessary to save rows or columns that do not belong to active transmissions, e.g., the rows and columns of l_2 and l_6 in Figure 3.2a.

The update of matrix C has time complexity $O(k^2)$, which means that the overall asymptotic complexity of the activation operation is not negatively influenced by the maintenance of C.



$$\begin{array}{|cccccc|c|}
c_{11} & & c_{13} & c_{14} & c_{15} & & \cdots \\
 & 1 & & & & & \cdots \\
c_{31} & & c_{33} & c_{34} & c_{35} & & \cdots \\
c_{41} & & c_{43} & c_{44} & c_{45} & & \cdots \\
c_{51} & & c_{53} & c_{54} & c_{55} & & \cdots \\
 & & & & & 1 & \cdots
\end{array}
\qquad
\begin{array}{|cccccc|c|}
c'_{11} & & c'_{13} & c'_{14} & c'_{15} & c'_{16} & \cdots \\
 & 1 & & & & & \cdots \\
c'_{31} & & c'_{33} & c'_{34} & c'_{35} & c'_{36} & \cdots \\
c'_{41} & & c'_{43} & c'_{44} & c'_{45} & c'_{46} & \cdots \\
c'_{51} & & c'_{53} & c'_{54} & c'_{55} & c'_{56} & \cdots \\
c'_{61} & & c'_{63} & c'_{64} & c'_{65} & c'_{66} & \cdots
\end{array}$$

(a) (b)

Figure 3.2: Matrix C^S stores the operations that resulted in the current state of matrix $[A^S|b^S]$. (a) Situation for $S = \{l_1, l_3, l_4, l_5\}$. (b) Situation after activation of l_6.

3.5.3 Deactivating a Transmission

In order to efficiently deactivate a transmission, we can use the fact that matrix C^S stores the operations that resulted in the current state of $[A^S|b^S]$. Let us assume that we want to deactivate some link $l_i \in S$, resulting in $S' = S \setminus \{l_i\}$ and $T' = T \cup \{l_i\}$. To remove the effects of l_i on the other active links $l_j \in S'$, we have to perform operations such that in the end it holds that $c_{ji}^{S'} = 0$ for all $l_j \in S'$. At the same time, we have to take care that invariants (3.7), (3.8), (3.9), and (3.17) stay fulfilled.

Thanks to invariant (3.9), we know that $a_{ij}^S = 0$ for all transmissions $l_j \in S \setminus \{l_i\}$. This means that we can subtract row i of $[A^S|b^S]$ from the rows of other active links $l_j \in S \setminus \{l_i\}$ without affecting columns that correspond to links in S'. Only the column of l_i and the columns of inactive links are affected, which is no problem concerning invariants (3.8) and (3.9). Thus, to get $c_{ji}^{S'} = 0$ for all $l_j \in S'$ without violating the invariants, we subtract in $[A^S|b^S]$ and C^S for each $l_j \in S'$ row i multiplied by c_{ji}^S/c_{ii}^S from row j, giving

$$a_{jk}^{S'} = a_{jk}^S - c_{ji}^S/c_{ii}^S \cdot a_{ik}^S \qquad \forall l_j \in S', l_k \in T' \tag{3.22}$$

$$b_j^{S'} = b_j^S - c_{ji}^S/c_{ii}^S \cdot b_i^S \qquad \forall l_j \in S' \tag{3.23}$$

$$c_{jk}^{S'} = c_{jk}^S - c_{ji}^S/c_{ii}^S \cdot c_{ik}^S \qquad \forall l_j \in S', l_k \in S \ . \tag{3.24}$$

The rows of $[A^{S'}|b^{S'}]$ and $C^{S'}$ corresponding to l_i can then simply be reset to the initial state. With $|S| = k$, we get that this update can be performed in time $O(k\,n)$. Without the information stored in C^S, it would not be easily possible to deactivate an arbitrary transmission without violating the invariants. Instead, one would have to generate $[A^{S'}|b^{S'}]$ by resetting the data structure to the initial state and then using $|S'|$ activation operations to activate the links of S' one-by-one, resulting in time complexity of $O(k^2 n)$.

3.5.4 Prediction for Deactivation

Analogous to the prediction of activation effects, one can also predict the effects that the deactivation of a single transmission would have, without having to update the full data structure. If some link l_i is deactivated, the powers for the links $l_j \in S \setminus \{l_i\}$ which remain active are given by

$$P'_j = b_j^{S'} = b_j^S - \frac{c_{ji}^S}{c_{ii}^S} b_i^S \quad \forall l_j \in S \setminus \{l_i\} \ . \tag{3.25}$$

Thus, for each link l_j that remains active, the new optimum power P'_j can be predicted in constant time. As l_i is deactivated, we get $P'_i = 0$. With $k = |S|$, the prediction of all updated powers can thus be achieved in $O(k)$ time. Note that this fast computation is only possible because of invariants (3.8) and (3.9). Without these invariants, subtracting row i of $[A^S|b^S]$ from some other row j would possibly affect entries a_{jk}, $k \in S'$, making additional adjustments necessary.

3.5.5 Changes in Background Noise

Next we examine the necessary operations to react to changes in background noise. One scenario where this kind of update may be important are local scheduling approaches, where scheduling decisions are solely based on local information. In this scenario, the interference from transmissions that are outside of the local neighborhood will be noticed as background noise. Accordingly, if the activity outside of the local neighborhood changes, it may be necessary to adjust the background noise level used in the calculations. Let us assume that the background noise level of one link l_i changed from η_i to η'_i. This changes the initial $b_i^\emptyset$ value by

$$\Delta b_i^\emptyset = \frac{\beta_i}{\gamma_{ii}}(\eta'_i - \eta_i) \tag{3.26}$$

which gives us

$$P'_j = (b_j^S)' = b_j^S + c_{ji}^S \Delta b_i^\emptyset \quad \forall l_j \in S \ . \tag{3.27}$$

This update can be performed in $O(1)$ time for each link, resulting in an overall time complexity of $O(k)$. Again, this efficiency is only possible thanks to the information stored in matrix C.

3.5.6 Changes in Link Gain

If some link gain γ_{ij} between an active sender s_i and an active receiver r_j changes, for example because a node moved, this has severe effects on matrix $[A^S|b^S]$. For $i \neq j$, value $a^{\emptyset}_{ji}$ changes. However, due to the performed matrix operations, this affects all active rows of column i. Restoring the invariants would require $O(k\,n)$ operations. For $i = j$, the situation is even worse. As γ_{ii} affects all columns of row i except of column i, in the end most entries of $[A^S|b^S]$ are affected. Additionally, it is likely that many link gains change at the same time. For example, if a node moves, its effect on all active transmissions that are in proximity will change. Accordingly, the easiest way to deal with link gain changes is to simply perform a deactivation operation for the corresponding node, adjust the corresponding matrix row to the new circumstances, and reactivate the transmission. The complexity of this operations is in $O(k\,n)$, thus asymptotically this approach is not worse than restoring the invariants directly.

3.5.7 Space Complexity

For matrix C, only the rows and columns that correspond to active transmissions have to be saved. With $k = |S|$ being the number of active transmissions, the space complexity is in $O(k^2)$. Accordingly, the space complexity of the whole data structure is dominated by matrix $[A|b]$, whose space complexity was analyzed in Section 3.4.4.

3.6 Data Structure without Prediction

So far, we involved all columns of $[A|b]$ in the updates, even those of inactive transmissions. This allowed us to achieve the activation prediction in $O(k)$ time, which is very handy for scheduling approaches that base their decisions on the effects that the activation of a transmission entails. If such predictions are not necessary or rare, one can improve the complexity of other operations at the cost of the prediction operation. We now describe how the operations can be adjusted, and which improvements are possible.

All operations that change $[A|b]$ basically do nothing but adding/subtracting matrix rows from each other or multiplying a single row with some constant value. Instead of adding/subtracting/multiplying the whole row, one can also restrict the operations to the columns that correspond to active transmissions. With this modification, the complexity of adding/subtracting/multiplying a row is reduced from $O(n)$ to $O(k)$. As the powers of inactive transmissions are set to zero, this has no direct influence on the solution of the equation system that is given by the rows of active links.

With this modification, the time complexities of transmission activation, transmission deactivation, and link gain updates are reduced to $O(k^2)$. However, for operations that involve the activation of a new transmission, additional modifications are necessary. If some transmission l_i is activated, the corresponding column is outdated as only columns of active transmissions have been updated. Accordingly, for all $l_j \in S$, the a_{ji} values have to be updated. To achieve this efficiently, we can use the fact that matrix C stores how row j of matrix $[A^S|b^S]$ evolved from $[A^\emptyset|b^\emptyset]$. Using this information, we get

$$a^S_{ji} = \sum_{l_k \in S} c^S_{jk} a^\emptyset_{ki} \quad \forall l_j \in S \tag{3.28}$$

Retrieving the a^S_{ji} values for all active links l_j has time complexity $O(k^2)$. For the activation of links, this means that the resulting time complexity is also in $O(k^2)$. Unfortunately, the activation prediction also needs the a^S_{ji} values, so the corresponding time complexity is dominated by the computation of the a^S_{ji} values and thus in $O(k^2)$. Asymptotically, the prediction thus is no longer more efficient than actually activating the transmission. Yet, in practice, it still allows to save some computations.

3.7 Comparison and Applications

Table 3.1 shows a comparison of asymptotical running times for the different operations of the presented data structures. The first variant of the data structure, which is optimized for TDMA scheduling approaches that require frequent power predictions for the activation of single links, is denoted *Variant 1*. The extended data structure for dynamic scenarios that also allows additional operations, such as the deactivation of links or updates after properties of the network change, is labeled *Variant 2*. Finally, the version that is optimized for situations without activation prediction is denoted *Variant 3*. Time complexities are given in dependence of the number k of active transmissions and the number n of considered links.

Note that the trivial approach that computes optimal transmission powers from scratch using a standard approach for the solution of equation system such as Gaussian elimination requires $\Theta(k^3)$ time for each operation and the iterative approach that approximates the optimum transmission powers needs a constant number of iterations with time complexity $\Theta(k^2)$ per iteration.

	Variant 1	Variant 2	Variant 3
Extracting Powers	$O(k)$	$O(k)$	$O(k)$
Activating	$O(k \cdot n)$	$O(k \cdot n)$	$O(k^2)$
Deactivating	-	$O(k \cdot n)$	$O(k^2)$
Activation Prediction	$O(k)$	$O(k)$	$O(k^2)$
Deactivation Prediction	-	$O(k)$	$O(k)$
Link Gain Update	-	$O(k \cdot n)$	$O(k^2)$
Noise Update	-	$O(k)$	$O(k)$

Table 3.1: Comparison of asymptotical time complexities for the different data structures.

Which approach is preferable strongly depends on the considered application. If we are simply given a set T of transmissions for which we want to compute good transmission powers once, the existing iterative algorithm that is used in many approaches for scheduling with power control might be best. It will return almost optimal powers in time $O(|T|^2)$, whereas stepwise activation with the proposed data structures requires $O(|T|^3)$ time.

This situation changes as soon as optimum powers for similar sets of transmissions have to be computed over and over, for example if a scheduling algorithm computes iteratively the effects of transmission activations or if we have an online situation where transmissions are activated and deactivated dynamically. Let us consider the case where a scheduling approach assigns transmissions one-by-one to slots. In this scenario, both the iterative method and the third variant of the proposed power control data structure have time complexity $O(|T|^3)$ to activate a set T of transmissions. However, the proposed data structure computes the optimum solution directly by solving the underlying equation system stepwise, whereas the iterative approach needs several iterations per transmission activation to approximate the optimum solution.

Lastly, we consider the situation where a scheduling approach bases its decisions on the effects that the activation of link entails. Examples for such algorithms are given in Section 4.3. Here, the additional effort for updating columns of inactive links is more than compensated by a frequent use of the activation prediction operation, which allows to predict the transmission power changes in $\theta(k)$ time, in contrast to $\theta(k^3)$ to get an exact solution by solving the underlying equation system every time from scratch using an approach such as Gaussian elimination, or $\theta(k^2)$ for approximating optimal transmission powers using the iterative power control approach.

3.8 Summary

We presented novel data structures for the computation of optimum transmission powers under the assumptions of the general SINR model. All data structures work by exactly solving the underlying equation system, which is defined by the link gains between nodes and by the background noise, stepwise. This makes them very efficient for situations where the final set of active transmissions is not known beforehand, but constructed iteratively by activating transmissions one-by-one.

Depending on the considered scenario, one can choose from three variants of the data structure: The first two variants aim at situations where predictions of power changes are frequent, whereas the third variant is more appropriate when transmissions are activated one after another, but without or with only few intermediate predictions of power changes. Concerning the difference between the first two variants, the first variant is optimized for situations where only link activation is necessary, whereas the second variant supports additional operations at the prize of maintaining some additional information.

The presented data structures can be used as building blocks for all kinds of approaches that require the centralized computation of optimum transmission powers. An example is given in the next chapter, where the data structures are used in connection with the scheduling of wireless transmissions.

Chapter 4

Energy-Efficient Scheduling

In this chapter, we consider the computation of energy-efficient time division multiple access (TMDA) schedules for wireless transmissions. Based on the data structures for dynamic power control that were presented in Chapter 3, new heuristics for the problem of scheduling with power control are proposed. The heuristics aim particularly at the computation of balanced schedules, which offer high throughput but at the same time also reduce the necessary transmission powers. The performance of the heuristics is evaluated—based on simulations—in comparison to several existing approaches for the TDMA scheduling problem. According to the simulation results, the presented heuristics achieve significant improvements in terms of both throughput and power-efficiency. Additionally, they can also be adjusted to achieve a compromise between throughput and power consumption. This makes them especially attractive for scenarios in which low transmission powers are more important than mere throughput.

4.1 Introduction

Even with optimal power control (cf. Chapter 3), it is not possible to carry out arbitrary many transmissions concurrently within a limited area. Thus, for reliable communication, it is unavoidable to deal with interferences and transmission failures. In the *Open Systems Interconnection (OSI) model*, it is the task of the *medium access control layer* (MAC layer) to manage the access to the shared medium in wireless networks. Therefore, the MAC layer has to make sure that no transmission gets lost due to interferences, either by retransmitting failed transmissions or by scheduling the transmissions such that interference does not lead to failures. Several *MAC protocols* have been proposed to achieve this goal:

In the *ALOHA* [Rob75] protocol, packets are sent as soon as they are generated. Every time a packet is received, the receiver confirms this with an acknowledgement message. If the sender does not receive an acknowledgment, the message is retransmitted after a random period. Using the ALOHA approach, it can easily happen that two or more nearby senders transmit concurrently. In such situations, both the transmission that started first and the transmissions that started later can fail. To avoid this, *carrier sense multiple access* (CSMA) [TK75] can be used. A sender using CSMA first checks if other nodes are transmitting. If the channel is clear, the transmission starts immediately. Otherwise, the sender waits for all active transmissions to finish and then starts to transmit after some random back-off time. However, using the CSMA protocol it still can happen that two senders s_1 and s_2 both want to send to the same receiver r but are unable to hear each other. Thus, they send at the same time, and both transmissions might fail. This problem is known as the *hidden terminal problem* [TK75]. To avoid the hidden terminal problem, a three-way-handshake is introduced in the *CSMA with collision avoidance* (CSMA/CA) protocol. If a receiver r is willing to accept some transmission, it answers with a *clear-to-send (CTS)* message. This CTS message is then received by all other senders in transmission range of r and prevents them from communicating with r. Today, most wireless networks use the CSMA/CA protocol. The most famous example is the

IEEE 802.11 wireless LAN protocol family [otICS97].

Time division multiple access (TDMA) methods are a different approach to deal with interference. In contrast to the aforementioned contention-based protocols, where a trial-and-error approach is used, in TDMA nodes agree in advance on a schedule. This schedule assigns nodes to time slots such that interference is minimized. Only nodes that are so far from each other that the caused interferences are neglectable are assigned to the same time slot. TDMA can help in several ways to conserve energy: Nodes can go to an energy-saving sleep mode between their assigned time slots, there is less contention-introduced overhead, nodes can transmit with less transmission power as they can estimate the occurring interference, and—at least theoretically—there are no collisions which result in energy-consuming retransmissions. Accordingly, especially if robustness and durability are of greater importance than network performance, TDMA-based methods are an interesting alternative to contention-based approaches.

In this chapter, we consider an extended version of the TDMA scheduling problem, the problem of scheduling with power control. While in the basic scheduling problem all nodes have fixed transmission powers and the task consists in finding an assignment of transmissions to time slots, the problem of scheduling with power control additionally involves the computation of adequate transmission powers. Power control adds a powerful degree of freedom to the scheduling problem. The quality of a schedule then substantially determines both throughput *and* power efficiency of the wireless communication. Obviously, distribution of transmissions to a small number of time slots is desirable to decrease the time to complete the requests, i.e., to improve the throughput. On the other hand, less interference between concurrent transmissions allows for lower transmission powers and can hence reduce energy consumption. Especially in wireless sensor networks, where energy is a limited and valuable resource, the computation of good TDMA schedules can help to extend the lifetime of the whole network.

Since schedules that are admissible without power control are also solutions if power control is possible, power control can be exploited to reduce the energy consumption of such schedules in a subsequent and largely orthogonal step. However, solving both problems simultaneously helps to compute schedules with higher throughput and less energy consumption: power control admits schedules that would otherwise violate interference constraints, since reducing transmission powers can also reduce interference between concurrent transmissions.

Overview and contributions. Two very natural optimization criteria for good schedules are the length of the schedule and the transmission power that is needed to process all transmissions. At the first glance, those two objectives seem to contradict each other. In shorter schedules, the number of transmissions per slot is higher. The consequence is higher interference, which in turn means that more power is needed. And surely, the most power-efficient schedule would be the one where every transmission has its own slot. However, there is also some synergy. In order to compute short schedules, the interference between concurrent transmissions has to be kept small. And smaller interference also means less power consumption. Therefore, transmission power minimization seems to be a good greedy strategy for both objectives.

Based on this observation, we propose new greedy heuristics for the TDMA scheduling problem, which aim at minimizing the required transmission powers. Two different strategies for the assignment of transmissions to slots are considered: In the first strategy, time slots are filled one after another. This is the usual approach of most existing heuristics. In the second approach, several slots are considered in parallel and transmissions are assigned to the slot where they fit best, i.e., the slot where their activation results in the lowest increase in transmission powers. As we will see, the second strategy allows for much more balanced schedules and makes it also possible to find a good compromise between schedule length and power consumption.

To evaluate the proposed heuristics, we implemented them together with several existing approaches, including heuristics and approximation algorithms. A simulation-based comparison of the different approaches is presented, which considers both the achieved throughput and the necessary transmission power.

4.1.1 Related Work

Scheduling with power control has originally been studied in the context of channelized cellular systems [Zan92a, Zan92b, FM93, LLS95] and code division multiple access (CDMA) systems [UY98]. All early approaches worked by deferring transmissions until a certain minimum signal-to-interference ratio (SIR) was achieved. To compute the achievable SIR, either an eigenvalue decomposition of a normalized variant of the link gain matrix was used [Zan92b] (cf. Section 3.1.1), or a distributed algorithm for power control, in which all senders increase their transmission powers iteratively until the required SINR is met at all receivers [Zan92a, FM93]. The decision which transmissions are deferred was usually based on the underlying link gain matrix. Some of these approaches are described in more detail in Section 4.4.3.

The first work dealing with the joint problem of scheduling with power control in wireless ad hoc networks was presented by ElBatt and Ephremides in [EE02, EE04]. The authors show that the power control problem in TDMA wireless ad hoc networks is similar to the one in channelized cellular systems, making it possible to apply the distributed power control algorithms of [Zan92a, FM93] to the considered problem. The presented algorithm for scheduling with power control alternates between scheduling and power control phases and defers transmissions until the required signal-to-interference-plus-noise ratio (SINR) is met. Distributed algorithms for scheduling with power control with focus on multicasting, i.e., one-to-many transmission, are proposed in [WCRP03, WCRP05]. Cruz *et al.* [CS03] and Bhatia *et al.* [BK04] go one step further and present approaches that consider the joint problem of routing, link scheduling and power control in multi-hop networks. The work by Kozat *et al.* in [KKT04] also considers communication in multi-hop networks. In their approach, links that cause maximum interference are removed until the required SINR is met. An algorithm also taking energy efficiency into account is given by Lu and Krishnamachari in [LK05]. Instead of deferring links until the required SINR is met, they continue deferring links until only one link is left. Among all feasible link sets that are computed by this approach, the one that offers the best relation between the number of scheduled links and necessary transmission powers is chosen.

The work presented so far has originated mostly from the networking and communication communities. In the algorithmic community, where the focus usually is on the theoretical analysis of algorithms, research on communication in wireless networks developed differently. Originally, most work was done on graph-based interference models such as the protocol model [GK00] (cf. Section 2.4.2). This made it possible to apply results from graph-theory to analyze algorithms theoretically, e.g., by reducing the considered problems to variants of independent set, matching, or coloring problems. Examples for such approaches are [RL93, GC01, KMPS04, MW05, vRSWZ05, SMS06, KMW08].

However, over the last years, the focus in algorithmic research shifted from graph-based models to physically-motivated interference-based models. The most famous such model is the *physical model* (SINR_G model) [GK00]. The introduction of the physical model to the algorithmic community is usually attributed to Gupta and Kumar [GK00], who proved bounds for throughput under random node placement and optimal node placement. In the SINR_G model, it is assumed that the received signal strength is fully determined by the distance between sender and receiver, which allows for theoretical worst-case analyses of algorithms.

The shift from graph-based models to variants of the SINR_G model was motivated by a series of studies, which demonstrated that graph-based interference models are too simplistic. For example, Grönkvist and Hansson [GH01] showed that interference-based scheduling protocols result in superior throughput in comparison to graph-based interference models. Furthermore, Behzad and Rubin show in [BR03] that many schedules that are computed in graph-based models are not feasible under the SINR model, as graph-based models do not model the aggregation of interference. Moscibroda *et al.* examine in [MWW06] experimentally nested pairs of transmissions, where a sender-receiver pair is placed between another sender-receiver pair. They show that for certain arrangements, both transmissions can be successful at the same time if the inner sender uses lower transmission power than the outer one. This situation is only reproducible with the SINR_G model, showing that the theoretical limits of protocols that obey the rules of graph-based models can be broken by protocols based on the SINR_G model. An experimental comparison of graph-based models and interference-based models is presented by Maheshwari *et al.* in [MJD08]. They also find that the physical model reflects reality significantly better than all considered graph-based models.

As the SINR_G model is more realistic than graph-based models but it still allows mathematical analyses, it was quickly accepted by the algorithmic community, and many interesting results emerged during the following years. We now present a selection of some of the results concerning scheduling in the SINR_G model. For additional overviews with focus on recent approximation results, we refer to [KV10, HM11, Kes11]. A survey on approximation algorithms for the physical interference model has been presented by Goussevskaia *et al.* in [GPW10].

The influence of power control in the SINR_G model is examined theoretically in [MW06] by Moscibroda and Wattenhofer. They show that uniform power assignments and linear power assignments can result in unnecessary long schedules, by presenting an algorithm that schedules a strongly connected set of links in $O(\log^4 n)$ slots, whereas every schedule with uniform or linear power assignment requires $\Omega(n)$ slots. The results of [MW06] are improved in [MWZ06]. Additionally, the authors prove theoretical bounds on the scheduling complexity of arbitrary topologies, i.e., the time that is required to schedule all communication requests of a topology.

The complexity of scheduling in the geometric SINR_G model was first considered in [GOW07] by Goussevskaia *et al.*. The authors show that the problem is NP-hard if uniform powers are used. In [VKW09], this result was extended to an NP-hardness proof for scheduling with power control under the assumption that the available transmission powers are bounded.

In [BBS06], Brar *et al.* study the scheduling of links with non-uniform link demands. Under the assumption of uniform random node distribution, they prove that their algorithm achieves an approximation factor for the length of the computed schedule relative to the optimum schedule.

Performance guarantees of the first approximation algorithms for the scheduling problem that did not depend on assumptions about the node placement usually still depended on structural properties of the networks, e.g., the ratio Λ of maximum and minimum sender-receiver separation [GOW07, CKM$^+$07]. The first wireless scheduling algorithm with an approximation guarantee independent of the network topology was presented by Goussevskaia *et al.* in [GHWW09]. They introduce a constant-factor approximation algorithm for the problem to schedule as many transmissions as possible in one time slot. By applying this algorithm repeatedly, they get an $O(\log n)$ approximation for the scheduling problem. This result was improved in [HW09] by Halldórsson and Wattenhofer to a constant-factor approximation algorithm for the scheduling problem. In [HW09], they also examine the robustness of the SINR_G model with respect to variations in signal attenuation or ambient noise levels. In particular, they show that constant model and parameter changes modify the schedule length only by a constant. A model which allows such deviations from the predicted signal strength is the *generalized SINR model*, which was proposed by Moscibroda *et al.* in [MWZ06]. There also exist extensions of the SINR_G model which allow to model different link qualities depending on the SINR, e.g., in [SMR$^+$09] a graded SINR model is proposed which assumes that the packet reception rate increases gradually with increasing SINR.

In [KVW08], Katz *et al.* deal with the problem of scheduling in the SINR_G model based solely on local information. They present lower bounds on the limitations that one has to accept in order to design local algorithms and present greedy algorithms with approximation guarantees for scheduling based on local information.

An approximation algorithm for scheduling with power control in the SINR_G model is presented by Halldórsson in [Hal09]. This algorithm guarantees an approximation ratio of $O(\log n \cdot \log\log \Lambda)$, where Λ is the ratio between the longest and shortest link length. The first constant-factor approximation algorithm for scheduling with power control was presented very recently by Kesselheim in [Kes11]. This algorithm works not only in fading metrics, but also in general metrics, then giving an $O(\log n)$ approximation.

Randomized approaches to the scheduling problem with performance guarantees are presented in [FKV09, KV10, HM11].

Besides these approximation results for scheduling in the SINR_G model, there also exist approaches for the computation of optimum or near-optimum schedules for small inputs. In [BVY04], Björklund *et al.* presented a column generation method for spatial TDMA scheduling in ad hoc networks. Further methods for the exact computation of TDMA schedules have for example been presented in [JPPQ03, BR05] based on integer linear programming and in [Völ08] based on constraint programming.

4.2 Problem Definition

In this chapter, we use the same basic setting and terminology as in the previous chapter on power control (cf. Section 3.2). Again, a wireless network $\mathcal{N}$ is considered. We assume that for all senders s_i and receivers r_j the corresponding link gains γ_{ij} are known. For the feasibility of transmissions, we assume the SINR model, i.e., a transmission between a sender-receiver-pair (s_i, r_i) is successful if and only if the corresponding SINR condition (cf. Equation (3.2)) is fulfilled. The SINR thresholds β_i of all receivers r_i are part of the input.

In the *scheduling problem*, one is given a set $L = \{l_1, l_2, \ldots, l_n\}$ of links and transmission powers P_i to be used by the senders. Usually the case of *uniform* transmission powers $P_i \equiv P$ is considered. In the scheduling problem, a partition of L into disjoint transmission sets has to be computed such that all sets are feasible. The single transmission sets are called *time slots* or simply *slots*, and we refer to the number of slots in a schedule as the schedule's *span* or the *length* of the schedule. The goal of the scheduling problem is to find a schedule with minimum span in order to maximize the *communication throughput*.

A generalization of the scheduling problem is the problem of *scheduling with power control*, in which one has to compute a partition of a set L of links and proper transmission powers P_i for all senders. We assume that an upper bound $P_{\max}$ on the available transmission powers is given, i.e., $0 \leq P_i \leq P_{\max}$. Again, the computation of a feasible schedule with minimum span is an important problem, but one can also be interested in minimizing the energy consumption for a given span. Note that pure energy minimization trivially leads to solutions where every time slot contains exactly one single link. In this chapter, we will addresses both relevant problems, pure throughput maximization and the bicriterial problem.

4.3 Heuristics for Energy-Efficient Scheduling

4.3.1 Greedy Slot-by-Slot Approaches

Most existing heuristics for the TDMA scheduling problem use a slot-by-slot approach, i.e., they try to fill one time slot after another as good as possible. To maximize the number of transmissions that fit into a time slot, they usually either start with all transmissions and defer single transmissions until the required signal-to-noise ratio is accomplished, or they fill time slots greedily until no more transmissions fit into the slot.

For the greedy strategies that are presented in this section, we use the second approach. Starting with empty time slots offers two benefits: First, the computation of feasible transmission sets is much faster, as we assume that the number of transmission requests significantly exceeds the number of transmissions that fit into a single time slots. Especially checking whether valid transmission powers exist for a given set of transmissions can be very time-consuming when many transmissions are considered concurrently.

Second, when filling slots, one can easily base scheduling decisions on knowledge about transmissions that definitely will end up in the considered time slot. In the approach that removes transmissions one after another, it could happen that a transmission is removed only because of another transmission which later is also removed from the set of active transmissions.

In many existing approaches, the order in which the links are added or deferred is precomputed and depends on node degree, sender-receiver distance, entries of the link gain matrix, or similar criteria. In contrast, the heuristics presented in this section use a strategy that select links depending on the effects that the decision has on all resulting optimum transmission powers.

Greedy-Least-Maximum-Power (GLMP). In the first considered strategy, we always add the link l_i that minimizes the maximum transmission power that some transmission in the time slot requires after l_i is activated. This way, links are picked which either can tolerate a lot of interference or which fit well to the links that are already in the slot. In the following, we refer to this strategy as *Greedy-Least-Maximum-Power (GLMP).* Formally, the selection strategy of *GLMP* can be stated as follows: Let S be the set of transmissions that are already active and T the set of transmissions that are still to be scheduled. $P(l_j)$ denotes the optimum transmission power of some already active link $l_j \in S$ and P_j^i denotes the optimum power for P_j if some link $l_i \in T$ is additionally activated. *GLMP* then chooses a link l_i such that

$$l_i \in \operatorname*{arg\,min}_{l_i \in T} \left(\max_{l_j \in S \cup \{l_i\}} P_j^i \right) \quad . \tag{4.1}$$

Greedy-Least-Additional-Power (GLAP). The second strategy that we consider, *Greedy-Least-Additional-Power (GLAP)*, picks in each step the link that minimizes the combined transmission power used by all links in the time slot after the new link is activated. While the first strategy, *GLMP*, avoids that a single node requires high transmission powers, *GLAP* aims at minimizing the average power of all links. Formally, in each step *GLAP* activates a link l_i with

$$l_i \in \underset{l_i \in T}{\arg\min} \left(\sum_{l_j \in S \cup \{l_i\}} P_j^i \right) \quad . \tag{4.2}$$

Complexity of link selection. Both approaches make it necessary to compute in each step for every link $l_i \in T$ the new optimum transmission powers of the resulting link set $S \cup \{l_i\}$, assuming that l_i is activated. Using the iterative power control algorithm that is used by most existing heuristics, this takes $O(k^2)$ time for each of the $O(n)$ possible links, where $k = |S|$ is the number of links that are already assigned to the considered time slot and $n = |S \cup T|$ is the overall number of links. (Note that in this work we assume the centralized computation of schedules and transmission powers). At this point, the power control data structure that was presented in Section 3.4 comes into play. It allows us to predict updated optimum transmission powers for $S \cup \{l_i\}$ in $O(k)$ time. This makes it possible to find the best link in $O(k\,n)$ time instead of $O(k^2\,n)$ time. As soon as we have determined the optimum link l_i, we can add it in $O(k\,n)$ time to the set S of active links.

Scheduling algorithm. In summary, the sequential slot-by-slot scheduling approach works as follows: Start with an empty set S of active links. While the set T of links that have to be scheduled is non-empty, find the link $l \in T$ that fits best to the current set of active links. Depending on the objective, this can either be the link that minimizes the maximum transmission power (*GLMP*) or the one that minimizes the combined transmission power after the new link is activated (*GLAP*). If such a link exists, add it to S and continue with the set $T' = T \setminus \{l\}$. If no more links fit to the set of active links, continue with a new time slot. This process is repeated until T is empty. The overall running time is in $O(k_{\max} n^2)$, where $k_{\max}$ is the maximum number transmissions that are active concurrently in any time slot. Note that the worst-case occurs when $k_{\max} \in O(n)$. In this case, the running time is $O(n^3)$. This is still not bad as the size of the input, given by the link gain matrix, is already in $\Theta(n^2)$. Even if one had an optimum assignment of links to time slots given as input, the computation of optimum powers using equation system (3.6) and common approaches to solve equation systems such as Gaussian elimination would in this case also require $O(n^3)$ time.

4.3.2 Balanced Approaches

As stated above, schedule length is not the only possible optimization criterion. In this section, we deal with the problem of finding a good compromise between schedule length and power consumption. Given a set T of links and a set $\mathcal{S} = \{S_1, S_2, \ldots, S_s\}$ of time slots, we want to distribute the links of T to the slots of $\mathcal{S}$ in a power-efficient manner.

For this purpose, we process the links in order of increasing sender-receiver-gain, so that we deal with the most sensitive links first. For each link $l \in T$, we greedily determine the best time slot $S_i \in \mathcal{S}$ for that link. At this, the goodness of a time slot is measured by the power consumption in the slot after activation of l. Again, we consider two strategies, which are similar to the ones used in the greedy slot-by-slot approach of the last section.

Balanced-Least-Maximum-Power (BLMP). In the first strategy, *Balanced-Least-Maximum-Power (BLMP)*, the considered link l is assigned to the time slot that minimizes the maximum transmission power that is necessary in the time slot after the link l is activated. Let for some link l_j, which is already assigned to some time slot S_i, P_j denote the transmission power that l_j requires to transmit concurrently with all other links that are assigned to slot S_i. Furthermore, let P_j^l denote the transmission power that l_j would require if l were activated additionally in the same slot S_i. Strategy *BLMP* then assigns l to a time slot S_i with

$$S_i \in \operatorname*{arg\,min}_{S_i \in \mathcal{S}} \left(\max_{l_j \in S_i \cup \{l\}} P_j^l \right) \quad . \tag{4.3}$$

Balanced-Least-Additional-Power (BLAP). In the second strategy, *Balanced-Least-Additional-Power (BLAP)*, link l is assigned to the slot S_i that minimizes the additional power consumption that is necessary to activate l. To determine the additional power consumption, we sum up the resulting transmission powers of l and all other links that are assigned to the same time slot, and subtract the sum over all transmission powers in that time slot before l was activated. Formally, *BLAP* selects a slot S_i with

$$S_i \in \operatorname*{arg\,min}_{S_i \in \mathcal{S}} \left(\sum_{l_j \in S_i \cup \{l\}} P_j^l - \sum_{l_j \in S_i} P_j \right) \quad . \tag{4.4}$$

In comparison to the slot-by-slot approaches of Section 4.3.1, the balanced approaches aim at significantly more balanced schedules, since all slots are expected to have on average a similar number of active links.

Complexity. Like *GLMP* and *GLAP*, strategies *BLMP* and *BLAP* require frequent computations of power changes that occur when single links are activated. For this reason, we again use the power control data structure of Section 3.4 to predict updated powers for a slot with k transmissions in $O(k)$ time. As described in Section 3.4.4, it is not necessary to maintain the complete matrix $[A|b]$ for every single slot. Instead, it is sufficient to maintain for each time slot only those rows that correspond to SINR constraints of links that are already assigned to that slot. Thus, for a slot with k active links, we only need $O(k\,n)$ matrix cells. This gives the whole algorithm an $O(n^2)$ space complexity, which is optimal, as the gain matrix in the input already has space complexity $\Theta(n^2)$.

Determining the number of slots. Due to the initially fixed number of slots, it may happen that a link does not fit in any of the available slots. In this case, one or more new slots can be added to the set of available slots.

When the desired number of time slots is not clear from the application, one can also start with a single slot. This approach still yields more balanced schedules than the slot-by-slot approach, since to links with high gain, which are scheduled latest, all opened slots are available.

However, if there is a need to create more than the initial number of slots, balancing is imperfect, since links scheduled early do not have all slots of the final schedule at choice. Hence, when interested in a balanced schedule with low span (i.e., a schedule that needs few time slots), it is best to find the minimum number of slots that is sufficient for this approach, e. g., using binary search. To achieve under normal circumstances a reasonable result with only a single restart, one can take the schedule returned for a single initial slot and use its span to make a good guess for the number of initial time slots for a second iteration. For example, 80% of the time slots that were necessary in the first iteration should already by a good choice for the initial number of slots in the second iteration.

4.4 Simulations

To examine the performance of our algorithms, we implemented them together with several existing approaches to the TDMA scheduling problem. In this section, we describe our simulations and present simulation results for different application scenarios.

4.4.1 Scenarios and Model Parameters

For the simulations, we use three different scenarios. The first one assumes that the sender-receiver pairs are randomly distributed in the Euclidean plane. There is no connected network structure. The signal strength is computed according to the log-distance path loss model with path-loss exponent $\alpha = 3$ and without noise (cf. Section 2.4.1). This scenario resembles a sensor network with high number of nodes where only some of the nodes want to send concurrently. This kind of scenario was for example used in [GHWW09]. An example is shown in Figure 4.1a.

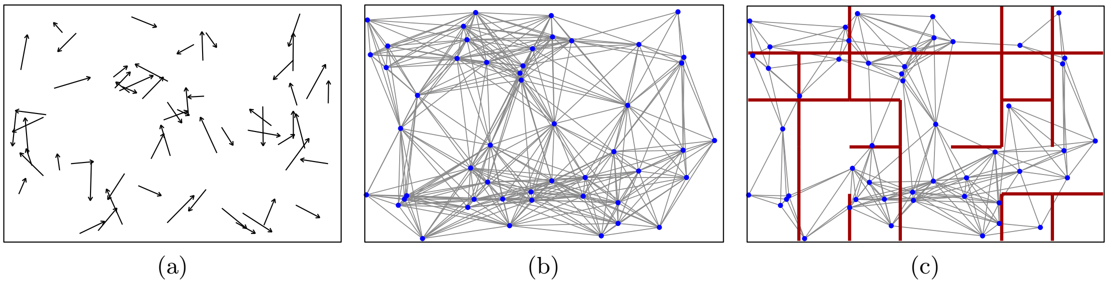


Figure 4.1: Simulation scenarios. (a) Random link scenario. (b) Network scenario. (c) Network in building with wall attenuation and random effects.

The second scenario is based on a network topology. The nodes are placed randomly in the Euclidean plane. Every pair of nodes that can communicate according to the SINR_G model ($\alpha = 3$, $\beta = 10$) is connected by two links, one in each direction (cf. Figure 4.1b). This scenario represents a network in which all nodes try to communicate frequently. Similar models were used in [EE02, EE04]. We assume that all links have the same traffic demands and that each link only has to be active for exactly one time slot, in which the corresponding SINR condition is fulfilled. This prevents that single links dominate the length of the computed schedules. Of course, all algorithms can just as well be used in situations with arbitrary link demands.

Our last scenario aims at recreating effects that occur in buildings with walls and obstacles. We assume that the signal strength falls of with path-loss exponent $\alpha = 2.5$ as long as there is no wall. Additionally, every wall that crosses the line of sight between a sender-receiver-pair results in normally distributed attenuation. Finally, the random effects that are caused by reflections and self-interference are represented by adjusting the signal strength of every link with a zero-mean, normally distributed random attenuation with standard deviation $\sigma = 2$dB. An example is shown in Figure 4.1c. One can see that the signal attenuation at walls leads to some kind of cluster formation within rooms, a feature that does not show up in the other scenarios.

In all scenarios, we assume an omnipresent background noise η and minimum SINR $\beta = 10$. The other model parameters are normalized such that the maximum transmission radius $d_{\max}$ with power $P_{\max} = 1$ equals distance 100 if there are no walls and if we assume $\alpha = 2$ for the path loss exponent. For $\alpha = 3$, this gives $d_{\max} = 21.54$.

Note that in all scenarios we make the simplifying assumption that the exact signal strength is known. This means that under the given model assumptions we do not have to care about transmission failures as long as the algorithms make sure that the required SINR ratios are fulfilled. In the following simulations, we study how many time slots each algorithm needs to process all links under this conditions, and how much transmission power (i.e., the sum over all transmission powers P_i) is required.

4.4.2 Input Generation

In the first scenario, we distributed between 100 and 2500 links randomly in an area with dimensions 400×400. First, the senders were placed randomly. Subsequently, the receivers were placed within a radius of $0.9d_{\max}$ around their senders.

In the second scenario, we placed between 25 and 250 nodes randomly in an area of dimensions 200×200. All pairs of nodes with distance less than $0.9d_{\max}$ were connected by two links, one in each direction. In this scenario, every node can act both as sender and receiver and be part of many links. However, we additionally ensured that in every time slot every node can only take part in one transmission.

In the indoor scenario, between 20 and 200 nodes were randomly distributed in an area of size 200×200. Additionally, wall segments were placed on a regular 8×8 grid. On each of the 112 inner grid segments, a wall was put with probability 60%. The attenuation of each wall segment was determined by a normally distributed random value with $\mu = 5$dB and $\sigma = 2$dB. Finally, links were added for all nodes that were able to communicate with each other according to the general SINR model.

4.4.3 Examined Algorithms

In order to evaluate our algorithms, we implemented several existing algorithms for the scheduling problem. At this, we only considered algorithms that are especially designed and optimized for the physical SINR model (including the $\mathrm{SINR_G}$ model). This section gives a short overview on the selected algorithms. In case an algorithm was not named by the authors, we gave a name based on author names and year of publication.

SRA (Stepwise Removal Algorithm) and *LISRA* (Limited Information Stepwise Removal Algorithm), which have been proposed by Zander in [Zan92b] and [Zan92a], are among the first approaches for scheduling with power control. Both approaches work by postponing transmissions to the next time slot until the required signal-to-interference ratio (SIR) β is achieved. The optimum achievable SIR for a given set of links is computed by an eigenvalue decomposition of the normalized link gain matrix Z (cf. Section 3.1.1). In *SRA*, in each step the link with the largest row or column sum of Z is removed, since these sums provide a bound on the maximum achievable eigenvalue. *LISRA*, on the other hand, removes in each step the link that achieves the lowest SIR when all senders transmit with equal power. We had to adjust both approaches slightly, as in their original formulation they do not consider background noise. The modification consisted in using optimum transmission powers respecting background noise instead of the eigenvalue decomposition to check whether a set of transmissions is feasible.

An approach very similar to *SRA* is *SMIRA* (Stepwise Maximum Interference Removal Algorithm), which has been proposed by Lee *et al.* in [LLS95]. The algorithm also defers transmissions until the achievable SIR exceeds the given threshold. Instead of using row or column sums of the normalized gain matrix, *SMIRA* bases its decision on the interferences that are caused under an optimum power assignment, and defers transmissions that either receive or cause a lot of interference. We implemented *SMIRA*, but due to its high computational cost in our simulation scenarios (as we consider many links at once, *SMIRA* requires many eigenvalue decompositions for large matrices), we did not include it in our simulations. However, in a comparison based on smaller instances, *SMIRA* performed about equally well as *SRA*.

ElBatt04 [EE04] is a very influential algorithm for scheduling with power control in the SINR model. The algorithm alternates between scheduling and power control and defers transmissions with minimum SINR until an admissible set of powers can be found.

Like the heuristics proposed in this thesis, *DiGreedy* [LK05] is an approach to find a good compromise between schedule length and power consumption. The underlying principle is similar to the one in *ElBatt04*. Links with high interference are deferred until a feasible link set is found. Additionally, every feasible set is rated based on schedule length and power consumption. In order to optimize the relation of throughput and power consumption, additional links are removed one by one from the set of active links. In the end, from all feasible link sets the one with the best rating is chosen. Using a parameter, one can decide whether throughput or power efficiency is more important. When optimized for throughput, *DiGreedy* is equivalent to *ElBatt04*. As *DiGreedy* is very computational expensive and at most as good as *ElBatt04* concerning the throughput, we did not analyze it further.

Instead of selecting links that are deferred, *LiEph05* [LE05] starts with empty slots and adds links according to some scheduling metric that takes into account queue sizes and the number of blocked links. As soon as no more links fit into a slot, a new slot is opened. A similar approach is used by *GreedyPhysical* [BBS06]. Instead of the scheduling metric that is used in *LiEph05*, an interference number is used to sort the links in the beginning. The links are then processed according to their order and added to the first feasible time slot. Under the assumption of a uniform random node distribution, the authors were able to prove an approximation factor for the length of the computed schedule relative to the optimum schedule.

ApproxLogN, which was proposed by Goussevskaia *et al.* in [GHWW09], was the first algorithm which guaranteed an $O(\log n)$ approximation for the scheduling problem in the SINR_G model without further assumptions concerning the node placement. The authors also showed that their algorithm is superior to the algorithm *ApproxDiversity*, which was proposed in [GOW07].

HaWa09, which was introduced by Halldórsson and Wattenhofer in [HW09], is a simplified version of *ApproxLogN*. *HaWa09* is the first algorithm for which the authors could prove that it achieves a constant factor approximation for the scheduling problem in the SINR_G model.

Note that *GreedyPhysical*, *ApproxLogN*, and *HaWa09* are optimized for scheduling with uniform transmission powers. We applied the algorithms according to their original description and computed subsequently optimum powers for the generated schedules.

4.4.4 Results

Throughput. We start with an examination of the lengths of the computed TDMA schedules, i.e., the number of required time slots. The schedule length determines the communication throughput, and all considered algorithms are intended for computing short schedules.

Figure 4.2 shows the distribution of schedule lengths for 500 independent runs based on the random link scenario. In each run, 800 sender-receiver pairs have been distributed randomly on the area with dimensions 400 × 400. We observe that the heuristics that are proposed in this work resulted in significantly shorter schedules than the existing approaches. In the following, we are going to interpret this result in detail.

Let us start with the approaches *SRA*, *LISRA*, and *ElBatt04*, which try to fill one slot after another as good as possible, by choosing links that are deferred. All three approaches have the same problem: To decide which link is removed from the slot, all links that are not yet deferred are taken into account in the decision process. Accordingly, it can happen that a link is removed due to another link l_1, which later is also removed due to yet another link l_2. As a result, some links might be removed although they actually would fit to the final set of links that are activated.

This problem has been analyzed theoretically by Moscibroda *et al.* in [MOW07]. The authors showed that it is possible to create an input with n links, where all three heuristics require $\Omega(n)$ time slots, although it is possible to schedule all links in $O(\log n)$ slots. Thus, the schedules computed by the heuristics can be exponentially worse than the optimal schedules.

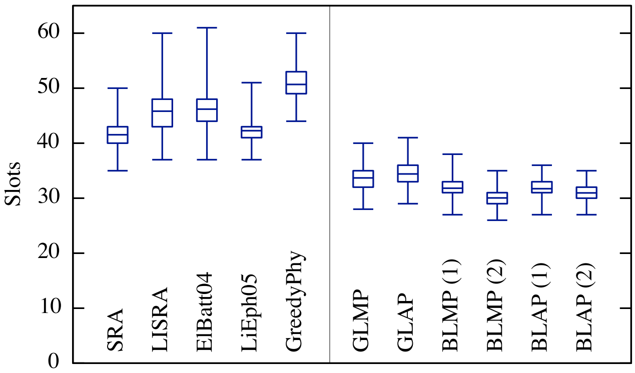


Figure 4.2: Schedule lengths for the random link scenario. The boxes show the upper and lower quartiles, the whiskers show the maximum and minimum schedule lengths. The bars in the middles of the boxes show the mean values.

Algorithm *LiEph05* fills slots starting with an empty slot, but it does not efficiently consider how well a new transmission fits to the transmissions that are already assigned to the slot. Thus, it could happen that two transmissions that are located closely are assigned to the same time slot. In consequence, both transmissions interfere strongly with each other and this possibly prevents other links from being activated. In the worst case, it is possible that a pair of transmissions is activated in the beginning such that one of the transmissions can tolerate no additional interference at all. In this case, according to the SINR model, no other transmission can be assigned to the same time slot.

The worse result of *GreedyPhysical* in comparison to the other existing approaches is explained by the fact that *GreedyPhysical* is the only algorithm considered so far that does not use power control to check whether links can be activated concurrently. Apparently, this has negative effects on the achievable throughput.

Next, let us consider heuristics *GLMP*, *GLAP*, *BLMP*, and *BLAP*. For both balanced approaches, *BLMP* and *BLAP*, two variants are presented. The first one starts with only one open slot, and new slots are opened as soon as necessary. In contrast, the second variant is executed after the first variant, and right from the beginning 80% of the slots that were necessary using the first variant are opened. Apparently, both *BLMP* and *BLAP* could improve their results slightly by using a second run.

Overall, the balanced approaches resulted in schedules with slightly higher throughput than the greedy approaches *GLMP* and *GLAP*, with the *BLMP* approach achieving the highest throughput of all considered approaches.

Note that algorithms *ApproxLogN* and *HaWa09* are not included in Figure 4.2, as they required on average significantly more time slots (292 and 694, respectively) than the other approaches. At first, this might sound surprising, as both approaches provide provable worst-case guarantees. However, there is an explanation. First of all, the intention in the design of the approximation algorithms was very different from the one in the design of the heuristics. While the heuristics are optimized for using maximum knowledge about the actual interference from concurrent transmissions, the approximation algorithms aim at provable guarantees under worst-case assumptions. In the considered scenario, it seems that these worst-case assumptions are more pessimistic than necessary.

It turns out that the size of the deployment area plays an important role in the throughput difference between the approximation algorithms and the other approaches. In our simulations, the considered area is rather small. Accordingly, all interference must emerge from a relatively small area. The approximation algorithms do not utilize this knowledge, but instead they are based on the worst-case assumption that there could also be interference from transmissions further away. In contrast, the heuristics measure the occurring interference and recognize that there is no interference from senders further away. The more conservative assumptions of the approximation algorithms thus cause that less transmissions are assigned to the same slot than possible. We also performed some simulations with significantly larger deployment areas. In our simulations, the difference between the heuristics and the approximation algorithms diminished with increasing size of the deployment area.

Note that part of the difference between the heuristics and the approximation algorithms is also caused by the fact that the approximation algorithms are designed for scheduling with uniform powers. However, the significantly higher throughput of *Greedy-Physical*, which is also based on uniform transmission powers, shows that this alone can only account for a small fraction of the difference.

Table 4.1 shows the mean schedule lengths for the described random link scenario. Additionally, mean schedule lengths for the other two scenarios are presented. For the network scenario, 175 nodes were placed on an area of dimensions 200 × 200, resulting in 830 links on average. In the scenario where the network is within a building, 125 nodes were placed, resulting in 864 links on average. Each value in Table 4.1 represents an average over 500 independent runs.

For heuristics *BLMP* and *BLAP*, two values are given per scenario. Again, the first value corresponds to the first run where the balanced approach starts with only one open slot, whereas the second value corresponds to the run where the number of open slots is

	Random	**Network**	**Building**
Algorithm	(∅ 800 links)	(∅ 830 links)	(∅ 864 links)
SRA	41.5	162.9	237.1
LISRA	45.8	158.4	237.5
ElBatt04	46.2	162.5	247.7
LiEph05	42.3	157.9	199.9
GreedyPhy	50.7	143.0	173.5
ApproxLogN	292.1	632.1	-
HaWa09	694.0	808.5	-
GLMP	33.7	125.2	177.7
GLAP	34.4	127.4	180.6
BLMP	31.8 / 30.0	113.4 / 112.4	161.4 / 161.9
BLAP	31.7 / 31.0	115.1 / 113.3	162.7 / 163.4

Table 4.1: Average schedule lengths for the considered scenarios.

initialized with 80% of the slots that were necessary in the first run.

In their original formulation, *ApproxLogN* and *HaWa09* rely on the assumptions of the $\mathrm{SINR}_{\mathrm{G}}$ model. Accordingly, they cannot guarantee successful reception in scenarios that include random effects, such as the considered indoor scenario. For this reason, for these algorithms no schedule lengths are given for the indoor scenario in Table 4.1.

We observe that in the network scenarios, the differences between the different approaches are significantly smaller. To some extent this is caused by the smaller deployment area, which leaves less possibilities to combine transmissions cleverly. Additionally, the requirement that every node can only take part in one transmission at the same time restricts the possibilities further. However, even in the network scenarios, the balanced approaches *BLMP* and *BLAP* compute the schedules with highest throughput.

In the following, the main focus of the evaluation will be on the approaches *ElBatt04*, *LiEph05*, and *GreedyPhysical*, as well as on the proposed heuristics *GLMP*, *BLMP*, and *BLAP*.

Throughput in dependence of link density. Figure 4.3 shows for the scenario with random sender-receiver pairs how the schedule length increases with increasing number of transmissions, i.e., increasing link density. Every data point shows the mean schedule length for the corresponding number of transmissions, averaged over 150 random runs. Due to the higher computational complexity of the approach, we computed the results for ElBatt04 only for up to 1500 transmissions.

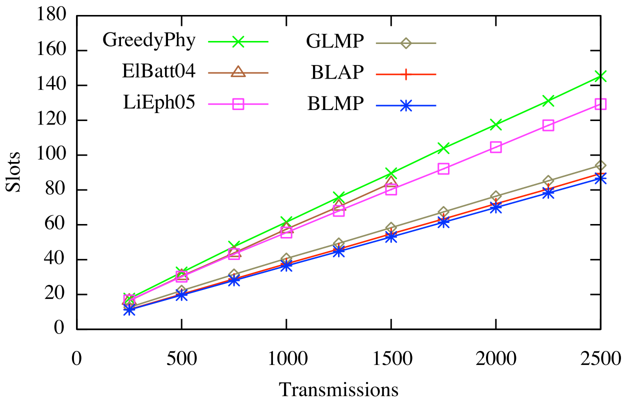
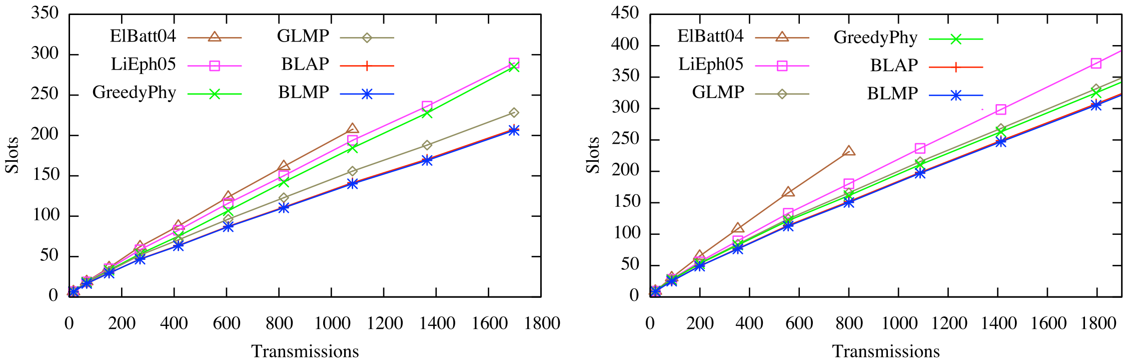


Figure 4.3: Average schedule lengths for the random link scenario.

We observe that all approaches show an almost linear relation between link density and schedule length. Like before, *BLMP* and *BLAP* compute the schedules with highest throughput, with *BLMP* being marginally better.

Figure 4.4 shows the corresponding information for the two network scenarios. Every data point corresponds to an average over 150 runs. In this setting, the relative performance differences between the algorithms are not as distinctive as in the setting with random sender-receiver pairs. Additionally, *GreedyPhysical* seems to gain performance in comparison to *ElBatt04* and *LiEph05*. But we observe that *BLMP* and *BLAP* again compute the schedules with highest throughput.

Power consumption. So far, we evaluated the algorithms based solely on the lengths of the computed schedules. Especially in wireless sensor networks, there might be situations where the available energy is limited and one is mainly interested in energy-efficient schedules. For this reason, we take in this section a closer look at the transmission power that is necessary to complete all transmissions. To estimate the used power, we simply sum up over all transmission powers that are necessary for successful reception according



(a) Network scenario

(b) Network in building

Figure 4.4: Average schedule length for the two network scenarios.

to the SINR model. Note that this is a significant abstraction from reality. In real-world scenarios, many other contributions and effects have to be considered to get a reasonable estimate of the energy that is required for a transmission.

Figure 4.5 shows the dependence between average power consumption and link density. For every algorithm, the average transmission power per node in throughput optimal schedules is given as a fraction of the maximum transmission power P_{max}. Moreover, a lower bound P_{req} on the necessary transmission power is shown. P_{req} is defined as the average power that is used when all transmission are scheduled in different slots. This completely avoids interference, and the transmission power only has to be chosen sufficiently high so that the received signal strength exceeds the background noise by the required signal-to-noise ratio β.

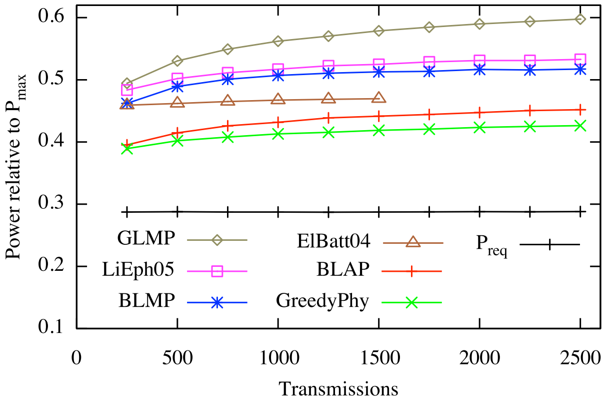


Figure 4.5: Power consumption (random link scenario).

Apparently, there is only a slight increase in average transmission power for higher link densities. The reason is that the average number of transmissions that fit into a single slot is determined by the parameters of the communication model and the distribution of the nodes. This means that using twice as many transmissions usually also means that almost twice as many time slots are needed. Accordingly, the required mean transmission powers do not grow arbitrarily, but they converge to some value that depends on the used algorithm, communication model and node placement strategy.

Of the algorithms considered in Figure 4.5, *GreedyPhysical* results in the lowest transmission powers, followed by *BLAP*. However, when interpreting Figure 4.5, one has to keep in mind that schedules of different lengths are compared. It is to be expected that schedules with lower throughput also need less power, as it is less likely that transmissions interfere with each other. This is also the reason why in the considered scenarios the average powers of *ApproxLogN* and *HaWa09* are almost equal to P_{req}—the algorithms only pack links together which are very distant from each other.

A direct comparison of the average power consumptions for the schedules of all algorithms is given in Table 4.2. Powers are given relative to P_{req}, thus highlighting the overhead that one has to accept for packing several transmissions into the same slot. Table 4.2 is based on the same simulations that were used in Table 4.1 to compare the throughput of the approaches. Accordingly, every value is based on 500 independent random runs.

Algorithm	**Random** (∅ 800 links)	**Network** (∅ 830 links)	**Building** (∅ 864 links)
SRA	155.1	144.5	123.3
LISRA	160.9	153.7	130.9
ElBatt04	168.1	153.7	131.8
LiEph05	164.0	174.9	155.3
GreedyPhy	142.3	158.3	147.0
ApproxLogN	100.2	100.0	-
HaWa09	100.0	100.0	-
GLMP	191.8	171.7	141.8
GLAP	185.1	166.3	137.6
BLMP	185.0 / 174.5	173.5 / 170.2	140.3 / 145.4
BLAP	156.4 / 148.2	153.4 / 146.6	126.5 / 121.6

Table 4.2: Average power consumption relative to P_{req} (in %).

Altogether, *BLAP* shows by far the best relation between throughput and necessary transmission power. As we will see in the following section, reducing the throughput of *BLAP* to the throughputs of the other algorithms results in significant power savings.

Throughput vs. power consumption. As argued in the beginning of this chapter, to some extent throughput and power consumption are conflicting optimization criteria. The lowest transmission powers are necessary when every transmission gets its own time slot. On the other hand, in order to achieve high throughput, many transmissions have to be active concurrently, which also means higher interference and thus higher transmission powers. In this section, we analyze this tradeoff between throughput and energy consumption in more detail. For this purpose, we randomly selected some of the schedules that were computed for the scenario with 1500 random links and plotted them in Figure 4.6. Each data point corresponds to a single schedule and shows the corresponding throughput and energy consumption. The further left a data point is, the higher is the throughput of the corresponding schedule, and the lower a point is, the less transmission power is used on average. The different point colors correspond to different scheduling approaches.

Apparently, in this scenario the different point clouds are nicely separated. *BLMP* and *BLAP* achieve the highest throughput. At the same time, *BLAP* is also highly competitive with respect to the required energy. Algorithm *GLMP* also computes short schedules. However, due to the slot-by-slot approach, the transmissions are not distributed in a balanced way, thus the energy demand is much higher than for *BLMP* and especially *BLAP*.

We already mentioned that *BLMP* and *BLAP* allow to compute energy-efficient schedules by using more slots than necessary. This compromise is visualized by the curves in Figure 4.6. For example, one can deduce that, using *BLAP*, one can schedule all 1500 links

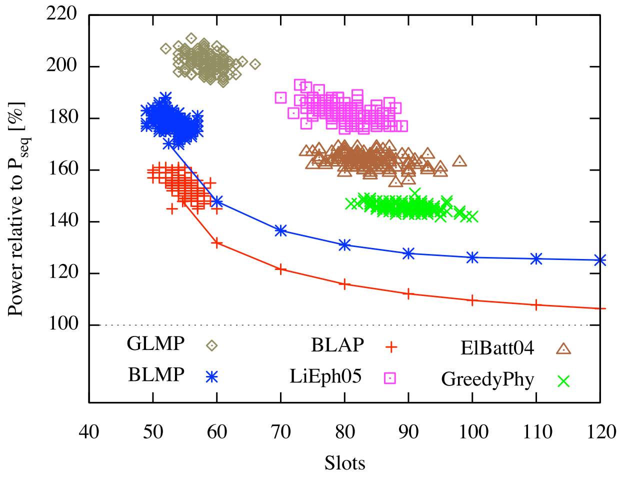
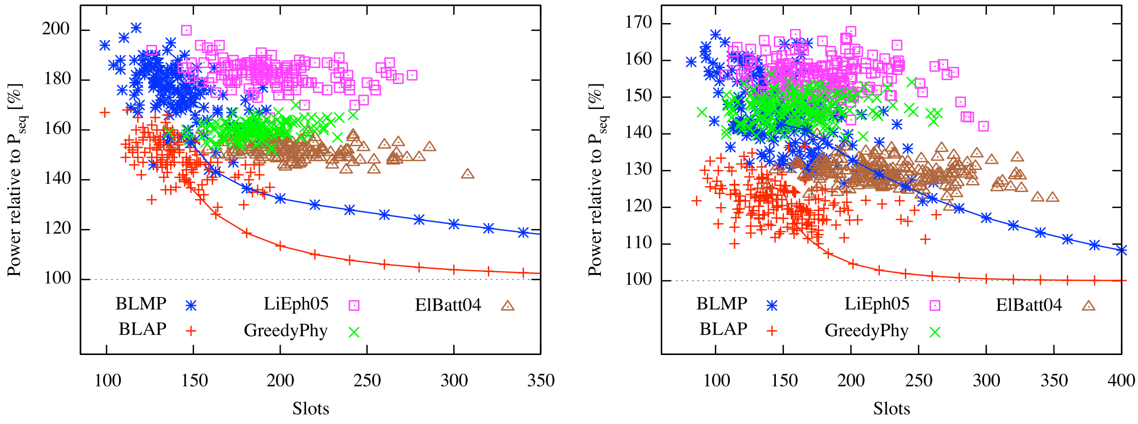


Figure 4.6: Throughput vs. power consumption for 1500 random links.

within 70 time slots using—on average—a mean transmission power of only about $1.2P_{\text{req}}$. The throughput-vs-energy curve of *BLAP* is way below all other algorithms. Thus, given the same number of time slots that any other algorithm uses, the schedules computed by *BLAP* are significantly more economical with respect to transmission powers.

Figure 4.7 shows the corresponding throughput vs. power consumption plots for the scenarios with network topologies. Again, compared to the scenario with random sender-receiver pairs, in these scenarios the differences between the single algorithms are not as distinct. Especially in the indoor scenario, the random effects lead to much scatter. However, the results of all scenarios suggest that *BLAP* computes the best schedules concerning both throughput and used transmission powers.



(a) Network scenario with 200 random nodes

(b) Network in building with 120 random nodes

Figure 4.7: Throughput vs. power consumption for the network scenarios.

Network lifetime. In wireless sensor networks, it is often assumed that the available energy of sensor nodes is limited and that the lifetime of the network thus depends on the power consumption of the single nodes. In this section, we examine how the scheduling algorithms could affect the lifetime of the network if the power consumption was mainly dominated by the used transmission powers. We exemplary consider the network scenario with 200 randomly distributed nodes and examine the time that passes until 10%, 20%, 30%, 50%, and 75% of the nodes run out of battery, assuming that all links are active for about the same time and that all nodes have the same initial battery charge.

The lifetime improvement that could be achieved under the aforementioned simplifying assumptions by using the *BLAP* heuristic is shown in Table 4.3. We always assume that *BLAP* is allowed to use the same number of time slots that the corresponding other algorithm would need. For example, when using *BLAP* instead of *ElBatt04*, using the same number of time slots it takes 29.3% longer until 10% of the nodes run out of battery.

compared to	**10%**	**20%**	**30%**	**50%**	**75%**
ElBatt04	29.3	32.5	35.1	40.2	48.9
LiEph05	45.0	51.4	57.8	67.7	85.7
GreedyPhy	42.2	38.7	36.6	33.5	29.9
GLMP	26.4	31.2	35.4	43.1	56.0
Fixed Power	157.6	214.6	270.9	406.4	738.2

Table 4.3: Lifetime improvement by heuristic *BLAP* [in %]

The last row of Table 4.3 shows the advantage of a throughput optimized schedule using *BLAP* in comparison to a schedule without power control. Note that in the considered scenario it does not matter which scheduling algorithm is used to compute the schedule with uniform powers, as all senders use the maximum transmission power either way. Apparently, even in short schedules with very high throughput, a lot of energy can be saved by using power control.

4.5 Final Remarks

Interpretation of the results. In all considered scenarios, the balanced approach that minimizes the sum of resulting optimum transmission powers achieved the best results when throughput and transmission powers are considered concurrently. Only the balanced approach that minimizes the maximum resulting transmission power was able to achieve a slightly higher throughput, at the cost of significantly higher average transmission powers.

While it is not surprising that the balanced approaches reduce the required transmission powers, one might wonder why they have such a positive influence on the throughput. At a first glance, the usual approach of most heuristics, to minimize the occurring interference in some way, seems more appropriate to throughput maximization. However, when comparing both approaches, it turns out that the increase in transmission powers contains more information than the increase in interference. Even if one transmission causes much interference at another concurrent transmission, this is no problem as long as the sender and receiver of the disturbed transmission are much closer than the interfering sender and the receiver. Considering solely the caused interference makes no distinction between those sender-receiver pairs that are close and others that are far apart. In contrast, considering the change in transmission power catches this difference. If sender and receiver of the transmission that is disturbed are close, even high interference causes only a slight increase in the required transmission power.

Worst-case behavior. Given the impressive performance of both *BLAP* and *BLMP* in our simulations, one might wonder how the heuristics perform in worst-case scenarios. Unfortunately, it is possible to construct worst-case instances for *BLAP* and *BLMP* in which both approaches behave extremely bad when they are started with only one open slot, i.e., for an input with n links the heuristics compute a schedule that requires $n/2$ slots, although it is possible to schedule all links within two slots. The basic idea of the construction is as follows: Starting with the link with lowest link gain, the links are distributed in increasing order of link gain such that always two consecutive links are located close together, but the different pairs of links are vastly separated. For each of the link pairs, the senders and receivers are located such that both transmissions can be active concurrently, but in this case the interference at one of the receivers is such high that the receiver cannot tolerate any more interference. If the distances between the different link pairs are chosen sufficiently high, it is possible to schedule $n/2$ links at once, one from each link pair. The remaining $n/2$ links can then be scheduled in a second slot. However, *BLAP* and *BLMP* are defined such that they schedule the single pairs together, as the links are strictly considered in order of increasing link gain, and by construction it is possible to schedule every link pair as long as no other transmissions are active. In consequence, both heuristics produce a schedule that requires $n/2$ slots.

Note that part of the problem was that we assumed that the heuristics are started with only one open slot. If, instead, they were initialized with 2 slots, they would also find the optimum schedule in the described example. For each larger number k of initial slots, they would compute a schedule with exactly k slots. This suggests that finding a good value for the initial number of slots, e.g., by means of a binary search, might be worthwhile.

Combination with approximation algorithms. Even though the worst-case analysis revealed that the presented heuristics can perform arbitrarily worse than a constant-factor approximation algorithm, our simulations also showed that even for approximation algorithms—at least the ones considered in this thesis—there is still a lot of potential for improvements when it comes to random networks. Fortunately, one can easily combine the presented heuristics with arbitrary approximation algorithms, thus combining the benefits of both worlds. First, a constant-factor approximation algorithm (e.g., [Kes11]) is used to compute a schedule that is by at most a constant factor longer than an optimum schedule. Subsequently, beginning with some arbitrary time slot, the transmissions of the considered slot are redistributed to the other slots, using one of the balanced heuristics. This is repeated until it is no longer possible to remove further slots and to redistribute the corresponding transmissions. Such a combined approach promises to produce good results for both random networks and worst-case networks.

Outlook. To achieve the observed improvements in throughput and power consumption, a combination of three strategies was necessary: First, time slots had to be filled beginning with empty slots. Second, instead of filling slots one after another, several slots had to be filled in parallel. Third, transmissions had to be distributed according to the effects that their activation had on the optimum transmission powers of all other links in the same slot. We hope that these observations help in the design of improved algorithms, optimally with worst-case guarantees.

4.6 Summary

In this chapter, we studied the problem of scheduling with power control. Our focus was on heuristic approaches that aim at the minimization of the transmission powers that are required to achieve a certain throughput. For this, the assignment of transmissions to time slots was done based on the effects that the activation of the transmission had on the resulting optimum transmission powers of all senders. To keep the computational costs of the frequent optimum power computations low, we utilized the new power control data structures that were presented in Chapter 3. In addition to the usual greedy slot-by-slot approach, which is pursued by most existing heuristics, we also considered a balanced approach in which several time slots are filled in parallel.

To evaluate the proposed heuristics, we implemented them together with several existing approaches to the problems of scheduling with and without power control. In all three considered scenarios, the balanced approaches achieved significantly higher throughput than the other approaches. Additionally, especially *BLAP* required surprisingly low transmission powers, taking the achieved throughput into account. Thanks to their design, the balanced approaches can be easily used to achieve a compromise between throughput and power consumption. By initializing them with more slots than necessary, the transmissions are distributed evenly, thus reducing the average transmission power that is necessary in the single slots. This allowed *BLAP* in our simulations to compute schedules that outperformed the schedules of the other approaches with respect to both throughput and power consumption at the same time, i.e., *BLAP* yielded Pareto-superior schedules.

Chapter 5

Complexity of Scheduling with Power Control in $\mathrm{SINR_G}$

Recently, the geometric $\mathrm{SINR_G}$ model has become very popular in the algorithmic community. Like abstract graph-based interference models, the $\mathrm{SINR_G}$ model enables a thorough theoretical worst-case analysis of algorithms that deal with communication. However, in contrast to graph-based models, which are usually based on strongly simplifying assumptions, the $\mathrm{SINR_G}$ model is believed to reflect reality reasonably well.

In this chapter, we consider the question concerning the complexity of scheduling with power control in the geometric $\mathrm{SINR_G}$ model. We show that the problem is NP-hard if the available transmission powers are bounded, independent of the actual bounds. This also implies that scheduling with a finite number of power levels is NP-hard. Thus, the presented proof covers most cases that are relevant in practice.

5.1 Introduction

In the general SINR model, link gains can be chosen arbitrarily. This flexibility makes it easy to reduce NP-complete problems to the problem of scheduling in the SINR model, thus proving the NP-hardness of the problem. In contrast, in the geometric $\mathrm{SINR_G}$ model, link gains are not part of the input. Instead, they are implicitly given by the node positions and the resulting distances between nodes. Given the assumption of the $\mathrm{SINR_G}$ model that the signal strength is fully determined by the distance between the sender and the receiver, it is no longer possible to construct arbitrary link gain matrices, making it harder to find reductions of NP-complete problems to the scheduling problem.

The complexity of scheduling in the geometric $\mathrm{SINR_G}$ model was first studied by Goussevskaia *et al.* in [GOW07]. The authors proved that both the scheduling problem and the one-shot scheduling problem, the problem to fit as many transmissions as possible into the same time slot, are NP-hard if all senders use uniform transmission powers. Moreover, approximation algorithms for both problems were given. However, the complexity of scheduling with non-uniform transmission powers stayed unresolved, and analyzing the complexity of the joint problem of scheduling and power control was proposed as an exciting research direction. In [LvRW08], the complexity of scheduling with power control was mentioned as one of five very essential problems to understanding sensor networks.

At a first glance, it seems straightforward that scheduling with power control is NP-hard when even the more constrained problem of scheduling with fixed powers is NP-hard. However, it turns out that this additional degree of freedom makes it actually harder to show the hardness of the problem. The reason is that slight changes of one transmission power directly affect the necessary transmission powers of all nodes which are nearby in a non-trivial way, and that these effects usually propagate through the whole network.

In this chapter, we consider a variant of scheduling with power control, where the available transmission powers are bounded. We assume that every node can choose its transmission power arbitrarily either from an interval $[P_{\min}, P_{\max}]$, with $P_{\min} > 0$ and

$P_{\max} < \infty$, or from a finite set $\{P_{\min} = P_1, P_2, \ldots, P_j = P_{\max}\} \subset [P_{\min}, P_{\max}]$ of *power levels*. We show that under this assumptions, both scheduling with power control (SchedPC) and one-shot scheduling with power control (OneShotSchedPC) are NP-hard. From a practical point of view, the limitation on arbitrary power bounds is not restricting, as wireless hardware usually has a maximum transmission power, and due to ambient noise every successful transmission has to exceed some minimum transmission power.

Our proofs extend the NP-hardness proofs given in [GOW07] for scheduling with uniform powers to the problem of scheduling with power control. The main idea is to construct problem instances such that some senders are forced to transmit with minimum power $P_{\min}$, while the remaining senders are forced to use the maximum power $P_{\max}$. With such a construction, we know all transmission powers in advance and can use a construction similar to the one used in [GOW07].

Related Work. Work related to scheduling with and without power control in the SINR$_\mathrm{G}$ model is presented in Section 4.1.1.

5.2 Problem Definition

In contrast to the previous two chapters, which were based on the general SINR model, this chapter deals with scheduling in the geometric SINR$_\mathrm{G}$ model. In consequence, the problem definitions in this chapter are slightly different from the ones used before. The models and notations used in this chapter are similar to those used in [GOW07] for the NP-hardness proof for scheduling without power control. The main distinction is that we allow arbitrary transmission powers for all senders.

In the two considered scheduling problems, we are given a set $L = \{l_1, \ldots, l_n\}$ of *links*, where each link $l_i = (s_i, r_i)$ represents a *communication request* from a sender s_i to a receiver r_i. The senders and receivers are distributed in the Euclidean plane and the distance between two nodes s_i, r_j is denoted by $d_{ij} = d(s_i, r_j)$. Thus, d_{ii} denotes the distance between a sender s_i and its corresponding receiver r_i. Unlike [GOW07], we do not assume that the senders use uniform transmission powers, but consider the choice of transmission powers P_i to be part of the addressed optimization problems.

The signal power $S_{r_j}(s_i)$ received at r_j from sender s_i depends on the transmission power P_i of s_i and the distance d_{ij} between nodes s_i and r_j. The SINR$_\mathrm{G}$ model assumes a strong relationship between distance and received signal strength. It is assumed that the *link gain* γ_{ij} between nodes s_i and r_j is fully determined by the distance d_{ij} between s_i and r_j, and that $\gamma_{ij} = (1/d_{ij})^\alpha$. The *path loss exponent* α defines how fast the signal decays with distance. Accordingly, in the SINR$_\mathrm{G}$ model, the signal strength $S_{r_j}(s_i)$ received by a node r_j from a sender s_i is given by $S_{r_j}(s_i) = \gamma_{ij} P_i = P_i/d_{ij}^\alpha$. As usual, it is assumed that $\alpha > 2$.

Every sender s_j that sends concurrently with s_i causes an *interference* $I_{r_i}(s_j) = S_{r_i}(s_j)$ at the receiver r_i of link l_i. The notation $I_{r_i}(s_j)$ is used in order to highlight that we talk about interference and not about a useful signal. It is assumed that all interferences accumulate. The *total interference* I_{r_i} experienced by receiver r_i is given as the sum of all interferences caused by concurrently sending nodes, i.e., $I_{r_i} := \sum_{s_j \neq s_i} I_{r_i}(s_j)$. Furthermore, it is assumed that every receiver is exposed to an ambient noise with power η. In the SINR$_\mathrm{G}$ model, a transmission (s_i, r_i, P_i) is successful if and only if the ratio of the received signal strength $S_{r_i}(s_i)$ and the total interference I_{r_i} plus background noise η exceeds some minimum signal-to-interference-plus-noise-ratio (SINR) β, i. e.,

$$SINR(r_i) = \frac{S_{r_i}(s_i)}{I_{r_i} + \eta} = \frac{P_i/d_{ii}^{\alpha}}{\sum_{j \neq i} P_j/d_{ji}^{\alpha} + \eta} \geq \beta, \tag{5.1}$$

with $\beta > 1$. In the following, we ignore the influence of background noise ($\eta = 0$) without loss of generality.

Scheduling with power control. In the problem of *scheduling with power control* (SchedPC), we are given a set $L = \{l_1, \ldots, l_n\}$ of transmission requests, as well as lower and upper bounds $P_{\min}$ and $P_{\max}$, $P_{\min} \neq P_{\max}$, on the available transmission powers. The goal is to assign every sender s_i a transmission power $P_i \in [P_{\min}, P_{\max}]$, and to distribute all requests of L to time slots such that all transmissions in the same slot can be executed simultaneously with the designated transmission powers. At this, a transmission $l_i = (s_i, r_i, P_i)$ is successful if and only if SINR inequality (5.1) is fulfilled. A *schedule* $\mathcal{S} = (S_1, S_2, \ldots, S_T)$ is a partition of L, where S_t denotes the set of links assigned to time slot t and T denotes the *length* of the schedule. Every sender is only active in the assigned time slot. A *power assignment* $\mathcal{P}$ is a function $\mathcal{P} : \{s_i | (s_i, r_i) \in L\} \to [P_{\min}, P_{\max}]$ that assigns to every sender s_i a valid transmission power $P_i := \mathcal{P}(s_i)$. A schedule $\mathcal{S}$ is said to be *valid* with respect to a power assignment $\mathcal{P}$ if all transmissions are successful in their corresponding time slots, using the designated powers. The SchedPC problem for $(L, P_{\min}, P_{\max})$ is to find a power assignment $\mathcal{P}$ and a valid schedule $\mathcal{S}$, such that $\mathcal{S}$ has minimal length among all valid power assignments and schedules.

One-shot scheduling with power control. Instead of asking for a shortest schedule for a given set of links, one can also ask for a maximum number of transmissions to be carried out in a single time slot. If power control is included, this problem is called the *one-shot scheduling with power control* (OneShotSchedPC) problem, and like in the SchedPC problem, we are given a set $L = \{l_1, \ldots l_n\}$ of links and upper and lower bounds $P_{\max}$ and $P_{\min}$ on the available transmission powers. Additionally, all links are weighted, i. e., we are given a weight w_i for every link l_i. Now, in the OneShotSchedPC problem, we try to fill one single slot as good as possible. Thus, the objective is to find a subset $\mathcal{S} \subseteq L$ as well as a power assignment $\mathcal{P}$ such that all links in $\mathcal{S}$ can be scheduled concurrently and $\mathcal{S}$ maximizes the total weight $\sum_{l_j \in \mathcal{S}} w_j$.

5.3 Scheduling with Power Control

In the following, we extend the NP-hardness proof for scheduling without power control, given in [GOW07], to the more general SchedPC problem. In particular, we show that SchedPC is NP-hard for arbitrary $P_{\min}$, $P_{\max}$.

To this extent, we will give a polynomial time reduction of an NP-complete problem to SchedPC, the Partition problem, which has been shown to be NP-complete in [Kar72]: Given a multiset $\mathcal{I} = \{i_1, \ldots, i_n\}$ of integers, find $\mathcal{I}_1, \mathcal{I}_2 \subset \mathcal{I}$ such that $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$, $\mathcal{I}_1 \cup \mathcal{I}_2 = \mathcal{I}$, and

$$\sum_{i_j \in \mathcal{I}_1} i_j = \sum_{i_j \in \mathcal{I}_2} i_j = \frac{1}{2} \sum_{i_j \in \mathcal{I}} i_j \quad .$$

Let $\mathcal{I} = \{i_1, \ldots, i_n\}$ be an instance of Partition. Without loss of generality, we assume that all elements are distinct and positive and we set $\sum_{j=1}^{n} i_j = \sigma$. In order to solve the Partition problem for $\mathcal{I}$, we construct an instance $L_{\mathcal{I}} = \{l_1, \ldots, l_{n+2}\}$ of

SchedPC with $n+2$ links for arbitrary $P_{\min}$ and $P_{\max}$. We will then prove that there exists a schedule of length 2 if and only if the Partition instance $\mathcal{I}$ has a solution.

What makes it hard to find an NP-hardness proof for SchedPC is that every sender can arbitrarily choose its transmission power from an interval of possible powers. In order to handle this, we construct our SchedPC instance such that some of the senders have to use transmission power $P_{\max}$, and the remaining senders have to use power $P_{\min}$. Thus, we know all the transmission powers in advance. This construction is shown in Figure 5.1.

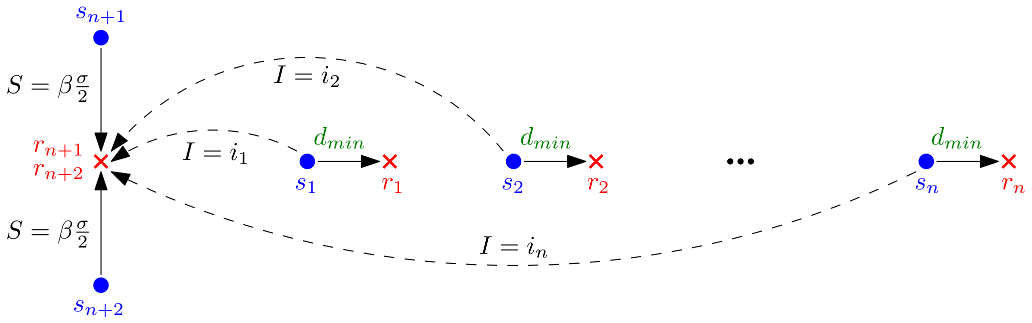


Figure 5.1: Reduction of Partition to SchedPC

For every integer $i_j \in \mathcal{I}$ we introduce a link $l_j = (s_j, r_j)$. Every sender s_j is placed at position

$$\operatorname{pos}(s_j) = \left(\left(\frac{P_{\min}}{i_j}\right)^{1/\alpha}, 0\right), \quad \forall 1 \le j \le n \ .$$

The position is chosen such that the interference caused at the origin $(0,0)$ of the coordinate system equals i_j when s_j sends with power $P_{\min}$. Next, we place the receivers such that every transmission l_j can be executed successfully, even if s_j sends with power $P_{\min}$ and every other sender sends with power $P_{\max}$. For this, every sender-receiver-pair has to be placed sufficiently close together. As we will show later, a distance

$$d_{\min} = P_{\min}^{1/\alpha} \cdot \frac{1/(i_{\max}-1)^{1/\alpha} - 1/i_{\max}^{1/\alpha}}{1 + (\frac{P_{\max}}{P_{\min}} n\beta)^{\frac{1}{\alpha}}} \ ,$$

where $i_{\max}$ is the maximum value in $\mathcal{I}$, is sufficient. Thus, we place every receiver $r_i, 1 \le i \le n$, at position

$$pos(r_i) = pos(s_i) + (d_{\min}, 0) \ .$$

Finally, we have to place l_{n+1} and l_{n+2}. We positioned the senders $s_1, \dots, s_n$ such that the interference which they cause at the origin is proportional to $i_1, \dots, i_n$. In order to take advantage of this property, we place r_{n+1} and r_{n+2} at the origin.

$$pos(r_{n+1}) = pos(r_{n+2}) = (0,0) \ .$$

Last, we place their senders s_{n+1} and s_{n+2} perpendicular to the x-axis at distance $(2P_{\max}/\beta\sigma)^{1/\alpha}$, i.e.,

$$pos(s_{n+1}) = \left(0, \left(\frac{2P_{\max}}{\beta \cdot \sigma}\right)^{1/\alpha}\right) \ ,$$

$$pos(s_{n+2}) = \left(0, -\left(\frac{2P_{\max}}{\beta \cdot \sigma}\right)^{1/\alpha}\right) \ .$$

In the following, we show that

- in every 2-slot solution of the SchedPC problem, the senders $s_1, \ldots, s_n$ have to use transmission power $P_{\min}$,
- in every 2-slot solution of the SchedPC problem, the senders s_{n+1} and s_{n+2} have to use transmission power $P_{\max}$,
- there exists a 2-slot solution to the constructed SchedPC instance if and only if the Partition problem has a solution, and
- every 2-slot solution of the SchedPC instance implies a solution to the corresponding Partition problem.

Let us start with some observations: As r_{n+1} and r_{n+2} share the same position, l_{n+1} and l_{n+2} cannot be scheduled simultaneously. Thus, every schedule needs at least two slots. Moreover, s_{n+1} and s_{n+2} have the same distance to every receiver.

Lemma 1. Every transmission l_i, $1 \leq i \leq n$, is successful with transmission power $P_{\min}$, no matter how many other links are active concurrently and no matter which transmission powers they use.

Proof. Obviously, the worst thing that can happen is that all senders s_j, $1 \leq j \leq n \wedge j \neq i$, and one of the senders s_{n+1}, s_{n+2}, transmit concurrently with power $P_{\max}$. Let $L_i = \{l_j | 1 \leq j \leq n+1, i \neq j\}$. Since the positions of the senders $s_1, \ldots, s_n$ depend on the values of $i_1, \ldots, i_n$, we can determine the minimum distance between two senders s_j and s_k, $j \neq k$,

$$\begin{aligned} d(s_j, s_k) &= |d(s_j, r_{n+1}) - d(s_k, r_{n+1})| &(5.2)\\ &= \left| \left(\frac{P_{\min}}{i_j}\right)^{\frac{1}{\alpha}} - \left(\frac{P_{\min}}{i_k}\right)^{\frac{1}{\alpha}} \right| &(5.3)\\ &\geq P_{\min}^{\frac{1}{\alpha}} \left(\frac{1}{(i_{\max}-1)^{1/\alpha}} - \frac{1}{i_{\max}^{1/\alpha}} \right) . &(5.4) \end{aligned}$$

Thus, the sender s_j closest to r_i, $i \neq j$, is located at least at distance $d(s_j, s_i) - d_{\min}$ from r_i. All other senders (including s_{n+1} and s_{n+2}) are farther away. Now, we can show a lower bound for $\mathit{SINR}(r_i)$:

$$\begin{aligned} \mathit{SINR}(r_i) &= \frac{\frac{P_i}{d_{ii}^{\alpha}}}{\sum_{l_j \in L_i} \frac{P_j}{d_{ji}^{\alpha}}} &(5.5)\\ &\geq \frac{\frac{P_{\min}}{d_{ii}^{\alpha}}}{\sum_{l_j \in L_i} \frac{P_{\max}}{d_{ji}^{\alpha}}} &(5.6)\\ &\geq \frac{\frac{P_{\min}}{d_{\min}^{\alpha}}}{\frac{nP_{\max}}{(d(s_j,s_i)-d_{\min})^{\alpha}}} &(5.7)\\ &\geq \frac{1}{n}\frac{P_{\min}}{P_{\max}} \left(\left(1 + \left(\frac{P_{\max}}{P_{\min}} n\beta \right)^{\frac{1}{\alpha}} \right) - 1 \right)^{\alpha} &(5.8)\\ &= \beta &(5.9) \end{aligned}$$

As $\mathit{SINR}(r_i) \geq \beta$, it follows that transmission l_i is successful. □

Lemma 2. There exists a solution to a Partition problem $\mathcal{I}$ if and only if there exists a 2-slot schedule for $L_\mathcal{I}$. In the corresponding schedule, the senders $s_1, \ldots, s_n$ have to use transmission power $P_{\min}$ and the senders s_{n+1} and s_{n+2} have to use transmission power $P_{\max}$.

Proof. We start with showing that every solution to the Partition problem implies a valid 2-slot schedule with corresponding power assignment $\mathcal{P}$. Let us assume that we know two subsets $\mathcal{I}_1, \mathcal{I}_2 \subset \mathcal{I}$, whose elements sum up to $\sigma/2$. For every $i_j \in \mathcal{I}_1$, we assign the link l_j to the first time slot. Moreover, we assign l_{n+1} to the first time slot. The remaining links are assigned to the second time slot. We set $P_1 = P_2 = \cdots = P_n = P_{\min}$ and $P_{n+1} = P_{n+2} = P_{\max}$. We know from Lemma 1 that transmissions $l_1, \ldots, l_n$ are always successful, so let us focus on the receivers r_{n+1} and r_{n+2}. The situation is the same for both receivers, so it suffices to examine r_{n+1}. The signal power r_{n+1} receives from its sender s_{n+1} is

$$S_{r_{n+1}}(s_{n+1}) = \frac{P_{\max}}{\left(\left(\frac{2P_{\max}}{\beta\sigma}\right)^{\frac{1}{\alpha}}\right)^{\alpha}} = \frac{\beta\sigma}{2}.$$

Besides, r_{n+1} experiences from each sender s_j the interference

$$I_{r_{n+1}}(s_j) = \frac{P_{\min}}{\left(\left(\frac{P_{\min}}{i_j}\right)^{\frac{1}{\alpha}}\right)^{\alpha}} = i_j.$$

This results in a total interference of

$$I_{r_{n+1}} = \sum_{i_j \in \mathcal{I}_1} i_j = \frac{\sigma}{2}.$$

For the *SINR* at r_{n+1} we get

$$\mathit{SINR}(r_{n+1}) = \frac{S_{r_{n+1}}(s_{n+1})}{I_{r_{n+1}}} = \frac{\beta\sigma/2}{\sigma/2} = \beta.$$

Altogether, the constructed 2-slot schedule is valid for the given power assignment $\mathcal{P}$. It is easy to see that $\mathcal{P}$ is the only possible power assignment for the given schedule. If s_{n+1} or s_{n+2} would send with less power than $P_{\max}$, the corresponding *SINR* would fall below β. The same thing would happen if one of the other senders would use a transmission power above $P_{\min}$.

Finally, we have to show that we cannot find a 2-slot schedule for $L_\mathcal{I}$ if there does not exist a solution to the Partition problem. No solution to the Partition problem implies that for every partition of $\mathcal{I}$ into two subsets, the sum of one set is greater than $\sigma/2$. This means that, even if all senders other than s_{n+1} and s_{n+2} use power $P_{\min}$, the minimum transmission power possible, the interference at r_{n+1} in slot 1 or the interference at r_{n+2} in slot 2 exceeds $\sigma/2$. Even if s_{n+1} and s_{n+2} send with the maximum transmission power $P_{\max}$, the signal only arrives at the receivers with power $\beta\sigma/2$. Thus, the *SINR* of either l_{n+1} or l_{n+2} is below β and the transmission fails. □

From the above construction and observations, we can conclude the main theorem of this section:

Theorem 1. The SchedPC problem in the SINR$_G$ model is NP-hard.

5.4 One-Shot Scheduling with Power Control

The one-shot scheduling problem without power control in the SINR$_G$ model was proved to be NP-hard in [GOW07]. In the following, we extend this proof to the decision problem of one-shot scheduling with power control (OneShotSchedPC).

Again, we give a polynomial time reduction from an NP-complete problem, in this case from the well-known Knapsack problem [GJ79]. In the Knapsack problem, we are given a set $X = \{x_1, \ldots, x_n\}$ of items with *values* p_j and *weights* w_j. The goal is to pick the most valuable set of items that does not exceed a given overall weight W. Formally, we aim at finding a set $X' \subseteq X$, with $\sum_{x_i \in X'} w_i \leq W$, that maximizes $\sum_{x_i \in X'} p_i$.

We give a polynomial time reduction from Knapsack to OneShotSchedPC, similar to the one from Partition to SchedPC. The corresponding construction is depicted in Figure 5.2. Given an instance $\mathcal{I}$ of Knapsack, we use $n+1$ links $L_{\mathcal{I}} = \{l_1, \ldots, l_{n+1}\}$ and arbitrary but fixed minimum and maximum powers $P_{\min}$ and $P_{\max}$. The first n links $l_1, \ldots, l_n$ represent the items of the Knapsack problem. Link l_{n+1} enforces the connection between optimal solutions of OneShotSchedPC and optimal solutions of Knapsack.

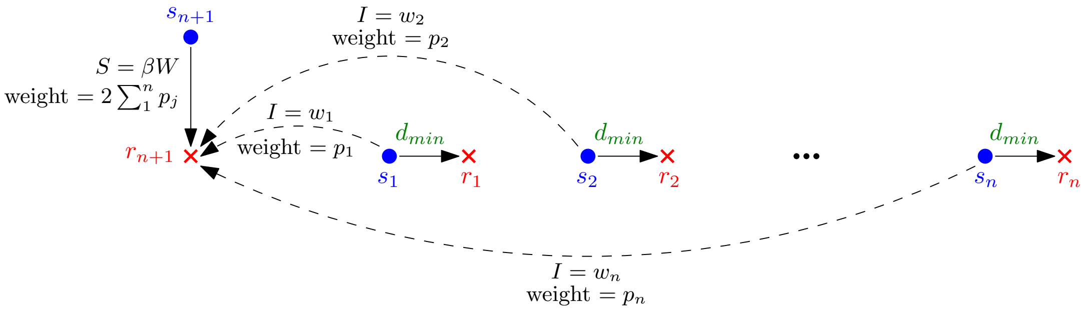


Figure 5.2: Reduction of Knapsack to OneShotSchedPC

Without loss of generality, we assume that all items have distinct integer weights. This time, we place the senders $s_1, \ldots, s_n$ such that the received power from s_i at the origin $(0,0)$ equals w_i if s_i transmits with power $P_{\min}$, i.e.,

$$pos(s_i) = \left(\left(\frac{P_{\min}}{w_i} \right)^{1/\alpha}, 0 \right), \quad \forall 1 \leq j \leq n \ .$$

Again, we make sure that $l_1, \ldots, l_n$ can be scheduled with power $P_{\min}$, regardless of all other links. Thus, we have to put r_i close enough to s_i. As in Section 5.3, the sender-receiver distance

$$d_{\min} = P_{\min}^{\frac{1}{\alpha}} \cdot \frac{\frac{1}{(w_{\max}-1)^{1/\alpha}} - \frac{1}{w_{\max}^{1/\alpha}}}{1 + (\frac{P_{\max}}{P_{\min}} n\beta)^{\frac{1}{\alpha}}} \ ,$$

where $w_{\max}$ is the largest weight in the Knapsack instance, is sufficient. This gives

$$pos(r_i) = pos(s_i) + (d_{\min}, 0) \ .$$

Thereafter, we have to place the additional link l_{n+1}. The receiver r_{n+1} is placed at $(0,0)$, i.e.,

$$pos(r_{n+1}) = (0,0) \ ,$$

and we place the sender s_{n+1} such that the received power at $(0,0)$ is βW if s_{n+1} sends with maximum power P_{max}:

$$pos(s_{n+1}) = \left(0, \left(\frac{P_{\max}}{\beta W} \right)^{1/\alpha} \right)$$

Finally, we have to assign the links appropriate weights. The links $l_1, \ldots, l_n$ are assigned the value of the corresponding item:

$$\textit{weight}(l_i) = p_i, \quad \forall 1 \leq i \leq n \tag{5.10}$$

The weight of special link l_{n+1} is set to twice the value of all items

$$\textit{weight}(l_{n+1}) = 2 \cdot \sum_{j=1}^{n} p_j,$$

in order to make sure that l_{n+1} is part of every optimal solution of the OneShotSchedPC instance.

Lemma 3. Let $(\mathcal{S}_\mathrm{OPT}, \mathcal{P}_\mathrm{OPT})$ with schedule $\mathcal{S}_\mathrm{OPT}$ and power assignment $\mathcal{P}_\mathrm{OPT}$ be an optimum solution of OneShotSchedPC instance $(L_\mathcal{I}, P_\mathrm{min}, P_\mathrm{max})$. Then, $(\mathcal{S}_\mathrm{OPT}, \mathcal{P}^*)$ with $P_1 = P_2 = \cdots = P_n = P_\mathrm{min}$ and $P_{n+1} = P_\mathrm{max}$ is also an optimum solution of $\mathcal{I}$.

Proof. If we set the transmission powers of $s_1, \ldots, s_n$ to P_min then we do not lose anything, as we defined d_min such that $l_1, \ldots, l_n$ are successful with power P_min, no matter which other senders are active simultaneously and no matter which transmission powers they use. Moreover, we also can set $P_{n+1} = P_\mathrm{max}$, as this does not influence the links $l_1, \ldots, l_n$ and it only increases $\mathit{SINR}(r_{n+1})$. So the schedule $\mathcal{S}_\mathrm{OPT}$ is also valid with respect to $\mathcal{P}^*$.

This means that we can assume without loss of generality that every optimum schedule has to be valid with power assignment $\mathcal{P}^*$. Now we use this property to show that every optimum schedule implies a valid solution to the Knapsack problem. In particular, we have to show that

$$\sum_{l_j \in \mathcal{S}_\mathrm{OPT}} w_j \leq W.$$

This follows from the SINR constraint of l_{n+1}:

$$\begin{aligned} \mathit{SINR}(r_{n+1}) &= \frac{S_{r_{n+1}}(s_{n+1})}{I_{r_{n+1}}} && (5.11)\\ &= \frac{P_\mathrm{max} / \left(\left(\frac{P_\mathrm{max}}{\beta W}\right)^{\frac{1}{\alpha}}\right)^{\alpha}}{\sum_{l_j \in \mathcal{S}_{OPT}} P_\mathrm{min} / \left(\left(\frac{P_\mathrm{min}}{w_j}\right)^{\frac{1}{\alpha}}\right)^{\alpha}} && (5.12)\\ &= \beta \cdot \frac{W}{\sum_{l_j \in \mathcal{S}_{OPT}} w_j} && (5.13) \end{aligned}$$

So, in order for l_{n+1} to be transmitted successfully, which means that $\mathit{SINR}(r_{n+1}) \geq \beta$, it must hold that $\sum_{l_j \in \mathcal{S}_\mathrm{OPT}} w_j \leq W$. Moreover, it is easy to verify that every solution X' of the Knapsack problem with value V implies a solution $(\mathcal{S}, \mathcal{P}^*)$ of the corresponding OneShotSchedPC problem with value $V' = V + 2 \cdot \sum_{j=1}^{n} p_j$. Thus, thanks to the choice of the weights in (5.10), every solution that maximizes the overall weight of our OneShotSchedPC instance at the same time maximizes the value of the corresponding solution to the underlying Knapsack problem. Altogether, we have shown that Knapsack is polynomial time reducible to OneShotSchedPC. □

Again, we conclude with the main theorem:

Theorem 2. The problem OneShotSchedPC in the SINR$_G$ model is NP-hard.

5.5 Final Remarks

Power control with a finite set of power levels. In modern hardware, the transmitters usually can choose their transmission power from a finite set of powers. Thus, the complexity of scheduling in this scenario has high practical relevance. The NP-hardness proofs given in the previous sections do not use the fact that the available transmission powers formed a continuous interval $[P_{\min}, P_{\max}]$, so they easily extend to any case where the available transmission powers have a minimum $P_{\min} > 0$ and a maximum $P_{\max} < \infty$. This includes the case of a finite set $\{P_{\min} = P_1, \ldots, P_k = P_{\max}\}$ of available power levels.

Proving NP-completeness. To show that Scheduling and OneShotScheduling lie in NP, one had to demonstrate that it is possible to check a solution to the considered problem in polynomial time by comparing all signal-to-noise ratios to the given threshold β. For computation models featuring real-valued arithmetic, this can be easily shown. However, even in the presumably easy case of $\alpha = 3$ and $\beta \in \mathbb{N}$, and all s_i and r_i placed on an integer grid, comparing signal-to-noise ratios to the threshold β boils down to the comparison of a sum of square roots of integers to an integer. Unfortunately, it is an open problem whether or not this can be done in polynomial time on a Turing machine, which is necessary to prove the above problems to be in NP [Che06]. At the time being, even the decision problem whether a set of nodes on the integer grid have a Euclidean minimum spanning tree with a length bounded by a given integer cannot be claimed to be in P. Thus, proving NP-completeness for SchedPC and OneShotSchedPC also calls for this gap in complexity theory to be closed.

5.6 Summary

We have shown that scheduling with power control and one-shot scheduling with power control in the geometric SINR_G model are NP-hard in situations where the transmission powers are bounded. This also includes the case where the senders are allowed to choose their power from a finite set of powers. To our knowledge, the complexity of the more general problem with completely arbitrary transmission powers is still unresolved. However, from a practical point of view, the variant with bounds on the transmission powers is more relevant, as the hardware usually sets an upper limit on possible transmission powers, and the omnipresent background noise defines a lower limit on any reasonable transmission power.

Part II

Localization

Chapter 6

Connectivity-based Detection of Network Boundaries

In this chapter, we study the connectivity-based detection of holes and boundaries in large sensor networks. A new distributed algorithm is presented, which enables a node to decide autonomously whether it is a boundary node. For this, only connectivity information of the node's local 2-hop neighborhood is required. The algorithm is evaluated in extensive simulations, and qualitative and quantitative comparisons with several previous approaches are presented.

6.1 Introduction

There are many applications in wireless sensor networks that require a certain knowledge of the underlying network topology, especially about outer boundaries of the network or areas within the network with low node density. Some examples where such information is very useful are intrusion detection, data gathering [WGM06], services like efficient routing within the network [FGG04, RRP$^+$03], or event detection [DLL09]. In many situations, holes can also be considered as indicators for insufficient coverage or connectivity. Especially in dynamic settings, where nodes can run out of power, fail, or move, an automatic detection of holes and boundaries is thus inevitable.

For this reason, several boundary recognition algorithms have been developed previously. However, most of them have certain disadvantages. Some algorithms rely on oversimplified assumptions concerning the communication model, others require knowledge about absolute or relative node positions, which is usually not available in large-scale sensor networks. Additionally, there are algorithms that are not distributed or require information exchange over long distances, so they do not scale well with network size. And those algorithms that solely work locally usually produce many misclassifications. Furthermore, many of the existing algorithms are too complex for an actual implementation on real sensor nodes. So there is still demand for simple and efficient algorithms for boundary recognition. In this chapter, we present a novel algorithm for connectivity-based boundary recognition.

6.1.1 Related Work

Since there is a wide range of applications that require boundary detection, there is an equally large number of approaches to detect and classify holes. Based on the underlying ideas, the approaches can be classified roughly into three categories.

Geometrical approaches use information about node positions, distances between nodes, or angular relationships to detect network holes and boundaries. As a consequence, these approaches are limited to situations where GPS devices or similar equipment are available. Unfortunately, in many realistic scenarios this is not the case. Examples for geometrical approaches are Fang *et al.* [FGG04], Martincic *et al.* [MS04], and Deogun *et al.* [DDHG05].

Statistical approaches try to recognize boundary nodes by low node degree or similar statistical properties. As long as nodes are smoothly distributed, this works quite well as boundary nodes usually have less neighbors than interior nodes. However, as soon as node degrees fluctuate noticeably, most statistical approaches produce many misclassifications. Besides, these algorithms often require very high average node degrees. Prominent statistical approaches are the algorithms by Fekete *et al.* [FKKL05, FKP+04] and Bi *et al.* [BTG+06].

Topological approaches concentrate on information given by the connectivity graph and try to infer boundaries from its topological structure. The algorithm of Kröller *et al.* [KFPF06] works by identifying complex combinatorial structures called flowers. Under certain assumptions about the communication model, such flowers exist with high probability if the average node degree is above 20. The algorithm requires that every node knows its 8-hop neighborhood. Funke [Fun05] and Funke *et al.* [FK06] describe algorithms that construct iso-contours and check whether those contours are broken. If a node recognizes that a contour is broken, it classifies the corresponding contour end-points as border nodes. The first algorithm requires that the whole network is flooded starting from some seed nodes. The second algorithm works distributed, based on 6-hop neighborhoods. An algorithm that works well even in networks with low average node degree is given by Wang *et al.* [WGM06]. The algorithm involves several steps, some of which require that the whole network is flooded. The centralized methods proposed by Ghrist *et al.* [GM05] and De Silva *et al.* [DSG06] detect holes by utilizing algebraic homology theory. In [SSG+08, SSGM10], Saukh *et al.* propose an algorithm that tries to identify certain patterns in the neighborhood of a node. Under certain conditions, they can guarantee that all nodes that are classified as inner nodes lie inside of the network. The algorithm is distributed and every node only needs information of its h-hop neighborhood. The radius h depends on the node density. For low density, $h = 6$ is used. For higher densities, it is possible to use smaller neighborhoods. A recent distributed algorithm by Dong *et al.* [DLL09] is especially aimed at locating small holes.

In previous work, there exist several classification schemes for boundary detection. Until recently, most boundary (or hole) definitions were based on an embedding of the connectivity graph. In [FGG04], Fang *et al.* determine the Delaunay triangulation of the embedded connectivity graph and remove edges of length greater than one. They classify faces of this reduced Delaunay graph with at least four edges as holes of the network. Boundary nodes are those nodes that induce these faces. In [KFPF06], Kröller *et al.* define boundaries according to a decomposition of the plane into faces based on the embedded connectivity graph. A face is called a hole if the circumference of its convex hull exceeds a minimum value. Since vertices of a face usually do not correspond to network nodes, the authors define boundary nodes to be the nodes on a cycle in the network graph surrounding this face. The approaches by Fekete *et al.* [FKKL05, FKP+04] apply a basic boundary definition for the continuous case. Given a set of holes, a closed cycle is called a boundary if it separates the area of a hole from the area occupied by the network. A mapping of the continuous boundary to network nodes is not defined. Saukh *et al.* [SSGM10] classify a node as boundary node if there exists any feasible embedding of the connectivity graph in which this node is located on the boundary of the embedded graph. The boundary definition proposed by Dong *et al.* [DLL09] uses topological properties. A cycle in the connectivity graph is defined to be a topological boundary if, given an arbitrary embedding of the graph, the embedded cycle can be continuously transformed into a boundary of the embedded graph.

6.1.2 Contributions

We examine a novel boundary recognition algorithm that enables a node to decide solely based on the *connectivity information* of its *local neighborhood* whether it is a boundary node. The basic principle of the algorithm is that every node infers from connectivity information of its 2-hop neighborhood whether it is enclosed by a closed circle of nodes that are exactly 2 hops away. If such a circle exists, it is used as a witness that the node is not a boundary node. The difficulty is to decide whether such an enclosing circle exists, using only connectivity information. We show that this can be checked very efficiently in a distributed fashion.

The presented algorithm has several benefits over existing approaches. Unlike many other algorithms, it is strictly local and suited for distributed application. It uses solely connectivity information, so no information about absolute or relative node positions is necessary. The algorithm also makes little assumptions about the communication model and it works even well if the underlying network is not a unit disk graph. Another very important aspect is that it is much easier to understand and implement than many existing approaches and that it is very robust to non-uniform node deployment and variations in node degree. Finally, the algorithm is equally well suited to detect extensive boundaries or small network holes that can occur when single nodes fail or move.

In extensive simulations, the algorithm is compared quantitatively and qualitatively to several other well-known algorithms. In order to enable an objective comparison, we give an intuitive definition of *network holes* and classify network nodes into three separate groups–*mandatory boundary nodes*, *optional boundary nodes*, and *interior nodes*. The classification is defined based on the position of a node relative to network holes and borders. Our simulations show that the presented algorithm, despite its simplicity, outperforms the other algorithms in most scenarios by detecting a higher percentage of boundary nodes correctly and, at the same time, misclassifying less interior nodes.

6.2 Models and Definitions

6.2.1 Network Model

We assume that the nodes of the sensor network are distributed in the two-dimensional Euclidean plane. The *connectivity graph* $\mathcal{C}(V, E)$, with graph nodes $v \in V$ corresponding to sensor nodes and graph edges $(u, v) \in E;\ u, v \in V$ representing communication links between sensor nodes, defines which of the sensor nodes can communicate directly with each other. An *embedding* $p : V \to \mathrm{I\!R}^2$ assigns two-dimensional coordinates $p(v)$ to each node $v \in V$. For easier reading, we normalize distances to the maximum possible communication distance between sensor nodes.

Communication model. A *communication model* defines under which conditions a pair of sensor nodes is able to communicate directly with each other. For our simulations, we use two simple communication models which are frequently used in the context of boundary recognition algorithms. In the *unit disk graph* (UDG) model, it is assumed that two sensor nodes $u, v \in \mathcal{C}$ can communicate with each other, i.e., there exists a communication link between them, if their distance $|p(u)p(v)|$ is at most 1. In the *d-quasi unit disk graph* (d-QUDG) model, sensor nodes $u, v \in \mathcal{C}$ can communicate reliably if $|p(u)p(v)| \leq d$ for a given $d \in [0, 1]$. For $|p(u)p(v)| > 1$ communication is impossible. For $d < |p(u)p(v)| \leq 1$, we assume that communication is possible with a probability of 50%.

Of course, in reality things are much more complicated and both UDG model and d-

QUDG model are extreme abstractions. However, we believe that they offer a reasonable starting point to compare the basic properties of different boundary recognition algorithms under well-defined conditions.

Node distribution. For the spacial distribution of network nodes, we consider two different strategies: Using *random placement*, nodes are placed uniformly at random on the plane. This models situations where sensor nodes are arbitrarily scattered in the environment, e.g., by throwing them from a plane.

Using *perturbed grid placement*, the nodes are placed on a grid with grid spacing 0.5 and subsequently translated by a uniform random offset taken from $[0, 0.5]$ in both dimensions. This models deployments where sensor nodes are placed in a regular pattern without the need to closely watch the exact placement. Compared with random placement, perturbed grid placement guarantees a more uniform node distribution with less variance in node degree. Perturbed grid placement is frequently used to examine boundary recognition algorithms.

6.2.2 Hole and Boundary Model

For a simulation-based comparison of boundary recognition algorithms, well-defined hole and boundary definitions are required. In this work, we take a practical look at what to label as holes and boundary nodes. In short, we call large areas with no communication links crossing them *holes* and nodes on the borders of these areas *boundary nodes*.

Hole definition. For our hole definition, we consider the actual embedding of the given sensor network. All faces induced by the edges of the *embedded connectivity graph* $p(\mathcal{C})$ are hole candidates. Similarly to [KFPF06], we define holes to be faces of $p(\mathcal{C})$ with a circumference of at least h_{min}. Figure 6.1 depicts a hole according to our definition. Note that the exterior of the network is an infinite face. Thus, it is regarded as a hole for the purpose of computation and evaluation.

Of course, the actual embedding of the sensor network is not known to the boundary recognition algorithms and only used for the final evaluation of the computed classifications.

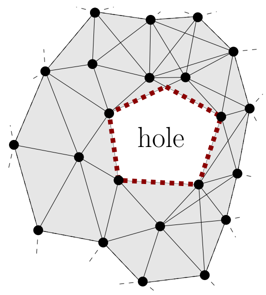


Figure 6.1: Hole definition. The hole border is shown by the dashed line.

Boundary node definition. As seen in Figure 6.2, hole borders and node locations do not have to align. Thus, there exists the problem which nodes to classify as boundary nodes. For example, it can be argued whether nodes A and B should be boundary nodes or not. One can find arguments for both possibilities, and it might depend on the application which definition is preferable. To alleviate this problem, we classify nodes into three categories:

- *Mandatory Boundary Nodes.* Nodes that lie exactly on the hole border are boundary nodes.
- *Optional Boundary Nodes.* Nodes within maximum communication distance of a mandatory node can be called boundary nodes, but do not have to be.
- *Interior Nodes.* All other nodes should not be classified as boundary nodes.

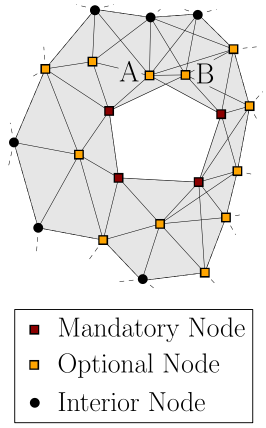


Figure 6.2: Boundary node classification.

The resulting node classification is shown in Figure 6.2. Mandatory boundary nodes form thin bands around holes, interrupted by structures like for nodes A and B before. Together with the optional boundary nodes, they form a halo around each hole. Any point within the halo is at most one maximum communication distance away from the border of the enclosed hole. A sample classification is depicted in Figure 6.3.

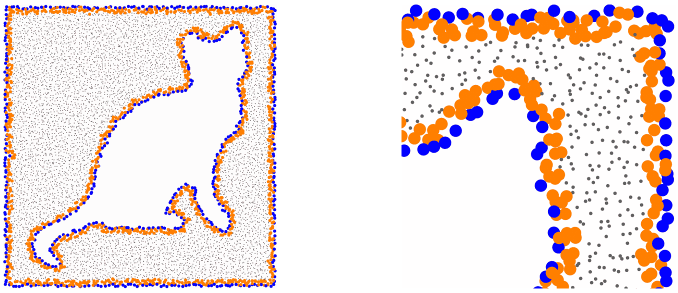


Figure 6.3: Node classification. Border outline of mandatory nodes (large, blue), halo of optional nodes (large, orange), interior nodes (small, black). Full network and magnified upper right corner are shown.

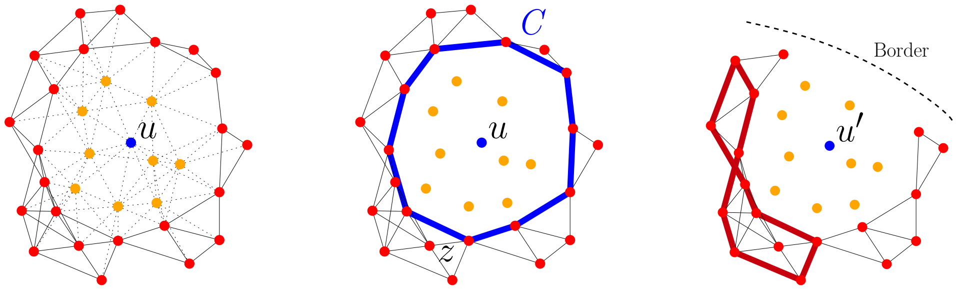
In our opinion, this distinction of mandatory and optional boundary nodes enables a fairer comparison between different boundary recognition techniques, as it does not lay down how to classify the controversial optional boundary nodes. Of course, there might be scenarios where a completely different classification is more appropriate, but the presented classification should cover a reasonable range of interesting applications.

6.3 Enclosing Circle Boundary Recognition

We now present a distributed algorithm that enables a node to detect reliably if it is surrounded by other nodes, using only connectivity information of its local 2-hop neighborhood. The basic idea of this *enclosing circle boundary recognition* (EC-BR) algorithm is as simple as efficient. The node ignores its direct neighbors and considers only nodes that are exactly two hops away. For a node u, we denote the corresponding node set as $N_u^{2\setminus 1}$ and the induced subgraph as $G_u^{2\setminus 1} = (N_u^{2\setminus 1}, E_u^{2\setminus 1})$. Based on the connectivity information in $G_u^{2\setminus 1}$, the node tries to decide if it is surrounded by a closed path C. If such a closed path exists, called *enclosing circle* in the following, the node can be sure that it is not a boundary node. Otherwise this is seen as an indication that the node lies somewhere near a hole or border (compare Figure 6.4).

6.3.1 Detection of Enclosing Circles

Knowing the actual node positions, it would be easy to decide if an enclosing circle exists. However, we do not have this information and we also do not want to reconstruct node positions as this would be computationally expensive. So, how can we distinguish between enclosing circles as the one in Figure 6.4b and non-enclosing circles such as the one in Figure 6.4c? The length of the circle is obviously no sufficient criterion as both circles have the same length and only the first one is enclosing. Fortunately, there is a structural difference between both types of circles: the circle in Figure 6.4b encloses the hole like a tight rubber band and there is no way to split it into smaller circles by adding edges from $E_u^{2\setminus 1}$, whereas it is quite easy to find edges in Figure 6.4c that could be used to split the circle into shorter circles. More formally, the first circle has the property that for each pair



(a) (b) (c)

Figure 6.4: Basic idea of EC-BR. (a) 2-hop neighborhood of a node u. (b) Enclosing circle C. (c) Boundary node u' without enclosing circle. A non-enclosing circle is highlighted.

v, w of nodes on the circle, the shortest path between them using only circle edges is also a shortest path between them in $G_u^{2\setminus 1}$. Now we try to find a preferably long circle with this property in $G_u^{2\setminus 1}$. If we assume a unit disk graph and node u is enclosed by nodes of $G_u^{2\setminus 1}$, there has to exist such a circle of length at least 7. On the other hand, if u lies somewhere near a hole then u is not fully enclosed by other nodes and it is highly unlikely that a large circle with the aforementioned property exists.

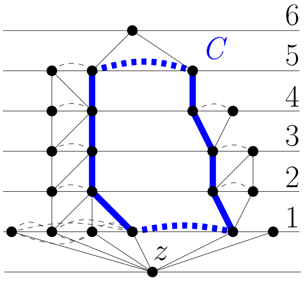
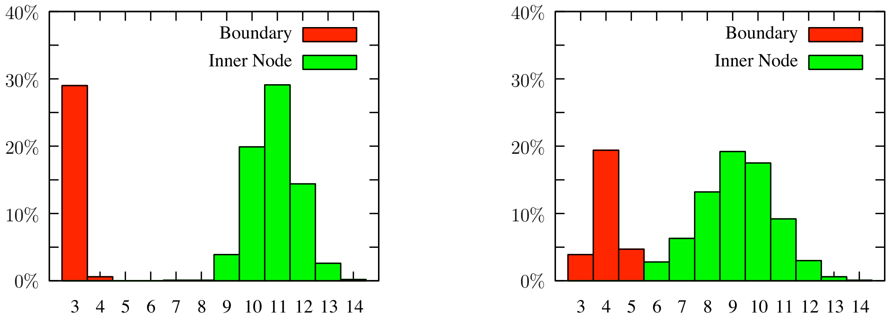


Figure 6.5: Modified breadth-first search for efficient circle detection.

To find a maximum circle with the given property, we can use a modified breadth-first search. The corresponding search tree for $G_u^{2\setminus 1}$ of Figure 6.4b is depicted in Figure 6.5. We start the search from a random node z in $G_u^{2\setminus 1}$ with maximum degree. In every step of the search, we maintain shortest path lengths for all pairs of visited nodes. When a new edge is traversed, there are two possibilities: either a new node is visited, or a previously encountered node is revisited. In the first case, we just set the shortest path distances between the old nodes and the new node. This can be done efficiently, as all distances can be directly inferred from the distances to the parent node. In the second case, we found a new circle in $G_u^{2\setminus 1}$. The length of the circle is the current shortest path between the endpoints v and w of the traversed edge $e = (v, w)$ plus one. Subsequently, we update the shortest path information of all nodes. During the search, we keep track of the maximum length of a circle encountered so far. Depending on the maximum length that was found during the search, the considered node is classified as either a boundary node or an inner node. For every edge in $G_u^{2\setminus 1}$, the update of pairwise distances has to be performed at most once. Thus, the asymptotic time complexity of this approach is in $O(mn^2)$, with $m = |E_u^{2\setminus 1}|$ and $n = |N_u^{2\setminus 1}|$. Later in this section we will show how this complexity can be reduced to $O(m)$.

Figure 6.6 depicts histograms of maximum circle lengths in our simulations with networks based on unit disk graphs and quasi unit disk graphs. There are apparently two very well defined peaks, corresponding to nodes with and without enclosing circles. Based on this distributions, we classify all nodes u that have a maximum circle in $G_u^{2\setminus 1}$ with length of at least 6 as inner nodes and all other nodes as boundary nodes. Our simulations indicate that this statistical classification into nodes with and without enclosing circle works extremely well for both UDGs and QUDGs. Later on, we will see how good this correlates with being in the interior or being on the boundary of the network.

It is also noteworthy that this kind of classification is extremely robust to variations



(a) UDG

(b) QUDG

Figure 6.6: Distribution of maximum circle lengths for UDGs and QUDGs.

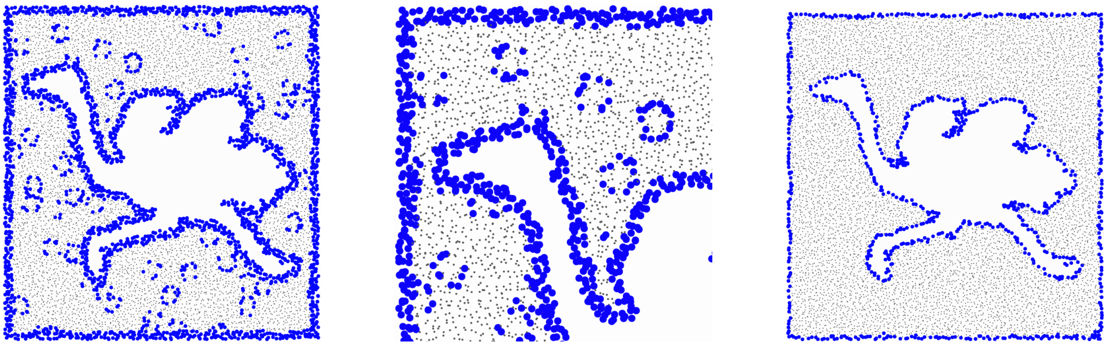
in node degree: it does not matter whether $N_u^{2\setminus 1}$ consists of a small number of nodes or hundreds of nodes, the same threshold 6 on the maximum circle length can be used to distinguish interior nodes from boundary nodes. The classification stays the same, as long as we assume that the node density is sufficiently high so that inner nodes are actually surrounded by other nodes. This robustness distinguishes EC-BR from other existing statistical approaches.

Enclosing circle detection in linear time. The enclosing circle detection of EC-BR runs distributed on every single node and each node only has to consider its 2-hop neighborhood, so its runtime is virtually uncritical. Nevertheless, for the sake of completeness, we mention how the search can be improved to run in time $O(m)$, with $m = |E_u^{2\setminus 1}|$, if the underlying network has properties of a quasi unit disk graph. The key insight is that it does not make any difference for the classification if a node u is enclosed by thousands of nodes or by just enough nodes so that the circle is closed. Thus, in a first step, node u can filter $N_u^{2\setminus 1}$ to a small set of representatives. By considering each edge in $E_u^{2\setminus 1}$ once, a maximal independent set $\mathcal{I}$ of $G_u^{2\setminus 1}$ can be computed in time $O(m)$. Based on packing arguments, the number of nodes in set $\mathcal{I}$ is bounded by a small constant for QUDGs. By iterating again over all edges, we assign each node v to the nodes of $\mathcal{I}$ that v is connected to. Next, two nodes in $\mathcal{I}$ are connected if there exists an edge $(v, w) \in E_u^{2\setminus 1}$ with v and w assigned to these two nodes. As the size of $\mathcal{I}$ is asymptotically independent of the network size, this can also be achieved in time $O(m)$. Now, node u is enclosed by nodes in $\mathcal{I}$ if and only if it was enclosed in $G_u^{2\setminus 1}$. Thanks to the constant size of $\mathcal{I}$, the time for the enclosing circle detection on $\mathcal{I}$ is asymptotically independent of the size of $G_u^{2\setminus 1}$. However, the classification thresholds have to be adjusted as the edges in the representative graph no longer correspond to 1-hop distances. Altogether, the enclosing circle detection can be done in time linear in the size of $E_u^{2\setminus 1}$.

6.3.2 Classification Result

Figure 6.7a shows an example of a classification with EC-BR. The large blue points correspond to nodes that have been classified as boundary nodes, the small dots to nodes that are believed to be inner nodes. Apparently, the nodes that are close to holes or the outer boundary are correctly identified. However, there are two aspects of the classification of EC-BR which attract attention.

One aspect is that the detected boundaries are rather broad, meaning that even nodes that are only in proximity to a hole or the outer boundary are classified as boundary



(a) Before refinement. (b) Magnification of upper left part. (c) After refinement.

Figure 6.7: Classification of EC-BR before and after refinement.

nodes. The reason for this is that in EC-BR, every node checks whether the nodes in 2-hop distance form a closed circle. And even for nodes that are almost one hop away from a hole or boundary such a closed circle does not exist. The result is that the recognized border has a thickness of about one hop.

The second striking aspect are the many small circles which do not belong to the large-scale boundaries. By looking at the magnification in Figure 6.7b, one can see that the marked circles enclose small holes in the network. We will see in the next paragraph how one can easily remove both kinds of artifacts, the wide borders and the circles around tiny holes in the network. However, in many situations exactly this kind of information might be of interest. The detection of small holes, for instance, can be used to detect node failures or areas of insufficient coverage. And having a broader border might increase the fault tolerance and makes it easier to distribute messages along the border as it is guaranteed to be connected. Additionally, neighboring boundary nodes can divide their workload and thus extend the lifetime of their batteries.

6.3.3 Refinement

Sometimes one might be only interested in large-scale boundaries and not in the small holes that occur in areas with low node density. For these situations, EC-BR can be extended with a very simple refinement, which removes most of the small holes and also makes the boundaries thinner. The basic insight behind the refinement is that a node which lies near a hole is surrounded by other nodes that are marked as boundary nodes and by the hole itself. So the node simply has to check whether a certain percentage γ of its neighbors are currently classified as boundary nodes. If this does not hold, the node changes its classification from being a boundary node to being an interior node. Under the idealized assumption that the connectivity graph is a unit disk graph, $\gamma = 100\%$ results in very precise boundaries. For more realistic communication models, a threshold $\gamma \approx 70\%$ is more reliable.

The effect of this simple refinement strategy is depicted in Figure 6.7c. Apparently, all nodes but the ones near large-scale holes are now classified as inner nodes and the border is very precise.

6.3.4 Detection of Large-Scale Holes

In some situations, one might only be interested in holes which exceed some given circumference r_{min}. Although EC-BR is not directly designed for such kinds of computations, it can be easily extended to this problem. One simple and efficient possibility is to use EC-BR to compute a set of boundary candidates. As long as r_{min} is larger than the circumference of a 1-hop neighborhood, one can expect that the wanted boundary nodes are among the boundary candidates computed by EC-BR. In a subsequent step, one can use the refinement step of the Multidimensional Scaling Boundary Recognition algorithm (MDS-BR), which is described in Section 6.4.2, to identify the nodes that are near holes with circumferences exceeding r_{min}. Of course, depending on r_{min} it might be necessary to consider a neighborhood with larger radius than 2 hops. However, as only those nodes have to be considered which have been identified by EC-BR as hole candidates, only information of a small subset of all nodes in the neighborhood has to be communicated. Thus, the presented approach can be used to recognize hole structures of arbitrary size with low communication expense.

6.3.5 Connected Boundary Cycles

Some applications require knowledge about connected boundary cycles. In its standard form, EC-BR does not provide this information, as boundary cycles cannot be determined locally. To compute connected boundary cycles using EC-BR, one can proceed as follows: First, EC-BR is used to compute a set of boundary candidates. Within the halo of boundary node candidates, a shortest path query is started from an arbitrary node to find an enclosing circle. As the halo consists of the one hop neighborhood of the boundary, we are guaranteed that a connected component around the hole exists. After a closed cycle is found, all nodes within the 1-hop neighborhood of the cycle are classified as interior nodes. This is repeated until all nodes either belong to a connected boundary cycle or they are classified as interior nodes. Special care has to be taken in the case that two large holes are less than one hop distance away from each other. In this case, the two boundary cycles have to share some nodes.

6.3.6 Misclassification of Nodes

For some approaches to the boundary recognition problem it is possible to prove theoretical guarantees concerning the classification quality. For example, one can show that under certain assumptions no boundary node is classified as an interior node, or that every node that is classified as a boundary node really lies near a hole or at least near an area with low node density. While it is definitely worthwhile to be able to give such theoretical guarantees, we decided to focus on the design of efficient heuristics. This has several reasons: To be able to provide theoretical guarantees, one usually has to pay some price. A minimum requirement are rather strict assumptions concerning the communication model. Additionally, it is often necessary to make further assumptions concerning the distribution of the sensor nodes, or to use a less intuitive definition of network holes. This alone is no big issue, as the algorithms still might work very well even in more realistic situations where the assumptions are not fulfilled. However, to enable the theoretical guarantees, the algorithms usually also have to consider a rather large neighborhood (e.g., 5 hops or even more) and to use more expensive operations. As our goal was to design efficient algorithms that are suited for real-world application in large-scale sensor networks, we decided to focus on heuristics which work well using as little information and communication as possible.

In the remainder of this section, we analyze under which circumstances nodes are misclassified by EC-BR. For this, it is important to keep in mind that the definition of misclassifications strongly depends on the considered application. There are situations where areas of low node density are to be detected, and other situations where only large-scale holes play a role. The hole definition used in this work includes rather small holes. This is only meaningful if the average node density is sufficiently high so that there is no excess of small holes. For low-density networks, another hole definition that aims at larger holes might be preferable.

Misclassification of boundary nodes. EC-BR classifies nodes based on the length of a longest cycle found by the enclosing circle detection. We now show that it is possible to construct degenerated settings for which a non-enclosing cycle is considered to be enclosing by EC-BR. Figure 6.8 shows the construction of such a cycle of length 7 which, assuming a UDG communication model, is wrongly assumed to be enclosing. The nodes have to be placed very carefully in order to prevent the emergence of communication links that split the cycle into smaller cycles. This makes this kind of misclassification very unlikely. Moreover, an additional node located within the depicted cycle would most likely split the cycle and thus prevent the misclassification. Accordingly, for networks with higher node densities this kind of misclassification becomes less likely.

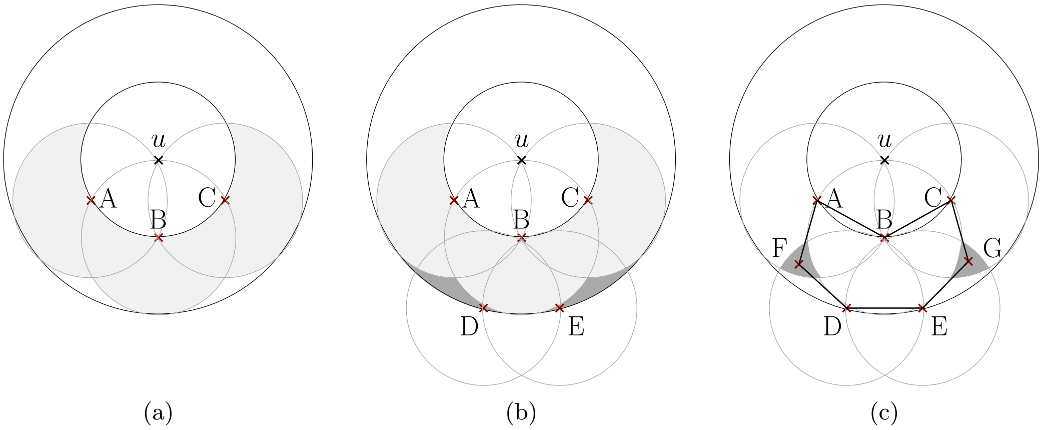


Figure 6.8: Example for a situation in which EC-BR wrongly classifies a circle to be enclosing. (a) Three nodes A, B, and C are placed such that they are slightly more than one communication distance away from node u, and both A and C have almost distance one to node B. (b) Two nodes D and E are added into the dark gray areas such that they are less than distance one away from each other, less than distance two away from u, and more than distance one away from node B. (c) Finally, two nodes F and G are placed into the gray areas such that F is less than distance one away from both A and D, G is less than distance one away from both C and E, and F and G are both more than distance one away from node B. The enclosing circle detection would wrongly classify the depicted circle of length 7 to be enclosing. (The nodes within 1-hop distance of u are omitted in this description. They have to be arranged such that the depicted nodes A-G are within 2-hop distance of u).

Misclassification of interior nodes. The second type of misclassification, classifying interior nodes as boundary nodes, occurs more frequently. This happens if the node density within the 2-hop neighborhood of a node is so small that no enclosing circle exists. Thus, misclassification of an interior node can usually be seen as an indication of insufficient node density. But as long as the overall node density is sufficiently high, this kind of misclassification can be easily fixed using the refinement step.

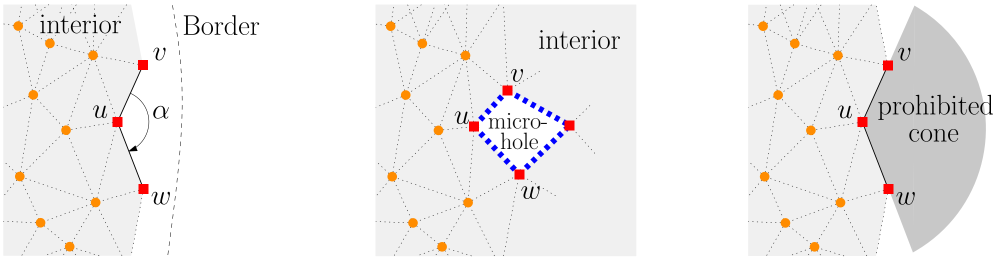
6.4 MDS Boundary Recognition

In [SVW11b], we also presented a second distributed algorithm, Multidimensional Scaling Boundary Recognition (MDS-BR), in which every node approximates the embedding of its neighborhood using multidimensional scaling [Tor52] and subsequently checks some angular conditions to decide whether it probably is surrounded by other nodes. In this work, we use MDS-BR mainly as an additional reference approach to EC-BR. This section gives a brief overview of the algorithm and the underlying ideas. For a more detailed study of the algorithm, including complexity, parameter analyses and possible extensions, we refer to [SVW11a, SVW11b].

6.4.1 Base Algorithm

To decide independently whether to classify itself as a boundary node, each node u first gathers the connectivity information of its 2-hop neighborhood N_u^2. The hop distances in N_u^2 are then used to approximate true distances between nodes. Using the approximated pairwise distances, multidimensional scaling is used to compute a two-dimensional embedding of $N_u^2 \cup \{u\}$. Based on this embedding, two conditions are checked:

The first condition states that the maximum opening angle α between two subsequent neighbors v, w of u in circular order and u must be larger than a threshold α_{min}. This situation is depicted in Figure 6.9a. This primary condition models the observation that boundary nodes exhibit a large gap in their neighborhood, while interior nodes usually are completely surrounded by other nodes.



(a) Minimum opening angle (b) Micro-holes (c) Prohibited cone

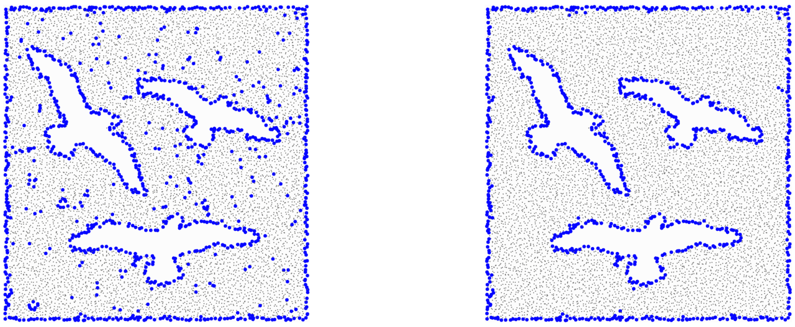
Figure 6.9: MDS-BR classification conditions.

The second condition is used to filter micro-holes framed by 4 nodes with a circumference of at most 4 maximum communication distances, such as the one depicted in Figure 6.9b. It is checked whether neighbors v and w of u have common neighbors other than u in the cone opened by (uv) and (uw) (cf. Figure 6.9c). If this condition is fulfilled, node u is classified as interior node. Of course, if such holes are to be detected, one can omit this additional condition.

Both conditions only require angular information, so any embedding algorithm yielding realistic angles between nodes could be used – one is not limited to MDS. Furthermore, as only very small graphs are embedded, it is not necessary to compensate for problems occurring in large graphs such as drifting or foldings. In particular, every node only has to compute the embedding of its 2-hop neighborhood, so the computational effort is rather low compared to many other approaches.

6.4.2 Refinement

Figure 6.10a shows a classification of the base algorithm of MDS-BR. Apparently, nodes around holes and the outer boundary are correctly classified. However, there is also some noise due to the detection of boundary nodes around small holes and misclassifications.



(a) Classification of the base algorithm

(b) Result after refinement

Figure 6.10: Classification result of MDS-BR on a sample network. Nodes that are classified as boundary nodes are highlighted.

To remove these artifacts, a refinement step can be used. Note that it is not possible to simply use the refinement of EC-BR, as MDS-BR does not have the property that mandatory boundary nodes are surrounded by a halo of nodes that are classified as boundary nodes. Thus, using the refinement of EC-BR would result in most mandatory boundary nodes being reclassified to be interior nodes.

Instead, we use the observation that interior nodes that are falsely classified as boundary nodes are usually isolated, with no or only few other nodes around them being classified as boundary nodes, while real boundary nodes usually form larger structures with many consecutive nodes classified as boundary nodes. The refinement is performed distributed on the current set of boundary node candidates. First, each boundary node candidate u gathers its r_{min}-hop neighborhood $\tilde{N}_u^{r_{min}}$ of other nodes marked as boundary nodes, where r_{min} is a free parameter. Then, u verifies if there exists a shortest path of at least r_{min} hops in $\tilde{N}_u^{r_{min}} \cup \{u\}$ that contains u. If no such path exists, u classifies itself as interior node. This approach removes boundary nodes that are not part of a larger boundary structure, with r_{min} specifying the desired size of the structure. Again, only connectivity information is required for the refinement.

The effect of this refinement can be seen in Figure 6.10b. Apparently, the noise could be successfully removed without negatively affecting the classification of mandatory boundary nodes.

6.5 Simulations

To assess the quality of boundary recognition algorithms, we use various simulations and consider the misclassification rates for both boundary nodes and interior nodes. In our opinion, it is extremely important to find a good tradeoff between misclassifications of interior nodes and boundary nodes. It is easy to design a simple algorithm that provably does not miss any boundary node, or never misclassifies an interior node as boundary node. This can be achieved by simply classifying all nodes as boundary nodes, or classifying all nodes as interior nodes. The difficulty thus lies in designing an algorithm that minimizes both kinds of misclassifications at the same time.

6.5.1 Simulation Setup

Network layout. We generate network layouts by iteratively placing nodes on an area of 50×50 maximum communication distances according to one of the distribution strategies described in Section 6.2: perturbed grid placement (pg) or random placement (rp). After each node placement, communication links are added according to the UDG or QUDG model. Nodes are added until an average node degree d_{avg} is reached. As we are not only interested in the detection of the outer boundary, we additionally apply hole patterns such as the ones in Figure 6.11 to generate areas without nodes. When a node is generated that would lie within a hole, it simply is discarded.

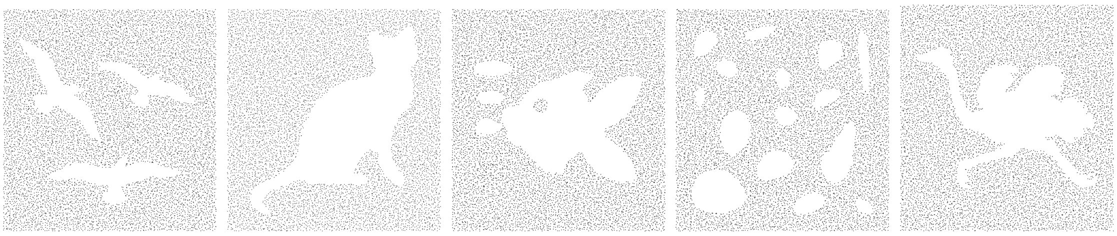


Figure 6.11: Hole patterns. Node distributions with perturbed grid placement and $d_{avg} = 12$.

For our simulations, we use different combinations of placement strategies, communication models and node densities. If not stated otherwise, the default layout uses perturbed grid placement, the UDG model, and average node degree $d_{avg} = 12$.

Considered algorithms. To analyze the performance of EC-BR, we compare it to several other boundary recognition algorithms. For the quantitative analysis, we use the algorithm by Fekete *et al.* [FKP$^+$04] (labeled Fekete04), the centralized and distributed algorithms by Funke [Fun05] and Funke *et al.* [FK06] (labeled Funke05 and Funke06, respectively), and MDS Boundary Recognition (MDS-BR) [SVW11b]. In addition, we show qualitative comparisons of these algorithms and the algorithms by Wang *et al.* [WGM06] and Saukh *et al.* [SSGM10]. We apply our own implementation of these algorithms according to their description in the respective publications and use the recommended parameters. For the refinement of EC-BR, we use the threshold $\gamma = 100\%$ if not stated otherwise.

Unfortunately, the number of existing approaches makes it impossible to include all of them in our comparison. Thus, we tried to select algorithms that assume similar conditions and constraints as our approaches. Many of the other existing algorithms are not directly comparable as they use additional information like absolute or relative node positions, they require connectivity information of large neighborhoods or even the whole network, they strongly rely on certain network properties such as much higher average node degrees, or they require expensive operations like flooding the whole network or centralized computation. Some of these approaches might even achieve better classifications by utilizing more information or more expensive operations. However, our goal in this work is to show that solely connectivity information of nearby nodes and a relatively low average node degree are sufficient to achieve very good classification results.

Measurement procedure. Each simulation scenario is evaluated 100 times for each of the hole patterns in Figure 6.11. Every face with circumference $h_{min} \geq 4$ is considered a hole, e.g., a square of edge length 1 with no communication link crossing it. In the quantitative overviews, mean misclassification ratios (false negatives) in percent are stated. For optional boundary nodes, we give the percentage of nodes classified as interior nodes. The best results for each setup are highlighted in bold.

As stated before, different boundary recognition approaches use slightly different definitions for holes and boundaries. Thus, it is possible that these approaches perform worse under the definition used in our comparison. Still, we believe that a simulation-based comparison such as the one presented here is necessary to get a feeling for the similarities and differences between the approaches. In our opinion, the used distinction of mandatory and optional boundary nodes helps to allow for a fairer comparison. Only nodes that are immediately at the boundary or at least one hop away from the boundary are considered for determining the classification quality. For the other nodes, it might depend on the scenario whether they should be classified as boundary nodes or not, thus they are not rated.

6.5.2 Visual Comparison

We start with a visual comparison of the classification results of the considered algorithms in Figure 6.12. In Figure 6.12a, the set of mandatory boundary nodes for the considered scenario is highlighted as bold points, all other nodes are plotted as thin dots. To compute the set of mandatory boundary nodes, the given embedding of the sensor network was used. Figure 6.12b shows the classification of EC-BR without refinement. One can clearly see that nodes within 1-hop distance of a large-scale hole or the outer boundary are reliably classified as boundary nodes. Additionally, there are several small circles around small areas with low node density. The effect of the refinement of EC-BR is shown in Figure 6.12c. After the refinement, the algorithm returns precise outlines of inner and outer borders with almost no artifacts. While originally most optional nodes were classified as boundary nodes, now most of them are classified as interior nodes.

Funke06 also correctly identifies the boundaries with some artifacts. Similar to EC-BR, the apparent noise is caused by small holes which are surrounded by marked nodes. The results of Fekete04 and Funke05 show many artifacts. Additionally, the algorithms failed to detect some mandatory boundary nodes. The global algorithm by Wang *et al.* [WGM06] produces closed boundary circles with no artifacts. Due to the nature of the algorithm, marked boundaries are not always at the true border but shifted inwards. Because of this, the misclassification rate for mandatory boundary nodes would be rather high. However, depending on the scenario this might be no problem, and the algorithm works much better in networks with very low average node degree than most other algorithms, including EC-BR. The algorithm of Saukh *et al.* also produces a very interesting classification. To some extent the result is similar to EC-BR without refinement, as nodes that are only in proximity to a border are also classified as boundary nodes. For the given example the algorithm produces no artifacts, like EC-BR with refinement. Unfortunately, the good classification also has its price, as the algorithm requires an immense computational effort and connectivity information from a larger neighborhood. The classification of MDS-BR, which is depicted in Figure 6.12i, is similar to the one of EC-BR with refinement and contains almost no misclassifications of mandatory boundary nodes or interior nodes.

Influence of the refinement. In order to convey a feeling for the influence of the refinement of EC-BR, we present some additional classification results before and after refinement in Figure 6.13. Later on, in Section 6.5.7, we will deal with the questions whether the refinement also is suited to improve the classification results of the other algorithms.

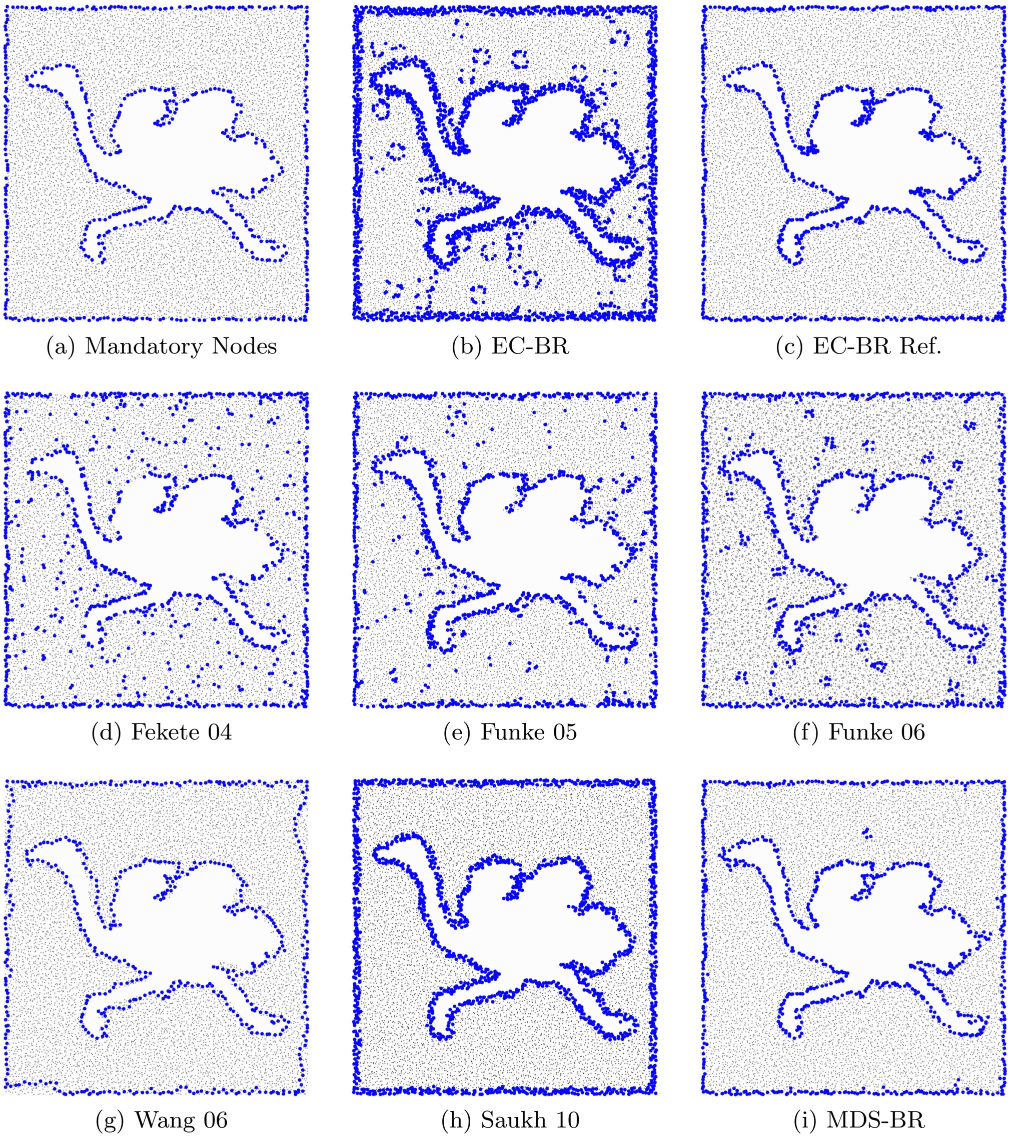

(a) Mandatory Nodes
(b) EC-BR
(c) EC-BR Ref.
(d) Fekete 04
(e) Funke 05
(f) Funke 06
(g) Wang 06
(h) Saukh 10
(i) MDS-BR

Figure 6.12: Visual comparison of several algorithms for boundary detection.

6.5.3 Network Density

In this section we consider how the classification performance depends on the average node degree d_{avg} of the network. A visual impression is given for the considered algorithms in Figure 6.14. Obviously, for all algorithms the number of interior nodes classified as boundary nodes increases rapidly for average node degrees 10 and below. The reason is that for low node densities many small-scale holes emerge. Thus, most nodes are close to a hole and (correctly) classified as boundary nodes. Accordingly, for such sparse networks different algorithms and boundary definitions might be more appropriate.

For node degrees 11 and above, EC-BR classifies almost only nodes close to the large-scale holes or the outer border as boundary nodes, while most other algorithms show a lot of “noise”, i.e., interior nodes falsely marked as boundary nodes. Furthermore, especially Fekete04 and Funke05 have problems detecting the boundary correctly.

In the classifications of Funke06, there are many small circles of nodes that are marked as boundary nodes. Similar but less such structures show up in the classifications of MDS-

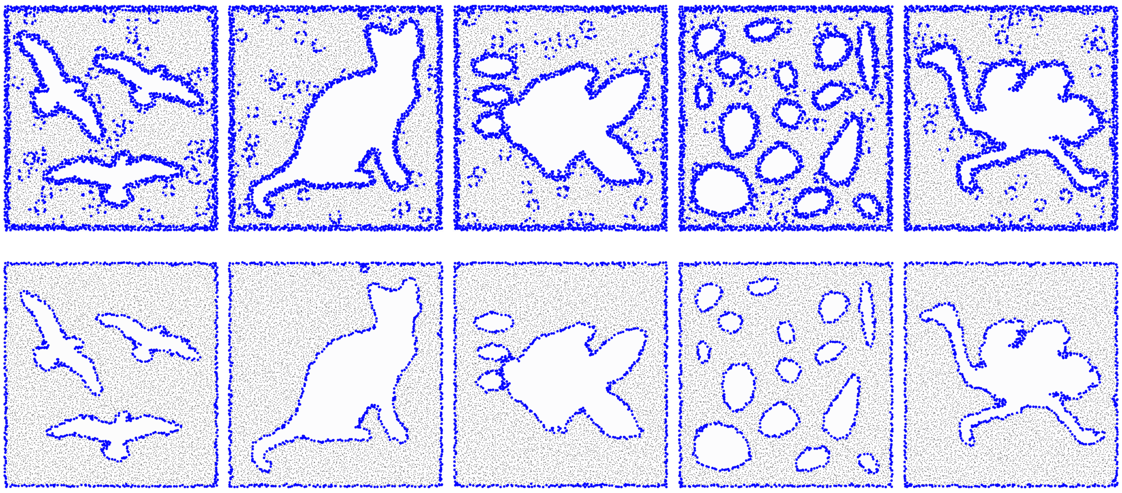


Figure 6.13: Examples for EC-BR on networks with average node degree 12 and different hole patterns. The top row shows the classification before refinement, the bottom row the classification with refinement.

BR. The circle shape is a clear sign that the corresponding classifications are no random misclassification, but that the nodes actually surround a larger area without nodes. EC-BR without refinement would also mark the corresponding areas, but the refinement step reverts the corresponding classifications, as they do not cover large enough areas. The algorithm of Saukh *et al.* again produces nice classifications for average node degrees of 11 and more. Again, all nodes in proximity to a large-scale hole are correctly classified as boundary nodes.

In Table 6.1, we present a quantitative comparison of misclassifications. The numbers state the percentage of false classifications for mandatory boundary nodes and interior nodes. Networks with different average node degrees d_{avg} are considered. For example, for an average node degree of 12, EC-BR without refinement classifies on average 0% mandatory nodes as interior nodes and 7.5% interior nodes as boundary nodes. After the refinement, the misclassifications of interior nodes are resolved, but 0.4% of the mandatory nodes have been wrongly reclassified to be interior nodes.

	Mandatory					**Interior**				
	9	12	15	18	21	9	12	15	18	21
EC-BR	2.1	**0.0**	**0.0**	**0.0**	**0.0**	54.8	7.5	3.8	2.2	1.6
EC-BR Ref	4.4	0.4	0.6	1.0	1.3	**7.1**	**0.0**	**0.0**	**0.0**	**0.0**
Fekete04	34.7	14.2	6.7	3.4	1.9	9.8	3.5	7.2	6.9	2.5
Funke05	16.6	6.3	5.7	5.1	5.0	21.7	3.5	2.0	1.3	0.9
Funke06	39.7	13.8	16.6	18.9	20.9	13.0	3.4	1.4	0.6	0.3
MDS-BR	**1.9**	2.9	3.5	3.8	3.9	19.0	0.7	0.3	0.1	**0.0**

Table 6.1: Misclassification ratios (false negatives) in percent for average node degrees between 9 and 21.

Like in the visual comparison, it turns out that the used hole definition is inadequate for networks with an average node degree as low as 9, as in this case many small holes occur which fall below the used threshold for interesting holes. For more dense networks, EC-BR without refinement classifies almost all mandatory nodes correctly, but according to the used hole definition, a small percentage of interior nodes is falsely classified to be

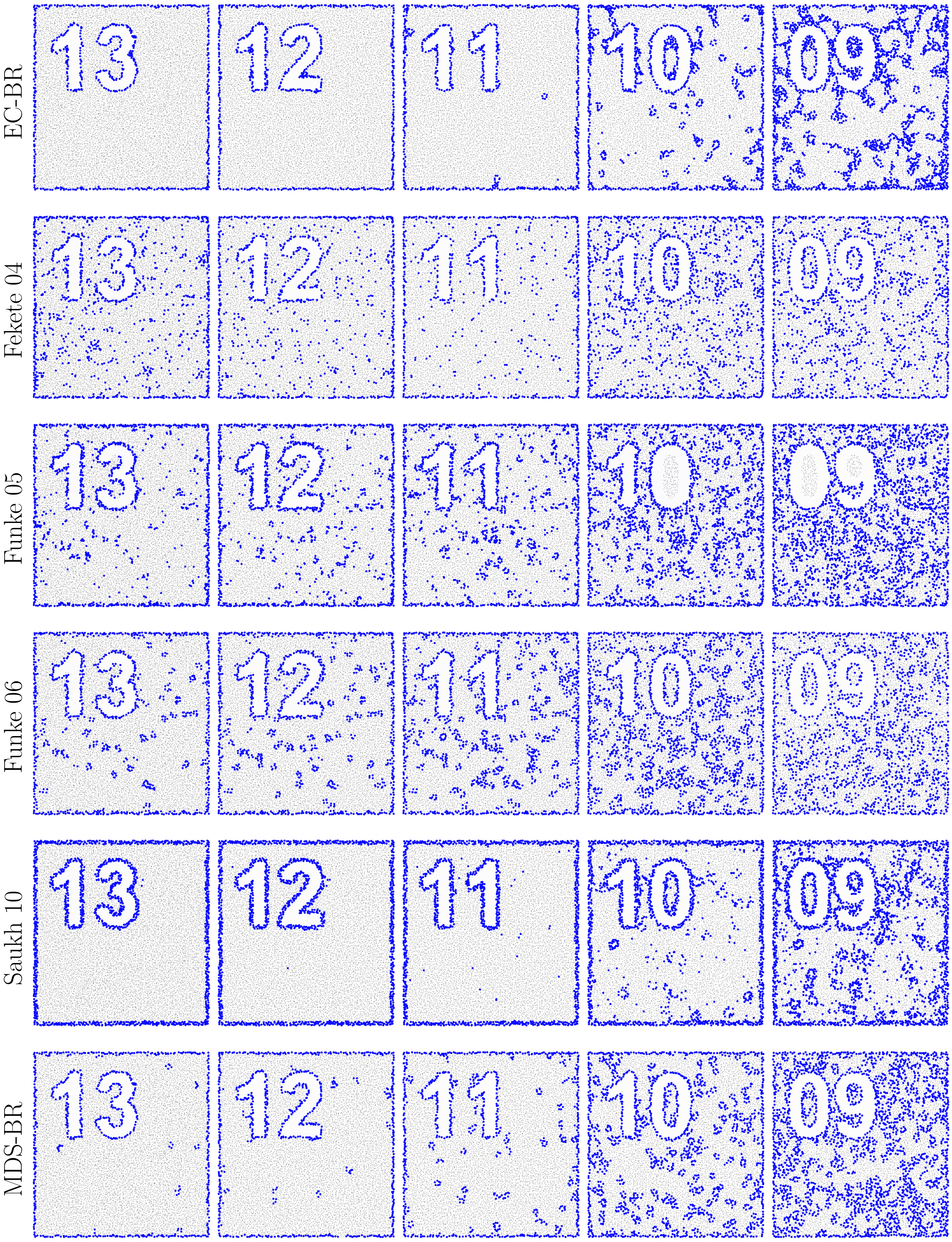


Figure 6.14: Influence of average node degree on the classification result. The numbers in the pictures state the average node degree of the respective network.

boundary nodes. The refinement fixes the misclassification of interior nodes very well, at cost of some misclassifications of mandatory nodes. Overall, the simulations indicate that for the examined hole definition EC-BR with refinement dominates the other considered approaches.

Table 6.2 shows the percentage of optional nodes that are classified as interior nodes. Recall that optional nodes are the nodes within one maximum communication distance of a hole that are not mandatory boundary nodes. As mentioned before, it mainly depends on the application whether optional nodes are rather boundary nodes or interior nodes. Here, the results for EC-BR are particularly interesting. Before refinement, the algorithm classifies almost all of the optional nodes as boundary nodes, while still providing a strict separation to the interior nodes. After refinement, EC-BR classifies less optional nodes as boundary nodes than all other approaches, while still recognizing almost all mandatory boundary nodes correctly. So depending on the scenario, one can decide whether or not it is advantageous to use the refinement.

	Optional				
	9	12	15	18	21
EC-BR	0.3	0.1	0.2	0.2	0.3
EC-BR Ref	51.2	80.4	81.3	82.8	84.3
Fekete04	83.2	80.3	69.0	63.5	64.6
Funke05	61.5	59.5	55.4	52.5	50.6
Funke06	80.6	70.7	71.9	72.5	73.2
MDS-BR	68.0	79.8	79.8	79.8	80.0

Table 6.2: Percentage of optional nodes that are classified as interior nodes for average node degrees between 9 and 21.

6.5.4 Random Placement vs. Perturbed Grid Placement

For most applications of wireless sensor networks, it is likely that the nodes are distributed in some systematic way. Accordingly, perturbed grid placement is frequently used in work on automatic boundary recognition. As perturbed grid placement ensures a low variance in node degree over the entire network, one can expect that it especially benefits statistical approaches for boundary recognition.

However, there are also scenarios where it is more appropriate to assume a truly random placement. Random placement results in a more chaotic distribution of nodes and in the emergence of many small holes, as can be seen in Figure 6.15. In this section,

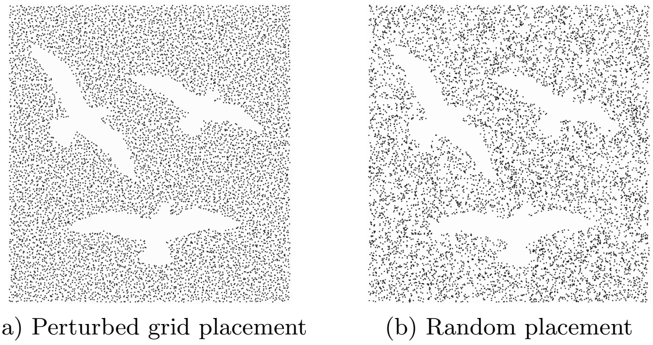

(a) Perturbed grid placement (b) Random placement

Figure 6.15: Comparison of different placement strategies.

we examine the influence of the placement strategy on the classification quality.

Table 6.3 compares misclassification rates for both placement strategies on networks with an average node degree $d_{avg} = 15$. Apparently, when random placement is used, EC-BR without refinement classifies many interior nodes as boundary nodes. The reason is that in this case many small holes emerge, which fall below the threshold of the hole definition but which are still large enough to be detected by EC-BR. By using the refinement, EC-BR can fix most misclassifications and achieves the best average misclassification rate for interior nodes. Despite the random placement, only 4.1% of the interior nodes are classified as boundary nodes. At the same time, EC-BR with refinement produces also the least percentage of misclassifications of mandatory boundary nodes. The performance of the other considered algorithms decreases dramatically compared to perturbed grid placement. Only MDS-BR has a similarly low misclassification rate for mandatory nodes when random placement is used.

	Mandatory		**Interior**	
	random placement	perturbed grid	random placement	perturbed grid
EC-BR	**2.0**	**0.0**	48.7	3.8
EC-BR Ref	4.0	0.6	**4.1**	**0.0**
Fekete04	26.2	6.7	13.0	7.2
Funke05	15.5	5.7	16.1	2.0
Funke06	45.8	16.6	7.9	1.4
MDS-BR	5.2	3.5	12.2	0.3

Table 6.3: Misclassifications for random placement and perturbed grid placement.

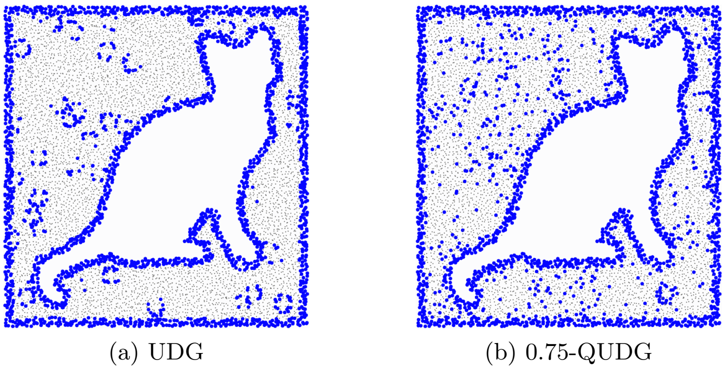
We also analyze the influence of the network density when using random node placement. Table 6.4 shows the classification results for mandatory and interior nodes when the average node degree is varied. We see that all algorithms perform better with increasing network density. Interestingly, Funke06 seems to miss much more mandatory nodes than the other algorithms when random placement is used. For sparse networks, all algorithms show an increased misclassification of interior nodes. This is partly caused by the recognition of very small holes, which are detected by the algorithms but fall below the threshold of the hole definition. If one is interested in such holes, EC-BR without refinement can be used, otherwise EC-BR with refinement seems to work very well.

	Mandatory			**Interior**		
	15	20	25	15	20	25
EC-BR	**2.0**	**1.6**	**0.5**	48.7	25.9	11.3
EC-BR Ref	4.0	3.1	1.9	**4.1**	**0.5**	**0.1**
Fekete04	26.2	13.7	7.7	13.0	13.9	12.3
Funke05	15.5	9.4	6.7	16.1	8.1	3.6
Funke06	45.8	29.1	26.4	7.9	6.1	2.8
MDS-BR	3.8	5.2	5.8	13.4	5.2	1.7

Table 6.4: Misclassifications in dependence of average node degree for random placement.

6.5.5 Beyond Unit Disk Graphs

Unit disk graphs are frequently used for theoretical analyses and in simulations. They are motivated by the fact that under good-natured conditions every sender has a transmission range which is roughly fixed. However, under realistic assumptions the transmission range also depends on environmental conditions and obstacles, as well as on unpredictable effects such as interference and signal reflections. In this section, we evaluate the algorithms under more realistic conditions. Uncertainties are taken into account by the use of the quasi unit disk graph model, which incorporates the observation that short-range transmissions are usually successful while long-range transmissions have some random behavior. A visual comparison between a classification result of EC-BR on a network which is based on the UDG model and a network that is based on the 0.75-QUDG model is presented in Figure 6.16. The most apparent difference is that there are many isolated misclassifications of interior nodes in the QUDG classification Fortunately, it is easily possible to eliminate these artifacts using the refinement of EC-BR.


(a) UDG

(b) 0.75-QUDG

Figure 6.16: Comparison of EC-BR classification results before refinement.

Another difference is that in networks based on the QUDG model it is not necessarily true that a mandatory boundary node is completely surrounded by EC-BR boundary candidates. Accordingly, for the EC-BR refinement a lower threshold than 100% has to be used. For our experiments, we use $\gamma = 70\%$. This threshold works well for both UDGs and QUDGs.

Table 6.5 compares the algorithms for average node degrees 12 and 15 in simulations on 0.75-QUDG networks. EC-BR without refinement classifies for networks with low node density many interior nodes as boundary nodes. The reason is that many nodes that are less than one maximum communication distance away still cannot communicate with each other. Thus it often is not possible to find an enclosing circle. However, as the misclassified

	Mandatory		**Interior**	
	12	15	12	15
EC-BR	**0.0**	**0.0**	28.5	7.7
EC-BR Ref	**0.0**	**0.0**	**4.9**	**0.3**
Fekete04	16.9	6.9	8.8	8.9
Funke05	9.0	7.4	12.9	5.2
Funke06	15.6	15.4	12.4	3.7
MDS-BR	8.3	11.2	8.3	1.6

Table 6.5: Misclassifications for the 0.75-QUDG model and different average node degrees.

nodes are usually isolated (cf. Figure 6.16), the problem can easily be avoided by using the refinement of EC-BR. With refinement, EC-BR clearly outperforms the other approaches in our simulations.

In Table 6.6, we go a step further and compare 0.25-QUDGs with 0.75-QUDGs. This means that there is a very high level of uncertainty. As expected, all algorithms produce more misclassifications. Again, EC-BR with refinement produces the best separation of interior nodes and mandatory boundary nodes.

	Mandatory		**Interior**	
	0.25-QUDG	0.75-QUDG	0.25-QUDG	0.75-QUDG
EC-BR	**3.0**	**0.0**	41.5	28.5
EC-BR Ref.	12.7	**0.0**	**1.7**	**4.9**
Fekete04	14.6	16.9	13.6	8.8
Funke05	11.5	9.0	17.8	12.9
Funke06	24.2	15.6	2.8	12.4
MDS-BR	27.7	8.3	11.8	8.3

Table 6.6: Misclassifications for 0.25- and 0.75-QUDG models with average node degree 12.

6.5.6 Parameter Selection

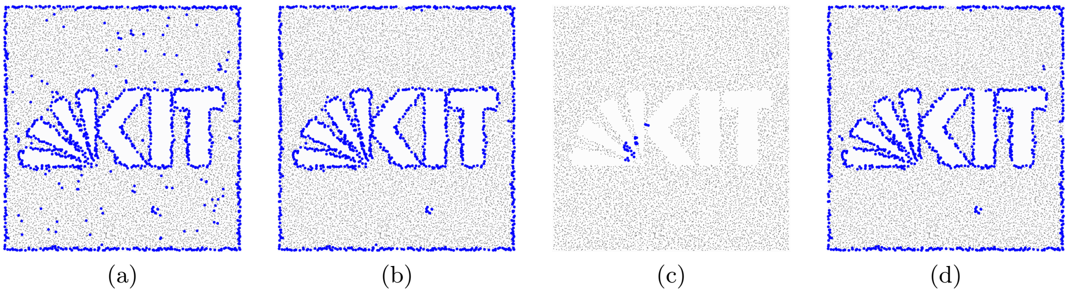
Most approaches to the boundary recognition problem involve several parameters which have to be tuned to a specific scenario in order to produce nice classifications. A very nice feature of EC-BR in this context is that a single fixed parameter set already works well for all scenarios. The circle length of 6 as a threshold to distinguish inner nodes from boundary nodes is mostly independent of node degree, kind of placement, and communication model. And for the refinement, checking whether 70% of the neighbors are marked as being on the boundary works well in all situations. If desired, for the special case of unit disk graphs a threshold of $\gamma = 100\%$ can be used to obtain a slightly thinner boundary.

6.5.7 Refinement

For EC-BR and MDS-BR, we used two different refinement heuristics to eliminate false positives. One might ask why two different refinement routines are necessary, or how well each of them works with the other algorithm. In short, each refinement step is adjusted to the corresponding algorithm and performs poorly with the other one. To illustrate this behavior, Figures 6.17 and 6.18 show the results of EC-BR and MDS-BR with both refinement heuristics.

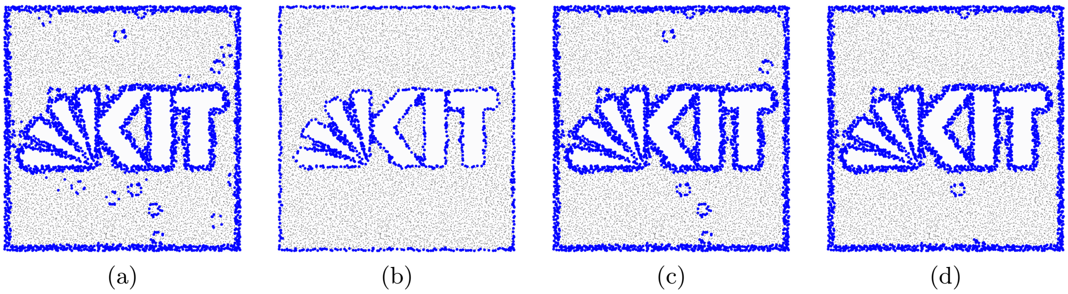
Using the refinement step of EC-BR in connection with MDS-BR, almost no node is classified as boundary node. This occurs as MDS-BR already yields a thin outline of the boundary, but the refinement of EC-BR reclassifies all boundary nodes that are not surrounded by other boundary node candidates. To obtain better results, threshold γ can be adjusted. In this scenario, the best results are achieved if a node has to be surrounded by only $\gamma = 20\%$ of boundary nodes. This considerably improves the classification, but there are still sections of the boundary missing while “noise” already starts to emerge.

If the refinement step of MDS-BR is applied to the classification results of EC-BR, only few boundary nodes are switched to interior nodes. As EC-BR yields a broad halo around each boundary node, the refinement of MDS-BR almost always finds a sufficiently long path within this halo. Even using more aggressive parameter values for the refinement does not improve the results by much. Some additional small structures are removed, but



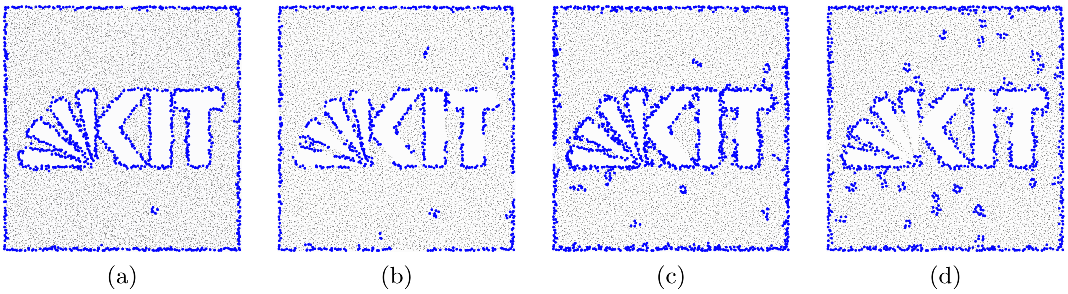
(a) (b) (c) (d)

Figure 6.17: Impact of using different refinement heuristics on the classification results of MDS-BR. (a) No Refinement. (b) MDS-BR Refinement. (c) EC-BR Refinement ($\gamma = 100\%$). (d) EC-BR Refinement ($\gamma = 20\%$).



(a) (b) (c) (d)

Figure 6.18: Impact of using different refinement heuristics on the classification results of EC-BR. (a) No Refinement. (b) EC-BR Refinement. (c) MDS-BR Refinement ($r_{min} = 3$). (d) MDS-BR Refinement ($r_{min} = 10$).



(a) (b) (c) (d)

Figure 6.19: Impact of MDS-BR refinement on different algorithms. (a) MDS-BR. (b) Fekete04. (c) Funke05. (d) Funke06.

the boundary remains a broad halo.

One might wonder whether the refinements of EC-BR and MDS-BR could be used to improve the other examined algorithms. The only approach that could possibly benefit from the refinement of EC-BR is the algorithm of Saukh *et al.*. However, this algorithms is already very computational expensive and it produces very little noise, so the main advantage would be that the computed boundary is thinner. Similar to MDS-BR, the other algorithms all produce boundaries which are already too thin for the refinement of EC-BR.

We now analyze the performance when MDS-BR refinement is applied to other algorithms. As seen in Figure 6.19, the refinement removes some of the initial noise, but the results remain worse than MDS-BR. In case of Kroeller04, the noise is almost completely gone, but so are large parts of the boundary. The results of Funke05 retain some of the original noise but still show a broad boundary structure at the border. Similar to EC-BR with MDS-BR refinement, Funke06 retains boundaries around micro-holes, but boundaries around small structures are completely lost.

Overall, it is evident that the refinements of EC-BR and MDS-BR are tailored for their respective base algorithms. In conjunction with them they yield significant improvements, but if applied to the results of a different algorithm, the final classification turns out to be quite poor. This also refutes the assumption that the presented algorithms only perform so well due to the refinement step, and others could close the gap easily by using the same refinement.

6.6 Summary

In this chapter, we studied the Enclosing Circle Boundary Recognition (EC-BR) algorithm, a *distributed* algorithm for *location-free* boundary recognition in wireless sensor networks. With EC-BR, every node only needs *connectivity information* of its local 2-hop neighborhood to decide with high probability whether it is near a hole or boundary. Its low communication overhead makes EC-BR an excellent choice for boundary recognition in large-scale sensor networks. Additionally, EC-BR should be well suited for scenarios which include mobility or dynamic changes of the network topology.

We showed in extensive simulations that EC-BR is very robust to different network densities, communication models, and node distributions. Its low computational complexity and communication overhead make it very attractive for real-world implementation even on weak sensor nodes.

Chapter 7

RSS-based Localization: Preliminaries

Localization based on received signal strengths is very popular in the context of wireless sensor networks. On the hardware side, all that is necessary are wireless receivers that provide some kind of received signal strength indication (RSSI). This kind of hardware cannot only be produced very small and cheap, but it is also already available on most devices that communicate wirelessly. Thus, RSS-based localization can be performed without adding further hardware, which would make the sensor nodes larger or more expensive. Unfortunately, RSS-based localization also has some shortcomings. Probably the biggest issue is that received signal strengths are influenced by a multitude of factors, many of which are rather unpredictable. In particular, the dependence between distance and received signal strength is strongly influenced by the used hardware, the frequency of the radio signal and the distribution and constitution of walls and other obstacles. As a result, it is not easily possible to directly compare localization results that were produced with different kinds of hardware or in different settings.

In this chapter, we describe the experiments and simulations that we use in this work to compare different localization approaches. We start with a short description of the experiments. Subsequently, we analyze the collected experimental data to study the dependence between distances and received signal strengths in the considered setting, focusing also on the attenuation effects at walls. The observations of this analysis are finally used to design a meaningful simulation setup.

7.1 Experimental Setup

7.1.1 Used Hardware

For the real-world experiments, a network of 61 self-built sensor nodes was used. The sensor nodes have been built in another project of the DFG Research Training Group 1194 on wireless sensor networks by Johannes Schmid. Each sensor node consist of a Texas Instruments MSP430 MCU and a 2.4 GHz IEEE 802.15.4 compliant CC2520 radio chip, with a receiver sensitivity of $-98\,\mathrm{dBm}$. The communication in the network is based on the ZigBee standard [Gis08]. SD-card interfaces at some of the sensor nodes make it possible to record received signal strengths to an SD card to enable offline processing and evaluation. Additionally, a Xsens MTi-G Inertial Measurement Unit (IMU) was used to collect acceleration data during the experiments. For further information on the used sensor network and the XSens MTi-G IMU, we refer to [SVG$^+$10].

7.1.2 Scenarios and Data Collection

For the sensor network experiments, 60 sensor nodes were deployed more or less randomly in two floors at different buildings of the *Karlsruhe Institute of Technology.* The first set of experiments was done at the *Institute for Theoretical Informatics (ITI)*. After the deployment of the sensor nodes, the corresponding node positions have been carefully marked in a map. Figure 7.1 shows a map of the ITI floor, with the sensor node positions marked by blue points. Next, several walking trajectories have been planned, and important waypoints have been marked and numbered on a map. One example for such a walking trajectory is shown by the red line in Figure 7.1.

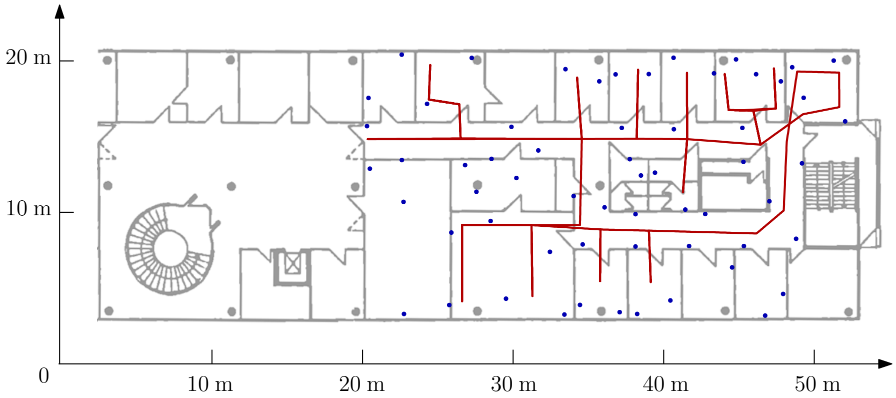


Figure 7.1: Floor plan of the ITI. The blue dots correspond to sensor node positions, the red line shows one sample walking trajectory.

The experimental data were then collected as follows: A person with an additional sensor node attached in front of the body followed each of the pre-planned trajectories from waypoint to waypoint. The other sensor nodes were programmed to broadcast their IDs and positions with a frequency of 4 Hz. While the person was walking, the on-body sensor node was used to save for every time step of 0.25 seconds the RSS values of 24 received messages, together with the node ID of the sender and the time the message was received. Every time when a waypoint was passed, the elapsed time since the start of the experiment was recorded. In the end, the positions of the waypoints together with the recorded passing times have been used to reconstruct the walked trajectory. To estimate the trajectory and arrival times for positions between the waypoints, linear interpolation was used. The final result of this procedure was a set of positions and corresponding passing times for the trajectory, and a set of RSS values with reception times and node IDs for the received messages. Additionally, an Xsens MTi-G Inertial Measurement Unit was used to detect steps during the experiments.

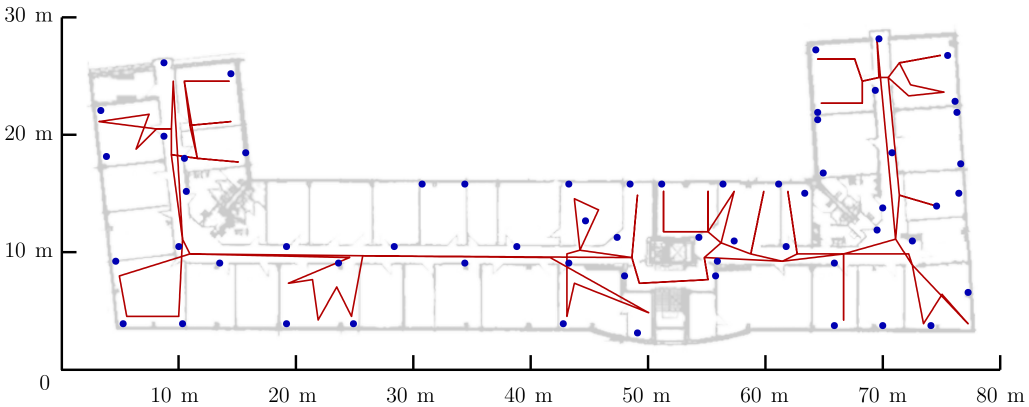


Figure 7.2: Floor plan of the ITIV. The blue dots correspond to sensor node positions, the red line shows one sample walking trajectory.

Using the same approach, an additional second data set has been taken at the *Institut für Technik der Informationsverarbeitung (ITIV)*. This time, up to 46 RSS measurements were stored per time step (0.25 seconds). The corresponding map of the ITIV floor, together with the positions of the sensor nodes and one sample trajectory, are depicted in Figure 7.2.

In this work, we mainly use the data from four experiments at the ITI and five experiments at the ITIV. Some statistical data for these experiments are given in Table 7.1. In the row *RSS Measurements*, only recorded RSS values are counted. The numbers are given in multiples of 1.000, e.g., in the first experiment at the ITI about 38.000 RSS values have been recorded.

Scenario	ITI				ITIV				
	1	2	3	4	1	2	3	4	5
Beacon Nodes	60	60	60	60	60	60	60	60	60
Time Step Duration (s)	1/4	1/4	1/4	1/4	1/4	1/4	1/4	1/4	1/4
RSS Values / Time Step	24	24	24	24	46	46	46	46	46
Duration (s)	408	334	104	325	289	681	658	910	206
Walked Distance (m)	198	214	87	261	257	460	461	615	218
Avg. Velocity (km/h)	1.8	2.3	3.0	2.9	3.2	2.4	2.5	2.4	3.8
RSS Measurements	38k	31k	9k	30k	34k	75k	70k	99k	24k
Detected Steps	395	424	155	497	443	879	890	1157	346

Table 7.1: Properties of the experimental runs.

The experiments at the ITI building have been performed in cooperation with Johannes Schmid, the data of the additional experiments at the ITIV building have been kindly provided by Johannes Schmid.

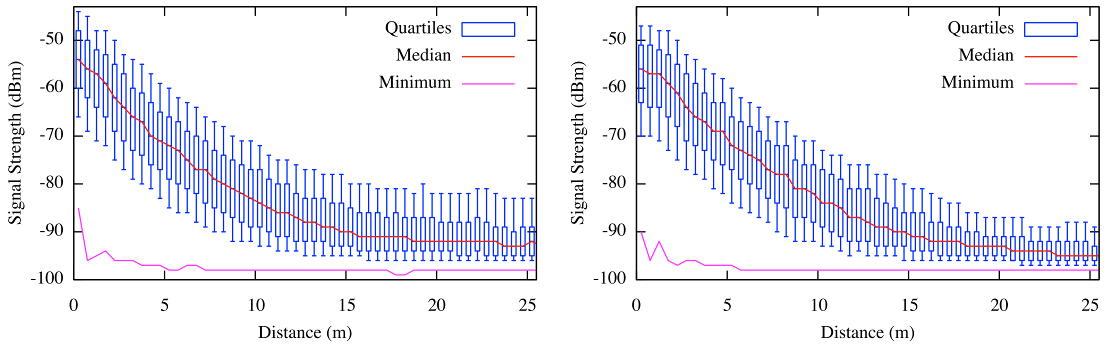
7.2 Data Analysis

A basic understanding of the characteristics of received signal strengths is crucial for the design of efficient algorithms and meaningful simulations. For this reason, we present in this section a short study of received signal strengths for the considered sensor network experiments. In particular, we examine the influence of sender-receiver-distances and of walls, which obstruct the line of sight between sender and receiver, on received signal strengths. For references to related work in the context of the study of signal propagation in buildings, we refer to Section 2.4.1.

7.2.1 Distance Dependence of RSS

Figure 7.3 shows a box-and-whisker plot for the dependence of received signal strengths on the distances between senders and receivers for experiments at the ITI and the ITIV. The boxes show the upper and lower quartiles, i.e., the signal strengths such that 25 % of the measurements for the considered distances have higher or lower RSS values, respectively. The whiskers show the 10^{th} and 90^{th} percentiles.

The decay of received signal strengths, at least on average, is evident in both plots. However, one can also see that the uncertainties are very high. For example, the upper and lower quartiles are separated on average by about 12 dBm. This corresponds to a difference of at least a factor of about $10^{1.2}$ (≈ 15) in signal strengths between the 25 %



(a) Experiments at the ITI

(b) Experiments at the ITIV

Figure 7.3: Distance dependence of received signal strengths.

measurements with highest RSS and the 25 % measurements with lowest RSS, for the same distance. Additionally, one can see that even for short distances some packets had a received signal strength close to the receiver sensitivity of −98 dBm.

For distances exceeding about 12 m, it appears that the signal strength decays much slower with distance. However, at least to some degree this is due to the limited receiver sensitivity: All packets that arrive with signal strength lower than the receiver sensitivity of −98 dBm are simply not detected, meaning that for larger distances only those packets that arrive the receiver with unusually strong power are included in the averaging. Overall, the plots for the ITI and the ITIV look very similar, suggesting that both buildings share similar characteristics concerning signal propagation.

Figure 7.4 shows the distribution of the measured signal strengths for several distance bins, corresponding to a 1 m range of distances each. The curves with points correspond to the measurements at the ITI, the plain curves correspond to the measurements at the ITIV. Astoninglishly, both kinds of curves are extremely close, which is very unexpected for RSS measurements in different buildings. This similarity is probably caused by the enormous amount of measurements, hundreds of thousands of RSS-distance-pairs were used, and by similar characteristics of both buildings with respect to the distribution and constitution of walls.

As expected, the distributions of RSS measurments for a given distance range follow a Gaussian distribution. For small distances up to about 8 m, the standard deviation is about 9 dBm. For larger distances, the standard deviation decreases due to the limited

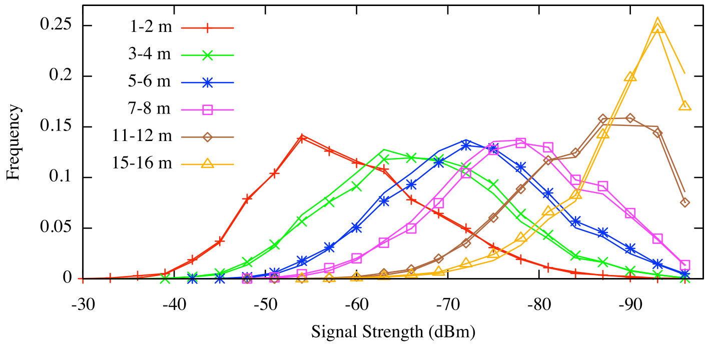


Figure 7.4: Signal strength distribution for several distance ranges. For each curve, only those measurements with the given sender-receiver distance were used. The curves with points are based on the experiments at the ITI, the plain curves are based on the experiments at the ITIV.

sensitivity of the receiver, which is unable to decode messages whose signal strength is below −98 dBm. There are some slight deviations from the normal distribution, which are evident for the experiments in both buildings, e.g., a slightly higher than expected frequency for the RSS range from −55 dBm to −53 dBm or a slightly lower than expected frequency for the range from −85 dBm to −83 dBm. These deviations are probably caused by characteristics of the radio receivers.

In Figure 7.4, one can also clearly see how the limited receiver sensitivity affects the otherwise normal distribution for measurements over large distances. As the receiver does not detect packets with signal strength below −98 dBm, even for distances of only about 15 m the distribution significantly differs from a normal distribution.

We also observe in Figure 7.4 that even the curves of distance bins that correspond to very different distances overlap significantly. This indicates that one has to expect very high uncertainties when RSS measurements are used to infer distances between pairs of nodes.

7.2.2 Influence of Walls

The high fluctuation of signal strengths for a given distance is caused by a multitude of effects. In this section, we take a closer look at the role that is played by walls that obstruct the line of sight between senders and receivers.

Figure 7.5 depicts the average number of walls that cross the line of sight between senders and receivers in dependence of the sender-receiver-distance. Interestingly, although the two buildings have very different shapes (cf. Figures 7.1 and 7.2), up to a distance of about 25 m the average number of walls per distance is almost equal in both buildings. Assuming a similar constitution of the walls, this observation might explain the very similar RSS characteristics of both buildings in Figure 7.4.

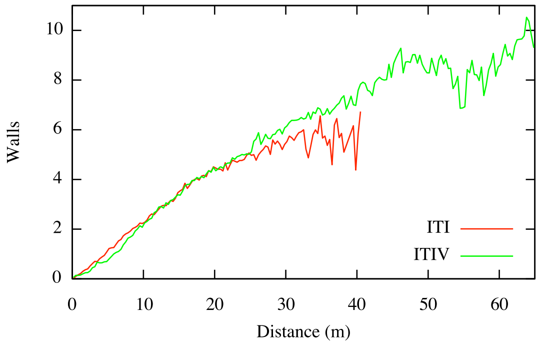


Figure 7.5: Average number of penetrated walls in dependence of distance.

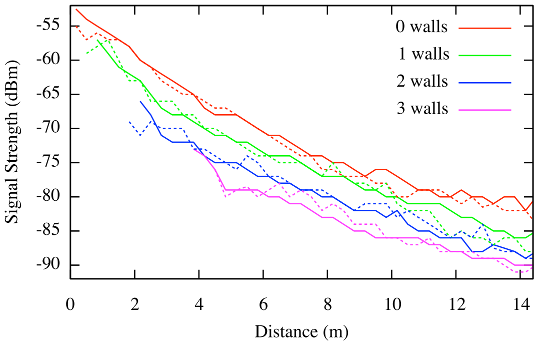


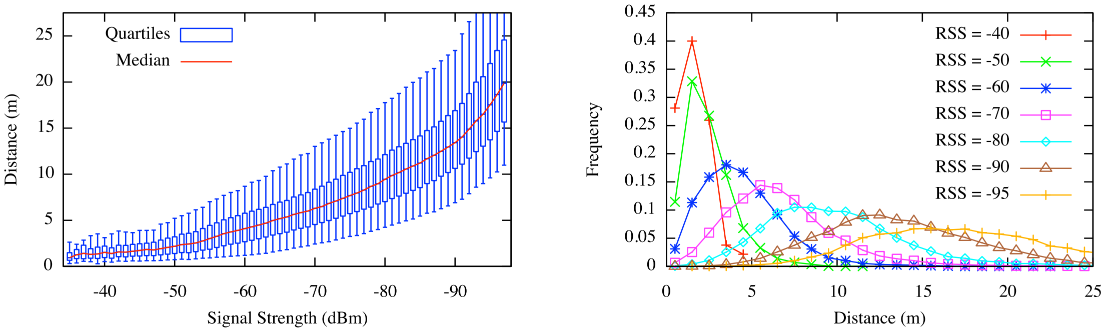
Figure 7.6: RSS in dependence of distance and penetrated walls for ITI (solid lines) and ITIV (dotted lines).

Next, we take a look at the signal attenuation caused by walls. For this, we partition all RSS measurements with respect to the number of walls that cross the line of sight between the position of the sender and the measurement position. Figure 7.6 shows the distance dependence of the RSS for the measurements where the line of sight between sender and receiver crossed between 0 and 3 walls. Again, the similarity between the measurements in the ITI building and those in the ITIV building is apparent. At least for the first few meters, one can roughly say that the average signal strength falls of with about 3 dBm per meter and additional 3.5 dBm per penetrated wall.

In some of the figures in this chapter, we focus only on the signal strength behavior over short distances up to about 15 meters. The reason is that we will base our localization approaches mainly on RSS measurements with high signal strengths, which usually do not occur for transmissions over larger distances. As soon as larger distances are considered, additional effects occur, for example due to the limited sensitivity of the receivers or border effects caused by the limited dimensions of the test areas.

7.2.3 Distance Distribution for Given RSS

In RSS-based distance estimation, a given RSS value is translated into some distance estimate. Figure 7.7 shows distance distributions in dependence of the received signal strength. Apparently, measurements with high RSS offer the most information. The lower the signal strength, the broader the distribution of sender-receiver-distances that resulted in the corresponding RSS measurement. For example, if a packet is received with an RSS of $-50\,\text{dBm}$, one can assume with high certainty that the sender is less than 6 meters away. But for an RSS of $-80\,\text{dBm}$, the distribution is already very spread out.



(a) The boxes show upper and lower quartiles, the whiskers show the 5th and 95th percentiles.

(b) Distance distribution for certain RSS values. Signal strengths are stated in dBm.

Figure 7.7: Distance distributions for given RSS.

The distributions in Figure 7.7b differ significantly from normal distributions. This has two reasons: First, there are more measurements over larger distances than over short distances, as the enclosed area grows quadratically with increasing radius. Second, there are also border effects, this time for strong signals. As there are no negative distances, distributions whose peaks are at small distances are skewed to the right.

7.3 Inferring Distances from RSS

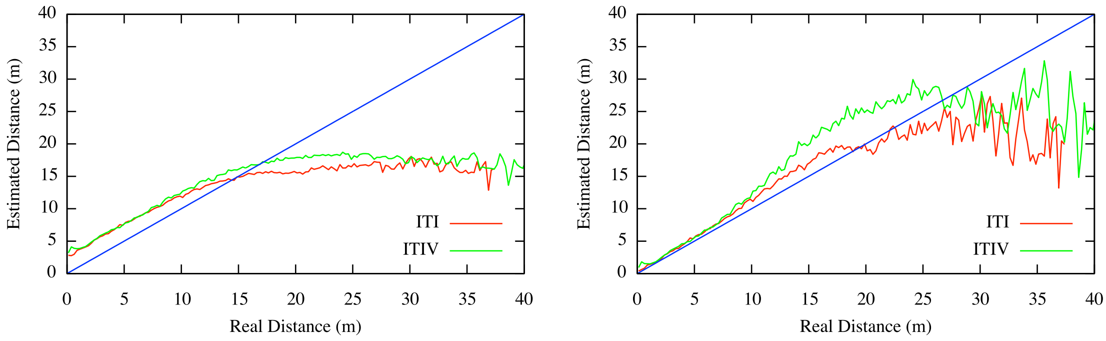
There are several possibilities to infer distances from RSS values. For example, given a signal strength measurement, one can use distributions like the ones depicted in Figure 7.7 to assign each distance d some probability that the sender is located in distance d. Such an approach is used for example in the probabilistic algorithm for indirect node localization presented in [SR04b].

Non-probabilistic algorithms usually require that each RSS value is mapped to a single distance. The standard approach in this context is based on the log-distance path loss model [SBN$^+$11]. First, signal strength measurements are taken for known distances. These RSS-distance pairs are then used to compute a good fit for the parameters of the log-distance model, i.e., the path loss exponent and the path loss at some reference distance d_0. The mapping from RSS values to distances is then computed by inverting the function of the log-distance model. There also exist possibilities to determine parameters of the model based only on RSS measurements, thus avoiding the difficult distance measurements for the initializing phase [MAF07].

Especially if only few measurements are available, the use of the log-distance model helps to avoid an overfitting on the training data, as unrealistic variations of the RSS-distance-relation are smoothed. However, the use of the log-distance model also has some shortcomings. Even though the model usually allows a good fit to the data, it is a rather strong assumption that the signal decay in buildings can be described fully using only two parameters. When the parameters of the log-distance model are fitted to the measurements, the resulting curve is usually optimized for medium to long distances, as there are more measurements over larger distances than over short distances. However, for localization, especially the measurements over short distances with high signal strength are interesting, as they offer the most information about the sender-receiver distance. Additionally, when optimizing the RSS-distance relation for a certain kind of wireless network and building, it even might sometimes be preferable to represent the special characteristics of the considered setting in the mapping of signal strengths to distances.

As we have a sufficiently large amount of RSS-distance pairs for the calibration, we use in this work a direct mapping from RSS values to estimated distances. For the calibration procedure, we use an additional training data set consisting of about $300,000$ RSS-distance pairs. This data set was determined with the same experimental setup as outlined in Section 7.1, but in additional runs.

Given a large amount of RSS-distance pairs and a single RSS value r for which the distance has to be estimated, it is tempting to take all sender-receiver distances of measurements that resulted in RSS r and to choose the mean of this set (cf. Figure 7.7a). This seems natural, as such an approach promises to minimize the average distance estimation error. Figure 7.8a shows how the distance estimation behaves on average for this assignment strategy. Apparently, for measurements over small distances, the distance is systematically overestimated. This overestimation can be explained by the asymmetric distribution of distances that result in a certain RSS, which was already discussed in connection with Figure 7.7. In experiments with this kind of mapping from RSS to distances, it turned out that the systematic overestimation of short distances had a negative influence on the localization accuracy.



(a) Strategy 1

(b) Strategy 2

Figure 7.8: Average distance estimation vs. real distance for two different mappings from signal strengths to distances.

To achieve that short distances are about equally often overestimated and underestimated, the function that represents the mean signal strength in dependence of distance (cf. Figure 7.3) has to be inverted. This is the standard approach that is usually used in connection with the log-distance model, and we also use this approach, just without the additional fit to the log-distance model. Figure 7.8b shows the resulting relation between estimated distances and real distances for the measurements in our experiments. Apparently, for distances up to about 8 m, the estimated distances fit the real distances on average very well.

7.4 Simulations

Realistic modeling of signal decay is a tough topic. This already holds true for outdoor scenarios, and in indoor environments things become even worse due to walls, obstacles, and a variety of other effects. Thus, simulation results involving simulated signal strengths always have to be handled with some care. However, simulations also offer some benefits in comparison with real world experiments: They make it possible to specify all assumptions and circumstances, making results better reproducible than it is the case for real-world experiments. Additionally, simulations make it easy to evaluate algorithms on a large data basis, whereas the execution of real-world experiments is very time consuming and error-prone.

One important aspect that motivates the use of simulations in the considered localization scenario is that simulations eliminate the uncertainties in connection with the exact positions of static and mobile nodes. Although we took immense care to protocol the positions of the sensor nodes as good as possible, there is still some uncertainty concerning the exact positions. The same holds even more true for the walked trajectory, which in the considered experiments is presumably only known with an accuracy of maybe one meter.

For this reasons, we supplement the real-world experiments with additional simulations. Of course, when considering localization based on signal strengths, it is important to model at least the dominant aspects of signal propagation reasonably well. For localization within buildings, this includes the attenuation effects that are caused by walls.

We put some effort into the design of reasonably realistic simulations. Our goals were to imitate the average signal decay over distance, the average attenuation at walls, the variance of signal strengths for fixed distances, and the acceptance rates of packets in dependence of distance and number of penetrated walls. In this section, we describe the design of our simulations and compare some simulation results with real-world measurements. All simulations are based on the floor plan of the ITI.

7.4.1 Modeling of Path Loss

Simulation of indoor signal propagation is frequently based on the log-distance path loss model (cf. Section 2.4.1), i.e., it is assumed that the signal decay can be represented by a simple power law. The popularity of the log-distance model is caused by the fact that the function which represents the average signal strength for a given distance (cf. Figure 7.3) can usually be fitted to some power law. The path loss exponent of the log-distance model, which depends on the characteristics of the building, then models how fast the signal decays.

For the considered indoor localization problem, this modeling entails two major drawbacks: First, it does not represent the systematic signal attenuation at walls. This means that, in the simulation, a sender-receiver-pair that is separated by several walls on average measures the same signal strength as another sender-receiver pair that is separated by the same distance, but without any walls obstructing the line of sight. Second, when fitting the log-distance model to the experimental data, one usually does not take into account that only a fraction of the packets that are sent over a large distance reach their destination. Thus, even for such distances for which the average signal strength at the receiver is below the receiver sensitivity threshold, the log-distance model assumes an average signal strength above the receiver sensitivity. As a result, if one does not take care of this effect, even for large distances between sender and receiver most packets reach the receiver with a signal strength above the sensitivity threshold. This can have significant impact on localization results.

In our simulations, we use an approach similar to the one used in the *free space + linear path attenuation model* that was proposed in [DBKR90]. In this model, to mimic the signal decay in buildings, a free space path loss is supplemented with an additional loss factor that increases exponentially with distance. Like [DBKR90], we also found for our experimental data that such an exponential component helps to fit the data better. In [DBKR90], the exponential factor already subsumed the effects of walls, and an additional floor attenuation factor (cf. Section 2.4.1) was used to represent the average attenuation that is caused by the penetration of floors. In this work, we only consider a single floor, so the floor attenuation factor plays no role. Instead, we consider the penetration of walls separately. To compute the received signal strength $P_{Rx}(d)$ for a single packet that is sent over distance d, we use the following formula

$$P_{Rx}(d) = -53\,\mathrm{dBm} - 20\log_{10}\frac{d}{d_0}\,\mathrm{dBm} - \alpha d - \sum_{i=1}^{w} X_i - X_\sigma\,, \tag{7.1}$$

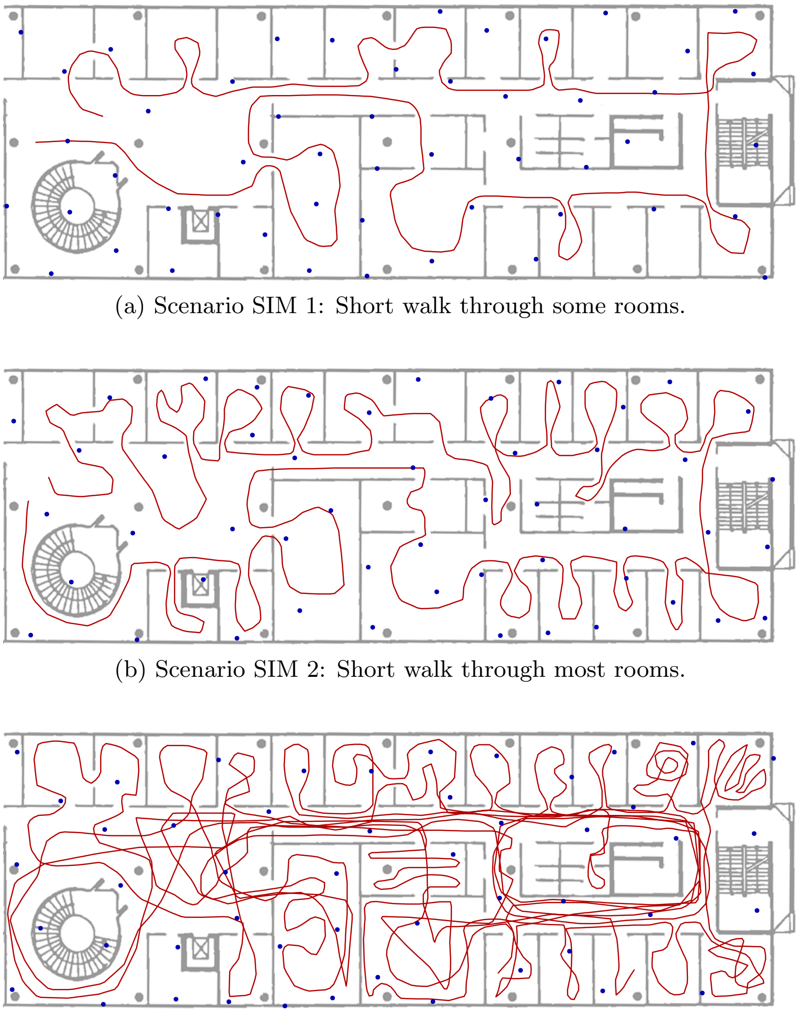
where $-20\log_{10}\frac{d}{d_0}$ models the free space path loss, $d_0 = 1\,\mathrm{m}$ is the corresponding reference distance, $\alpha = 0.5\,\mathrm{dBm/m}$ is an attenuation constant that controls the exponential path loss caused by obstacles such as furniture and office equipment, w is the number of walls that obstruct the line of sight between sender and receiver, $X_i \sim \mathcal{N}(-3.5, 1^2)$ [dBm] is a normally distributed random variable that models the attenuation at the i-th wall crossing the line of sight, and $X_\sigma \sim \mathcal{N}(0, 8^2)$ [dBm] is a normally distributed random variable that subsumes all other random effects. The parameters in Equation 7.1 have been chosen to fit the experimental data. In Section 7.4.3, we present a comparison of the resulting signal decay characteristics with the data from the experiments. Additional details about the modeling of signal propagation and path loss are given in Section 2.4.1.

7.4.2 Simulation Scenarios

Node Placement. For the simulations, we use random node placements. To achieve that the node distribution is somewhat balanced, we use the following approach: First, 200 nodes are distributed randomly within the considered area. The intended number of nodes is then selected by first choosing the two nodes with maximum distance from each other and subsequently selecting additional nodes one by one. In each step, the node which maximizes the minimum distance to any already chosen node is selected. This is repeated until the intended number of nodes is reached.

If not stated otherwise, we use 50 sensor nodes in the simulations. If several algorithms are compared, all algorithms get the same node placements as input.

Walking Trajectories. We consider 3 different walking scenarios, which mainly differ in the length of the walked trajectory and the areas of the floor that are covered. The consideration of different path lengths will allow us later in Chapter 10, where the indirect localization of a sensor network by a mobile sensor node is studied, how the localization accuracy depends on the walked trajectory. The first scenario (*SIM 1*, Figure 7.9a) models



(a) Scenario SIM 1: Short walk through some rooms.

(b) Scenario SIM 2: Short walk through most rooms.

(c) Scenario SIM 3: Long walk covering the whole floor.

Figure 7.9: Overview of the simulations scenarios. In each figure, the red line shows the corresponding walking trajectory and the blue points show one sample distribution of 50 wireless sensor nodes.

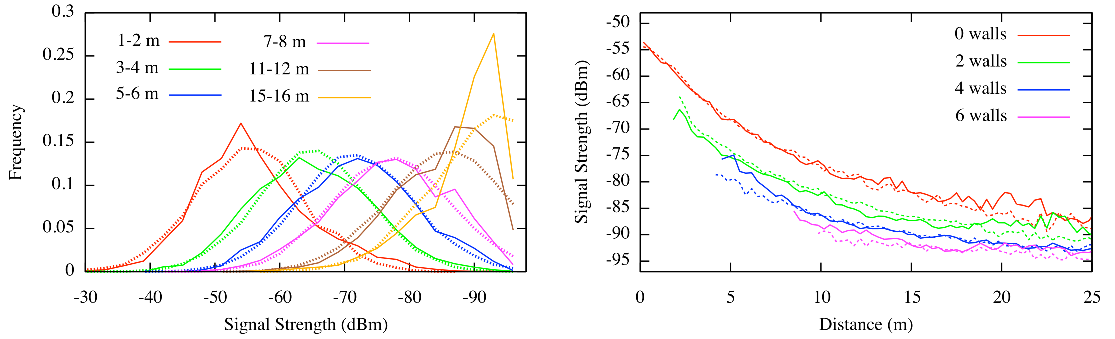
a brief walk through parts of the floor. In the second scenario (*SIM 2*, Figure 7.9b), most of the rooms are visited once. The third scenario (*SIM 3*, Figure 7.9c) represents a long walk that covers most of the area of the floor. In all simulations, we assume that the mobile device moves along the trajectories with constant speed of 4 km/h.

Communication. Communication in the simulations is modeled in analogy to the real-world experiments. Every sensor node broadcasts its positions four times per second. The received signal strengths are modeled according to Equation 7.1. If the RSS of a message falls below the receiver sensitivity of -98 dBm, it is assumed that the receiver has no knowledge of the message. If not stated otherwise, we assume that the receiver saves the RSS values of all received packets, i.e., there is no limitation to 24 RSS values per time step of 0.25 seconds like in the experiments at the ITI.

7.4.3 Comparison between Simulation and Experiment

In this section, we present a brief comparison between the simulations and the real-world measurements. For the simulations, the same trajectories that have been used in the real-world experiments are used. To make the simulations and the experiments comparable, we assume in this section that like in the experiments at the ITI, for each time step only RSS measurements for up to 24 different sensor nodes are saved.

Figure 7.10a shows signal strength distributions for certain distance intervals. For each distribution, only those measurements are used for which the sender-receiver distance falls into the considered distance interval. The comparison between simulation and experiment reveals that at least up to a distance of 12 m, the statistical distributions of simulation and experiment are very similar.



(a) Signal strength distributions

(b) Signal decay in dependence of penetrated walls

Figure 7.10: Comparison of received signal strengths between real-world measurements (ITI, solid lines) and simulations (dashed lines).

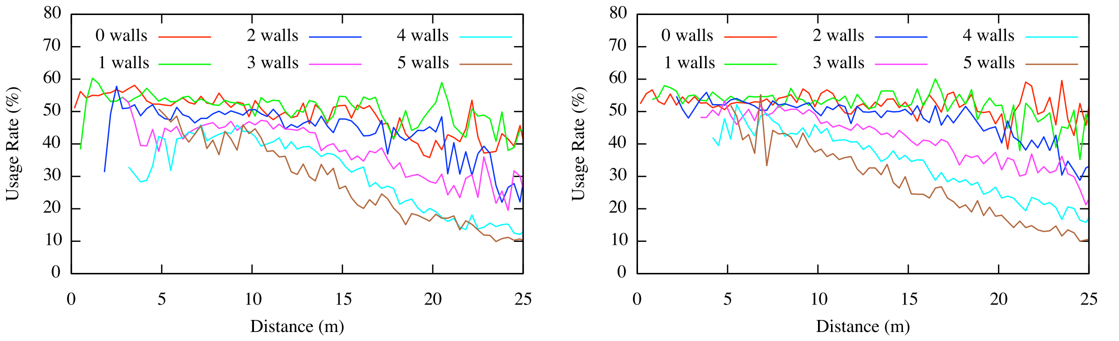
Figure 7.10b shows the influence, which the number of penetrated walls has on the received signal strengths. The systematic signal attenuation due to walls is apparent for both experiments and simulations.

Finally, in Figure 7.11 we study how the probability that a certain message belongs to the 24 measurements that are stored in the database in each time step depends on the distance between the sender and the receiver and the number of penetrated walls.

Whether or not a message is stored depends on two conditions. First, the signal strength at the receiver must exceed the receiver sensitivity. Otherwise, the message simply cannot be decoded. Second, among all messages that exceed the sensitivity threshold, only the RSS values of 24 random messages are saved in the database.

Figure 7.11a shows that in the experiments even for short distances only the RSS values of about half the broadcasted messages ended up in the database (e.g., a message that was sent over a distance of only 5 meters hand only a chance of about 50 % to be saved). This suggests that per time step on average about 48 of the 60 broadcasted messages could be decoded by the mobile node (twice as many as could be saved). As can be seen in Figure 7.11b, the simulations result in quite similar usage rates. Even the influence of walls on the probability that a message ends up in the database seems to be comparable.

Altogether, the comparison between experiments and simulations suggests a close correspondence of both in the considered statistical characteristics of signal decay.



(a) Real-world measurement

(b) Simulation

Figure 7.11: Usage rate of broadcasted messages in dependence of sender-receiver distance and number of penetrated walls. Of all messages whose signal strength at the receiver exceeds the receiver sensitivity, only the RSS values of 24 random messages are used (=stored) due to hardware limitations.

7.5 Summary

In this chapter, we laid the groundwork for the following chapters on RSS-based localization. First, we described the setup and the hardware of the sensor network experiments that are used in the following. Next, we analyzed the experimental data, showing the dependence of the received signal strength on the distance between sender and receiver and on the number of penetrated walls. The observations of the experimental study were then used to motivate the modeling of the simulations that are used in this thesis, and the simulation setup was described. Finally, a direct comparison between simulation and experiment was presented. It turned out that both experiment and simulation show similar behavior concerning the considered statistical aspects.

Chapter 8

RSS-based Position Estimation

Force-directed approaches, also known as spring embedders, have proven very successful in the areas of graph drawing and network embedding. In this thesis, we examine the application of force-directed methods to the problem of RSS-based tracking in wireless networks. To create a competitive force-directed tracking approach, reasonable representations of distance estimates and movement knowledge are necessary. In this chapter, we focus on the design of forces for the representation of RSS-based distance estimates. For this, we consider the static position estimation problem that does not involve movement of nodes. Several possible force definitions are examined, and a comparison with some other approaches for RSS-based position estimation is given.

8.1 Introduction

In this chapter, we consider the RSS-based position estimation problem, where a static node with unknown position analyzes the received signal strengths of wireless messages from beacon nodes with known positions to estimate its position.

This chapter is a preparation for Chapter 9, where additionally movement knowledge is used to track mobile nodes. The quality of tracking algorithms is determined by several intertwined factors, namely the efficient usage of knowledge from RSS measurements, the efficient usage of knowledge from movement models or inertial data, and finally the fusion of both kinds of data sources. Looking at the static position estimation problem makes it possible to study design decisions for the RSS-based component independently from effects that are caused by the use of movement information.

In the following sections, we study various kinds of localization approaches. The main focus lies on different possibilities to realize force-directed localization. RSS measurements are transformed into attractive and repulsive forces between the node that is to be located and the beacon nodes. The position estimate is then found by iteratively computing a force equilibrium. Several force definitions are compared, to study how RSS measurements can be efficiently incorporated into force-directed localization approaches.

To investigate which design decisions are important for good RSS-based localization and to get a feeling for achievable accuracies, some other position estimation techniques are included in the comparison, ranging from least-squares trilateration approaches to probabilistic approaches. The insights that are gained in this chapter will then be used in Chapter 9 for the design of a force-directed tracking approach.

Related work. This chapter on RSS-based position estimation presents mainly a preparation for the force-directed tracking approach that is introduced in the next chapter. As both chapters share large parts of the related work and our main focus is on the force-directed tracking, the overview on related work for both chapters is given in Section 9.1.1.

Note that it is not our intention to promote the force-directed position estimation approach as a novel approach to position estimation. Instead, we study it solely as a building block of the force-directed tracking approach.

8.1.1 Problem Definition

We consider a wireless network consisting of a node w with unknown position $p_w = (x, y)$ and a set (b_1, ..., b_n) of nodes with known positions, called *beacon nodes* or short *beacons*. Let $p_i = (x_i, y_i)$ denote the position p_i of beacon node b_i.

It is assumed that node w receives one wireless message m_i from each beacon b_i. Situations where several messages from the same beacon node b_i are received can easily be represented by assuming that all messages come from different beacon nodes b_j that are located at position $p_j = p_i$. The received signal strength S_i of message m_i is then used to estimate the distance between node w and beacon node b_i. Let $d_i = \text{dist}(w, b_i)$ denote the Euclidean distance between node w and beacon node b_i, and $\hat{d_i}$ the distance that is estimated based on the RSS S_i of message m_i. The goal is to estimate the position of w either directly based on the received signal strengths of the received messages, or based on the estimated distances to the beacon nodes.

8.2 Force-Directed Position Estimation

Force-directed approaches are widely used in the context of graph drawing and network embedding. Usually they are used to compute embeddings for whole networks. In this work, we examine to what extent force-directed approaches are suited for the problems of position estimation and tracking in wireless sensor networks.

8.2.1 Basic Idea

Basically, all applications of force-directed methods to compute embeddings of graphs and networks work similar: distance estimates between graph nodes are modeled as virtual *springs*, and an embedding is found by moving nodes until a force equilibrium is established [Ead84, FR91].

To model the considered position estimation problem by such a spring network, we introduce two types of nodes. For each beacon node b_i in the real network, we introduce an *anchor node* a_i in the spring network. The positions of the anchor nodes are fixed and correspond to the positions of the original beacon nodes. The node w that is to be located is represented by a so-called *position node* z, whose position is variable. Additionally, for each distance estimate a virtual spring is introduced. The springs are defined such that either an attracting force is exerted if the position node is farther away from the anchor node than expected, or a repelling force otherwise. If multiple measurements are available, multiple springs exert forces in different directions. Figure 8.1 depicts the described modeling.

Initially, the position of node z is initialized with a guess for the position of node w. Node z is then moved until a force equilibrium is found. The *position estimate* for node w can finally be inferred from the final position of node z. In comparison to the graph embedding problem, where the goal is to find an embedding for a whole network, the situation here is much simpler, as only a single node has to be embedded. As we will see in Chapter 9, things get more interesting in the tracking problem, where additional kinds of springs are introduced to represent movement knowledge.

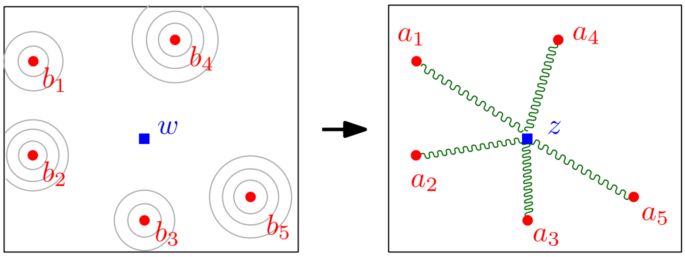


Figure 8.1: Signal strength measurements are translated into distance estimates, which then are represented by springs. The position estimate is computed by finding a force equilibrium. *Left:* Real-world sensor network. Circles correspond to beacon nodes, the square represents the device with unknown position. *Right:* Spring graph. Each beacon b_i is represented by an anchor node a_i, the device w that is to be located by a position node z. Distance estimates are represented by springs.

8.2.2 Forces for RSS-based Distance Estimates

An important aspect of the force-directed approach is that it makes almost no restrictions on the mathematical representation of the modeled forces. This allows the design of forces that are highly adjusted to the characteristics of the underlying measurements. In particular, one cannot assume that the same force definitions that work well for graph drawing are also optimally suited to represent distance estimates that are based on RSS measurements. In this section, we describe several force definitions that could be used to model RSS-based distance estimates. The performance of the different definitions will then be examined in the experimental section based on real-world measurements. As we will see, it pays off to adjust the force-definitions to the characteristics of the underlying measurements.

For each RSS-based distance estimate between node w and a beacon node b_i, we introduce one virtual spring between position node z and the corresponding anchor node a_i in the spring network. Let (x, y) be the current position of z, (x_i, y_i) the position of anchor node a_i, $\hat{d}_i$ the estimated distance from w to beacon node b_i, $\vec{v}_i = (x_i - x, y_i - y)^T$ the vector pointing from z to a_i, and $|\vec{v}_i|$ the Euclidean length of $\vec{v}_i$. The first force $\vec{F}_1(\hat{d}_i)$ considered in our study is defined as follows:

$$\vec{F}_1(\hat{d}_i) = \frac{\vec{v}_i}{|\vec{v}_i|} \cdot \hat{d}_i - \vec{v}_i \tag{8.1}$$

This definition corresponds to the desired behavior: If the estimated distance is larger than the current distance, then F_1 points in the direction of the anchor node, otherwise it points in the opposite direction. The length of the force vector is proportional to the absolute difference between the current distance and the estimated distance.

The second force definition is motivated by the graph drawing algorithm of Fruchterman and Reingold [FR91], which is known to work very well for the computation of graph embeddings, and which has also been used to compute embeddings of wireless sensor networks [EFI$^+$10]. For each range measurement, an attractive force $\vec{F}_{\text{attr}}$ and a repulsive force $\vec{F}_{\text{rep}}$ are introduced as follows:

$$\vec{F}_{\text{attr}}(\hat{d}_i) = -\frac{|\vec{v}_i|^2}{\hat{d}_i} \cdot \frac{\vec{v}_i}{|\vec{v}_i|}, \quad \vec{F}_{\text{rep}}(\hat{d}_i) = \frac{\hat{d}_i^2}{|\vec{v}_i|} \cdot \frac{\vec{v}_i}{|\vec{v}_i|}\,. \tag{8.2}$$

The acting force $\vec{F}_2$ is then given by

$$\vec{F}_2(\hat{d}_i) = \vec{F}_{\text{attr}}(\hat{d}_i) + \vec{F}_{\text{rep}}(\hat{d}_i) = -\frac{|\vec{v}_i|^2}{\hat{d}_i} \cdot \frac{\vec{v}_i}{|\vec{v}_i|} + \frac{\hat{d}_i^2}{|\vec{v}_i|} \cdot \frac{\vec{v}_i}{|\vec{v}_i|} \quad . \tag{8.3}$$

The remaining force definitions, $\vec{F}_3$ to $\vec{F}_6$, take into account that the average distance estimation error is higher for distance estimates based on weak signals (cf. Figure 7.7). To avoid that the position estimation is negatively influenced by inaccurate measurements over long distances, additional normalization factors are used to weaken the influence of measurements with low received signal strength. These normalization factors are either based on the estimated distance $\hat{d}_i$ to the considered beacon node b_i, or on the standard deviation σ_i of the estimation error that is expected for a distance estimation based on the received signal strength S_i of message m_i. The definitions of $\vec{F}_3$ to $\vec{F}_6$ are as follows:

$$\vec{F}_3(\hat{d}_i) = \frac{\vec{F}_1(\hat{d}_i)}{\hat{d}_i}, \quad \vec{F}_4(\hat{d}_i) = \frac{\vec{F}_1(\hat{d}_i)}{\hat{d}_i^2}, \quad \vec{F}_5(\hat{d}_i, \sigma_i) = \frac{\vec{F}_1(\hat{d}_i)}{\hat{\sigma}_i}, \quad \vec{F}_6(\hat{d}_i, \sigma_i) = \frac{\vec{F}_1(\hat{d}_i)}{\hat{\sigma}_i^2} \tag{8.4}$$

If data about received signal strengths and corresponding distance estimation errors are available, one can infer the σ-values directly from this data. Otherwise, the standard deviations can be estimated based on experiences with the used sensor network and knowledge about characteristic of the considered building. In this work, we computed the standard deviations directly by analyzing the data from additional experiments in the ITI and ITIV buildings.

The same normalization factors that are used in $\vec{F}_3$ to $\vec{F}_6$ could also be used in connection with $\vec{F}_2$. In our experiments, using a normalized version of $\vec{F}_2$ instead of one of the force definitions $\vec{F}_3$ to $\vec{F}_6$ did not result in improved localization results.

In principle, each of the described force definitions could be used to model RSS-based distance estimates in a force-directed approach. In our experimental study in Section 8.4.2, $\vec{F}_3$ and $\vec{F}_6$ have been found to be best suited for representing RSS measurements. If not stated otherwise, in this work $\vec{F}_3$ will be used to model RSS-based distance estimates.

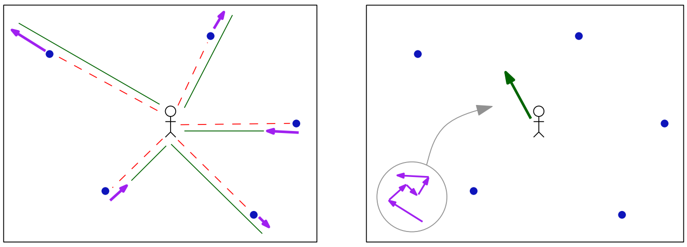
8.2.3 Initializing the Position Estimate

The initial position of z plays an important role in the performance of the force-directed position estimation. For one, the farther z is from the true position of w, the more iterations are necessary to move z to a position where all acting forces are in equilibrium. Additionally, it is not guaranteed that the force-directed improvement steps move z to some global optimum. Instead, if the initial position of z is chosen unfavorably, it is possible that the optimization ends in some local optimum, i.e., a force equilibrium that does not have to be close to the true position of w. Accordingly, it is worthwhile to put some effort into choosing a good initial position for z.

For the static position estimation considered in this chapter, initializing the position of z with the position of the beacon node from which the message with highest RSS was received worked very well in our experiments. However, depending on the density of beacon nodes, it is possible that this initial position estimate is several meters away from the true position. In the tracking problem as considered in the next chapter, the situation is much better. If the update frequency for the position estimation is sufficiently high, one can expect that the tracked device does not move far between two successive distance estimations. Thus, by initializing the new position to the old position, it is likely that the initial position estimate is already close to the final position of z.

8.2.4 Approximating a State of Equilibrium

Starting from the initial position of z, the position estimate is improved iteratively by moving z in small steps until a force equilibrium is reached. To get the movement direction for the next improvement step, all forces acting on z are simply added up (see Figure 8.2). In this work, all forces are considered to be 2-dimensional vectors. Accordingly, the movement direction can be computed very efficiently in time linear in the number of forces.



RSS measurement forces

Resulting movement direction

Figure 8.2: Computation of the movement direction for an incremental improvement step. *Left:* The dashed lines show the current distances according to the current position estimate, the solid lines show the estimated distances as inferred from the RSS measurements. The small arrows symbolize the resulting spring forces. *Right:* The final direction for the next movement is computed by adding up the force vectors.

The movement is done in small steps into the computed movement direction until either the length of the resulting force vector falls below some predefined threshold, indicating that the forces are almost in equilibrium, or a maximum number of improvement steps have been performed. The second rule assures that the approach does not run into an infinite loop.

For the force-directed position estimation, we use up to 80 steps with 20 cm movements, followed by up to 5 steps with 5 cm movements. After each movement, the acting forces are recomputed to get the new movement direction. The maximum number of improvement steps and the used displacement distances depend on the expected distance between the initial position estimate and the actual position of w. Thus, in situations where the initial position estimate is believed to be quite close to the real position, only a small number of improvement steps is necessary.

Instead of using a fixed displacement distance, one could also make the movement distance dependent on the length of the computed force vector. Yet another possibility would be to use some cooling approach like it is used in the Fruchterman-Reingold embedding algorithm [FR91], where the displacement distance is large in the beginning and becomes smaller over time. We examined both possibilities, but they did not result in an additional improvement of localization accuracy.

8.3 Reference Approaches

To put the localization results of the force-directed position estimation approach into a larger context and to get a feeling for achievable accuracies, we implemented some standard approaches for RSS-based position estimation in wireless networks. This section gives a brief overview.

8.3.1 Least-Squares Trilateration

Trilateration is a very intuitive method to compute position estimates from distances to known reference points. If the positions of the reference points and the corresponding distances are known exactly, one can compute the unknown position (x, y) by intersecting circles around three reference points with radii corresponding to the distance estimates. In reality, the distances, and often also the positions of the reference positions, are only known inaccurately. In these cases, the circles do not have to intersect in a single point.

If estimated distances $\hat{d}_i$ to more than three reference positions are known, one gets an overdetermined system of equations:

$$\begin{aligned} (x - x_1)^2 + (y - y_1)^2 &\approx \hat{d}_1^2 \\ &\vdots \\ (x - x_n)^2 + (y - y_n)^2 &\approx \hat{d}_n^2 \end{aligned} \tag{8.5}$$

In such a case, sometimes the expression *multilateration* is used instead of trilateration. Note that the equations only hold approximately, as the estimated distances $\hat{d}_i$ are assumed to contain errors. The usual approach to solve equation system (8.5) consists in finding a solution that minimizes the error. In the following, we consider the trilateration approach that is described for localization in wireless networks in [Bou09].

First, the non-linear system of equations is linearized by subtracting the last equation from the other equations. This gives the linear system

$$\underbrace{\begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\mathbf{x}} \approx \underbrace{\begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + \hat{d}_n^2 - \hat{d}_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 - \hat{d}_{n-1}^2 + \hat{d}_n^2 \end{bmatrix}}_{\mathbf{b}},$$

which can be solved using a standard least squares approach [GVL96], using the following parameter estimation:

$$\mathbf{x} = (A^T A)^{-1}(A^T \mathbf{b}) \tag{8.6}$$

As we will see in the experimental section, it actually makes a big difference which of the n equations is subtracted from the other ones. The larger the error in the equation that is chosen, the worse the final localization result. In the experimental part, we will label the variant that subtracts the equation corresponding to the measurement with highest signal strength *Trilat #1* and the variant that subtracts the equation of the measurement with the weakest signal strength *Trilat #2*.

One problem with the described approach is that it does not take the uncertainties of the measurements into account. To examine the difference this makes, we additionally implemented trilateration variants that are based on a weighted least-squares approach [Kay93, TBC11], using the parameter estimation

$$\mathbf{x} = (A^T W A)^{-1}(A^T W \mathbf{b}), \tag{8.7}$$

with weighting matrix W. Let σ_i^2 denote the variance of the distance estimation error corresponding to the signal strength S_i, which was measured for the packet from beacon node b_i. Assuming that the distance estimation errors are uncorrelated, W is chosen to be the diagonal matrix with $W_{ii} = 1/\sigma_i^2$.

We denote the weighted least squares approach that subtracts the equation of the strongest signal with *Trilat #3*, and the one that subtracts the equation of the weakest signal with *Trilat #4*.

8.3.2 Probabilistic Approaches

In probabilistic approaches, the measured RSS value is not mapped to a single distance, but to a probability distribution that assigns each distance a specific likelihood. By combining probabilities from different RSS measurements, it is possible to compute probabilities of presence for arbitrary positions in space. In this work, we compare four slightly different variants of probabilistic approaches.

The first considered approach is inspired by [SR04b], where a mobile beacon node b with known position is used to localize the nodes of a static sensor network. To keep the computational effort manageable, the possible locations are restricted to points on a 2-dimensional grid. Let X denote the set of valid x-coordinates, Y denote the set of valid y-coordinates, and $\mathcal{P}(x, y)$ denote the estimated probability that a certain static node is at position (x, y). In the beginning, all possible positions are assigned the same probability. Each time a static node receives a new message from the mobile beacon with signal strength S, the RSS information of the packet together with the current position (x_b, y_b) of the beacon is used to construct a new constraint on the position estimate of the static node. The updated probabilities $\mathcal{P}'(x, y)$ are computed as

$$\mathcal{P}'(x, y) = \frac{\mathcal{P}(x, y) \cdot P_S\left(d((x, y), (x_b, y_b))\right)}{\sum_{\bar{x} \in X} \sum_{\bar{y} \in Y} \mathcal{P}(\bar{x}, \bar{y}) \cdot P_S\left(d((\bar{x}, \bar{y}), (x_b, y_b))\right)} \quad \forall (x, y) \in X \times Y\,, \tag{8.8}$$

where $P_S\left(d((x, y), (x_b, y_b))\right)$ is the probability that a node receiving a packet with signal strength S is at distance $d((x, y), (x_b, y_b))$ from the sender (cf. Figure 7.7). This approach can easily be transferred to the considered scenario by assuming that the messages are not received from a single mobile beacon broadcasting from different positions, but from different static beacon nodes that broadcast from their corresponding positions.

To evaluate the performance of this approach, we consider two variants to infer the position estimate $(\hat{x}, \hat{y})$ from the computed probabilities. In the first variant (*Prob #1*), we assume that the position estimate corresponds to a grid point with maximum probability, i.e.,

$$(\hat{x}, \hat{y}) \in \underset{(x,y) \in X \times Y}{\arg\max}\, \mathcal{P}(x, y)\,. \tag{8.9}$$

In the second variant (*Prob #2*), which is the variant used in [SR04b], the position estimate is determined as a weighted average:

$$(\hat{x}, \hat{y}) = \left(\sum_{x \in X} \sum_{y \in Y} x \cdot \mathcal{P}(x, y), \sum_{x \in X} \sum_{y \in Y} y \cdot \mathcal{P}(x, y) \right)\,. \tag{8.10}$$

The second considered approach is based on Bayesian inference [Win03]. Again, we assume that the node w with unknown position receives a message with signal strength S from some beacon node b. Let $\bar{X}$ be the proposition that node w is at some given position (x, y) and $\bar{Z}$ the proposition that a message is received from beacon b with signal strength S. Using Bayes' theorem, the posterior belief $P(\bar{X}|\bar{Z})$ in $\bar{X}$ after $\bar{Z}$ is observed is then given by

$$P(\bar{X}|\bar{Z}) = \frac{P(\bar{Z}|\bar{X})}{P(\bar{Z})} \cdot P(\bar{X})\,, \tag{8.11}$$

where $P(\bar{X})$ is the *prior*, the degree of belief in $\bar{X}$ before $\bar{Z}$ is observed, and $P(\bar{Z}|\bar{X})$ is the *likelihood*, the probability that $\bar{Z}$ is observed, given that $\bar{X}$ holds. Transferred to the notation of Equation (8.8), this gives

$$\mathcal{P}'(x, y) = \frac{\mathcal{P}(x, y) \cdot P_{d((x,y),(x_b,y_b))}(S)}{\sum_{\bar{x} \in X} \sum_{\bar{y} \in Y} \mathcal{P}(\bar{x}, \bar{y}) \cdot P_{d((\bar{x},\bar{y}),(x_b,y_b))}(S)} \quad \forall (x, y) \in X \times Y\,, \tag{8.12}$$

with $P_{d((x,y),(x_b,y_b))}(S)$ being the probability that a message that is received from a sender in distance $d((x,y),(x_b,y_b))$ is received with signal strength S (cf. Figure 7.4).

Again, we consider the strategy of Equation (8.9), which uses a grid point with maximum probability as position estimate, and the variant of Equation (8.10), which determines the position estimate as a weighted average. In the following, we call the first variant *Prob #3* and the second variant *Prob #4*.

8.3.3 EKF Measurement Update

Tracking approaches that are based on extended Kalman filters (EKFs) use two kinds of updates to estimate the position of a moving object: *time updates* and *measurement updates*. The time updates use (noisy) knowledge about the movement of the object to predict position changes between consecutive time steps. The measurement updates incorporate additional knowledge that is gained by measurements, e.g., by RSS-based estimations of distances to beacon nodes. The weighting between both kinds of updates is controlled by the corresponding uncertainties, and the key to good tracking results lies in the combination of both. Nevertheless, in this chapter we are going to reduce EKF approaches to their measurement part. The intention behind this somewhat unusual approach is that we want to analyze how much knowledge an EKF gains from RSS measurements in comparison to the other approaches that are analyzed in this chapter. Due to the strong intertwining of both kinds of updates in the EKF, it is not easily possible to isolate the measurement update from the time update. For instance, the EKF uses a linearization of the measurement function around the current position estimate to approximate the non-linear system by a linear one. The more the position estimate differs from the real position, the larger are the effects this linearization exerts on the localization accuracy.

For our study of the measurement update, we use the tracking EKFs that will be described later in Section 9.3. To examine the measurement update under optimal conditions, we initialize the EKF with the true position (according to the reference path) as prediction and then analyze how the measurement update distorts this ideal position estimate. To minimize the influence of the prediction, the corresponding covariances are set extremely high so that in the final result only the measurement result is weighted. In our experiments, this approach is labeled *EKF Meas #1*.

Additionally, to get a feeling how the initial position estimate influences the final localization result, we implemented a second variant that initializes the position estimate to the position of the beacon node from which the strongest signal was received. This approach is labeled *EKF Meas #2*.

8.4 Evaluation

8.4.1 Experimental Setup

To evaluate the different approaches for RSS-based position estimation, we use the experimental data from the sensor network experiments in the ITI building (cf. Section 7.1). During each time step of 0.25 seconds, RSS measurements of messages from up to 24 different beacon nodes have been collected. As the person carrying the mobile sensor node can only move a very short distance during that time interval, it is justified to assume that all packets have been received at the same position. Thus, by considering the measurements of each time step as a single position estimation instance, our experiments at the ITI give a total of about 4600 real-world instances.

In our evaluation, we consider the influence of the number of used measurements and the number of beacon nodes on the localization accuracy. If less than 24 measurements are used, the measurements with highest signal strength are chosen. The intention behind this is that the average distance estimation error on average increases for estimations based on weaker signals. Thus, it can be better to use only measurements with high RSS. Instead of using the n strongest signals, one could achieve similar results by defining a RSS threshold below which measurements are simply ignored.

For the experiments where only a subset of the 60 available beacon nodes is used, the beacons are chosen as follows: First, we selected the two beacons with maximum distance from each other among all beacons. Subsequently, we iteratively chose the beacon that maximized the minimum distance to any already selected beacon, until the intended number of beacons was reached. The motivation behind this strategy is to get a set of beacons that are preferably uniformly distributed.

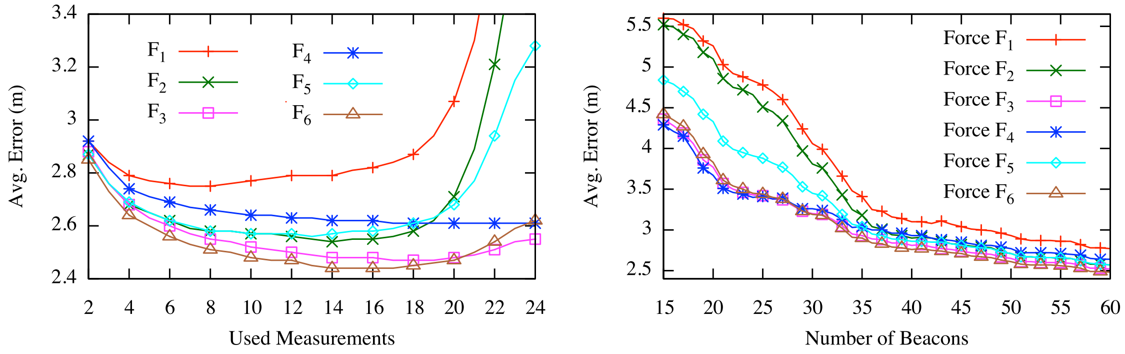
In the experiments that focus on the number of used measurements, all 60 beacon nodes are used. In the experiments that analyze the influence of the number of beacon nodes, for each localization instance the 10 measurements with highest RSS are used.

To compare the localization accuracies of different position estimation techniques, we consider the *average localization error*, i.e., the average distance between the computed positions and the true positions.

8.4.2 Results

Different force definitions. We start with a comparison of the different force definitions that were presented in Section 8.2.2. Figure 8.3a shows how the localization accuracy depends on the number of used measurements. Apparently, when $\vec{F}_1$, $\vec{F}_2$, or $\vec{F}_5$ are used, the localization error increases quickly if more than 18 measurements are used. The reason is that $\vec{F}_1$ and $\vec{F}_2$ do not use a weighting that weakens the influence of measurements with low signal strength, which usually result in higher distance estimation errors. Force definition $\vec{F}_5$ uses a weighting, but it seems that dividing the force by the standard deviation corresponding to the measured RSS is not sufficient for very weak signal strengths. In the considered experiments, the best results are achieved by using either $\vec{F}_3$ or $\vec{F}_6$ in connection with about 14 RSS measurements.

When looking at the influence of the number of beacon nodes in Figure 8.3b, it is not surprising that the localization accuracy improves when more beacon nodes are used.



(a) Error in dependence of used measurements.

(b) Error in dependence of beacon number.

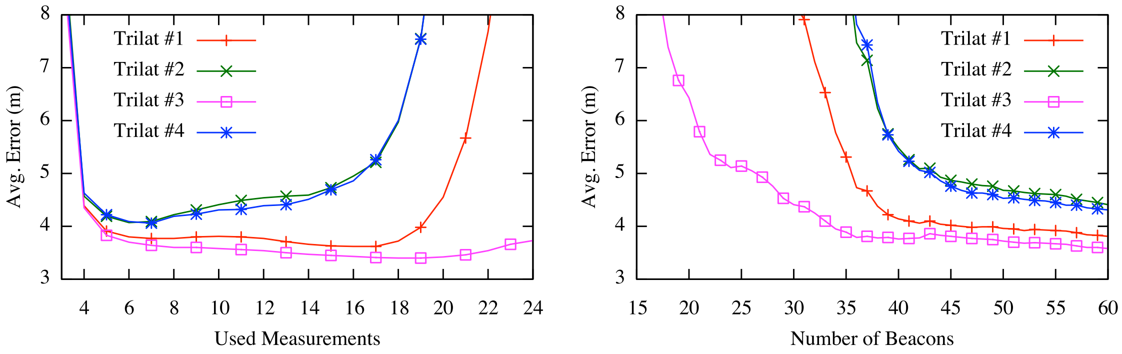
Figure 8.3: Position estimation error for different force definitions.

Closer inspection of the plots reveals, that for less than 35 beacons the average localization error starts to increase quickly for forces $\vec{F}_1$ and $\vec{F}_2$, and to some degree also for force $\vec{F}_5$. The reason for this effect is the same that has already been observed in the plots dealing with the influence of the number of used measurements. As a fixed number of 10 RSS measurements is used independent of the number of beacon nodes, the weakest of the 10 measurements corresponds to measurements over larger distances when less beacon nodes are used. It seems that for 35 used beacons and below, bad distance estimates come into play. As we will see in the next paragraphs, all approaches that do not use a weighting based on the RSS will perform significantly worse when less than 35 beacons are used.

Trilateration approaches. The first observation that strikes the eye when looking at the plots for the trilateration approaches in Figure 8.4 is that it makes a big difference which equation of equation system (8.5) is subtracted from the other equations to linearize the equation system. *Trilat #2*, the unweighted approach that subtracts the equation that corresponds to the weakest of the considered signals, produces significantly larger errors than *Trilat #1*, the unweighted approach that subtracts the equation of the strongest signal.

Comparing *Trilat #3* with *Trilat #1* shows that the additional weighting results in improved position estimates. Particularly, it avoids that the localization results become much worse when more than about 18 measurements are used.

Interestingly, for the trilateration variant that subtracts the equation of the weakest signal, the additional weighting has almost no effect. The reason is that for this approach the large distance estimation error that can be expected due to the low signal strength is added (without weighting) to all other equations.



(a) Error in dependence of used measurements.

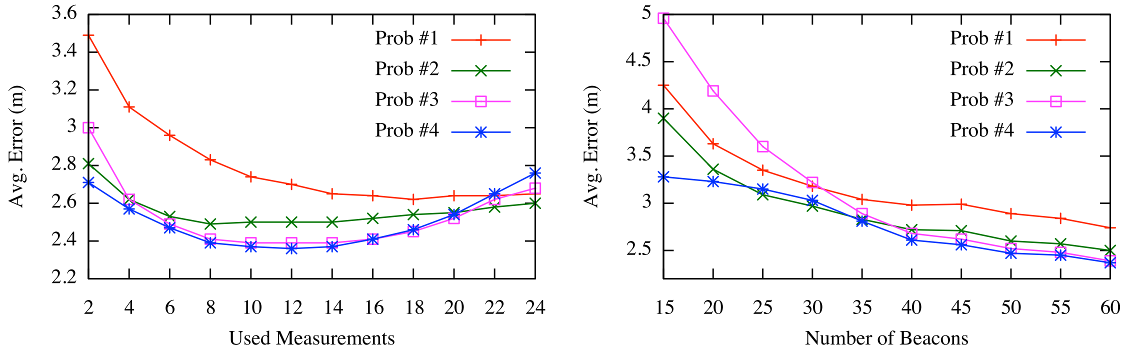
(b) Error in dependence of beacon number.

Figure 8.4: Position estimation error of different trilateration approaches.

The plot that considers the error in dependence of the number of beacon nodes shows a similar picture, with *Trilat #3* performing best in all cases. However, in comparison with the force-directed approaches, even *Trilat #3* performs rather bad. This is a little surprising, as dedicated trilateration approaches are considered. It seems that the considered analytical trilateration approach is rather unsuited for RSS-based localization in wireless networks. Instead, iterative trilateration approaches that cope without linearization might be preferable, e.g., [LLW06].

Probabilistic approaches. For the probabilistic approaches, we computed for each localization instance the probabilities according to equations (8.8) and (8.12) for all positions on a grid of dimensions 50 m × 50 m, centered at the true position. The used grid spacing was 25 cm.

The average localization errors are shown in Figure 8.5. We observe that even for the probabilistic approaches, the localization accuracy can be improved by using only a subset of the available measurements. Among the probabilistic approaches, *Prob #4*, the variant that is based on Bayesian inference and that computes the position estimate as a weighted average, performs best in our experiments.



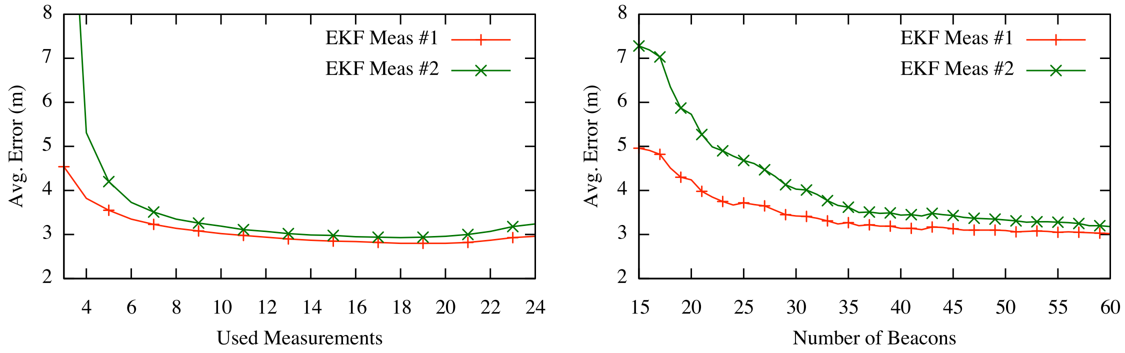
(a) Error in dependence of used measurements.

(b) Error in dependence of beacon number.

Figure 8.5: Position estimation error for different probabilistic approaches.

EKF measurement update. In the EKF, the influence of single measurements is weighted based on the corresponding covariances. In consequence, the measurement update is relatively robust even when measurements with low RSS are included.

As the EKF uses a linearization around the current position estimate, the localization result also depends on the quality of the predicted position. In the experiments with all 60 beacon nodes, the difference between *EKF Meas #1*, the variant that initializes the position estimate to the reference position, and *EKF Meas #2*, the variant that initializes the estimate to the position of the beacon from which the strongest signal was received, is rather small as long as a sufficient number of measurements is used.



(a) Error in dependence of used measurements.

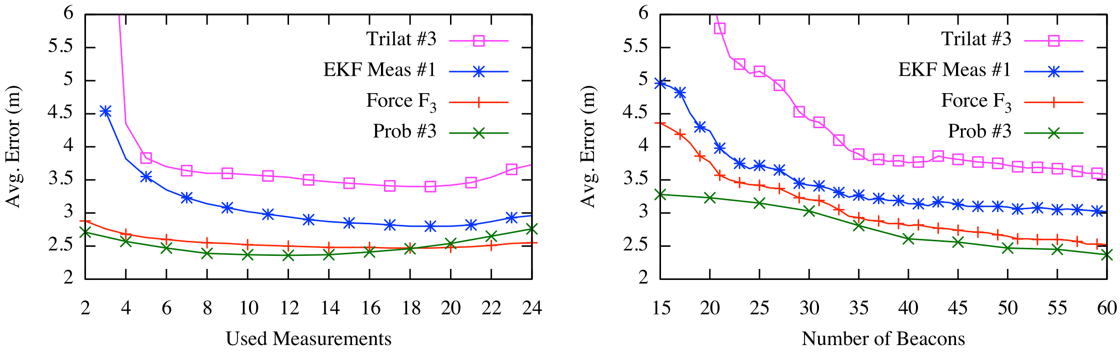
(b) Error in dependence of beacon number.

Figure 8.6: Position estimation error of the EKF measurement updates.

Figure 8.6b shows that the performance of *EKF Meas #2* gets significantly worse when only few beacon nodes are used. As weighting is used, this is probably not caused by the use of weak RSS values. Instead, if less beacons are available, then the initial position estimate, i.e., the position of the beacon with highest RSS, is on average further away from the true position. Thus, the linearization is computed at a position that is more distant to the true position. This seems to have an additional negative influence on the localization accuracy.

Comparison. We conclude this study with a brief comparison between the top candidates of the different kinds of approaches in Figure 8.7. In our experiments, the probabilistic approach *Prob #3* achieved the best localization results. This is not surprising, as this is also the most complex approach with the highest computational costs. Surprisingly, the iterative force-directed approach was able to achieve similar accuracies, using much less computational effort.

The trilateration approaches and the EKF measurement update resulted on average in higher localization errors. The higher errors are probably caused by the linearization of the problem.



(a) Error in dependence of used measurements.

(b) Error in dependence of beacon number.

Figure 8.7: Comparison of different kinds of approaches.

8.5 Summary

In this chapter, we took a look at the problem of RSS-based position estimation in wireless networks. For the force-directed approach, we analyzed different possibilities to represent RSS-based distance estimates by forces. Additionally, to get a feeling for achievable accuracies and to see how different design decisions influence the localization result, we implemented and examined several other kinds of localization approaches.

According to our experiments, the force-directed approach allows the computation of competitive position estimates. As we will see in the next chapter, it is very easy to extend the force-directed position estimation approach to more complex problems, e.g., the tracking of mobile devices, by introducing additional kinds of forces.

Chapter 9

Force-Directed Tracking

In this chapter, we study the force-directed tracking of mobile devices based on signal strength measurements. In the considered tracking problem, it is assumed that additional knowledge about the movement of the mobile device is available. This knowledge can originate from movement model assumptions or from inertial data. As an example application, we consider the tracking of mobile devices that are carried by pedestrians, using a network of wireless beacon nodes with known positions. We show how the force-directed position estimation approach that was presented in the last chapter can be easily extended with additional kinds of forces that model the additional movement knowledge. The proposed force-directed tracking approach is then compared to a similar model in an extended Kalman filter (EKF), and similarities and differences between both kinds of approaches are examined. Finally, the performance of both approaches is evaluated based on real-world experiments and simulations.

9.1 Introduction

In the tracking problem that is considered in this chapter, the position of a mobile device that is carried by a pedestrian in an indoor environment has to be estimated continuously over time. Of course, one could repeatedly use one of the position estimation techniques that were studied in Chapter 8 to keep track of the position of the mobile device. Figure 9.1 shows an example of a trajectory that was computed in this way. For the computation of the trajectory, the force-directed position estimation technique from Section 8.2 was used once for each time step.

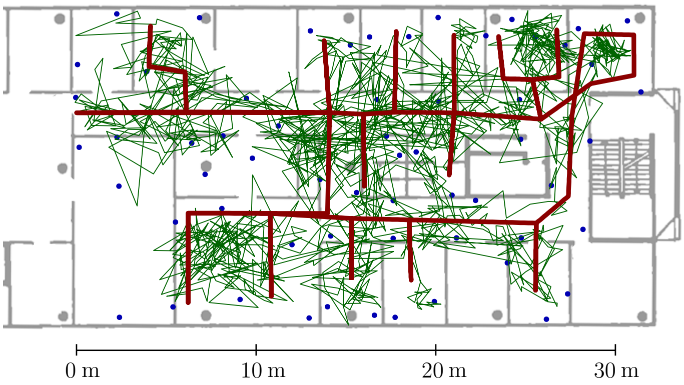


Figure 9.1: Computed path using the force-directed position estimation once per time step. The red lines show the reference trajectory, the green lines connect successive position estimates.

Apparently, due to the high fluctuation of RSS measurements, the position estimates jump significantly around the true position. This behavior is not limited to the force-directed approach, but all approaches that are discussed in Chapter 8 show similar behavior when used for RSS-based tracking. If additional information about the movement of the device is available, e.g., in form of movement model assumptions or inertial data, this information can be used to get significantly improved localization results. In this chapter, we examine how such additional information can be utilized in a force-directed tracking approach.

9.1.1 Related Work

In this overview on related work for chapters 8 and 9, we mainly focus on three lines of research: First, the localization of static devices based on signal strengths. Second, the RSS-based localization of mobile devices. And third, force-directed methods for localization in wireless sensor networks.

RSS-based position estimation. For the localization of static devices based on signal strengths, there exist two main approaches: fingerprinting and range-based methods.

In fingerprinting, a map of signal strengths to all beacon nodes is created for various positions in an initial training phase and stored in a database. Every time when a node is to be localized, the signal strengths that the node measures are compared to the signal strengths in the database, and the position with closest correspondence to the measured signal strengths is returned. Examples for fingerprinting approaches are [BP00, PAK$^+$05, ACZ05, LW07, KPV07, YYN08, AC09, PCC$^+$10, BOGVB10].

In range-based methods, knowledge about the signal decay over distance is utilized to estimate distances based on the received signal strengths. The position estimate is then computed using some trilateration approach, based on the distance estimates in combination with the known beacon node locations. In the circular positioning algorithm [Li06, LLW06], a gradient method is used to iteratively minimize the sum of the squared differences between the estimated distances and the true distances. Another trilateration approach is the hyperbolic positioning algorithm [LLW06], in which a linearization is used to transform the non-linear position estimation problem into a linear problem. Subsequently, a least-squares approach is used to computed an optimum solution for the linearized problem. Both the circular positioning algorithm and the hyperbolic positioning algorithm have been extended in [TBC11] with an additional weighting of distance estimates based on their uncertainties.

Especially in wireless sensor networks, many approaches for localization use trilateration based on hop-distances instead of RSS-based distances, e.g., [NN01, SPS02].

In addition to trilateration approaches, there also exist probabilistic approaches to turn RSS measurements into position estimates. Probabilistic approaches utilize knowledge about the probabilities that certain signal strengths are measured for a considered distance. This usually allows for higher localization accuracy at the prize of higher computational costs. An example for such a probabilistic approach is [SR04b], where measurements of a GPS-equipped mobile node are used to localize static sensor nodes.

Surveys with additional information on wireless position estimation can be found in [SCGL05, Gez08].

RSS-based tracking. Most approaches for RSS-based tracking utilize additional information about the movement of the tracked object to improve the localization result. An exception are the aforementioned fingerprinting methods, which are frequently used without additional knowledge. However, the relatively high effort of the initial map-building phase makes fingerprinting rather inapplicable for many application scenarios of RSS-based tracking. Besides fingerprinting, there exist two popular kinds of approaches that are mostly used for tracking based on RSS measurements: variants of Kalman filters and particle filters.

Kalman filters themselves are optimal estimators for linear systems if the noise is normally distributed and zero-mean and the covariances of the noise are exactly know [May79]. However, due to the non-linear nature of the localization problem, non-linear variants of Kalman filters have to be used. One of the most well-known variants is the extended Kalman filter (EKF). EKFs are frequently used for RSS-based tracking in wireless networks, e.g., in [RM09, PW09, SVG$^+$10, YJGJ10, SGSMG11, SBN$^+$11]. Besides the EKF, sometimes the newer unscented Kalman filter (UKF) [JUDW95] is used for tracking in wireless networks [SKL10, SBN$^+$11]. Like the EKF, the UKF is a non-linear filter, but in contrast to the analytical EKF, the UKF uses sample-points, called sigma-points, to represent the probability distribution.

Both EKF and UKF can be used to fuse information from different data sources into one state estimate. Examples for the use of EKFs in connection with RSS-based distance estimates and information from an inertial measurement unit (IMU) can be found in [RGHR10, JSPG10].

Particle filters are simulation-based model estimation techniques, which represent probability distributions by a set of randomly chosen weighted samples. In contrast to EKFs, they allow to represent non-Gaussian and multi-modal probability distributions. This usually allows for improved localization accuracies. However, to achieve this improved estimation, particle filters also have significantly higher computational costs than EKFs. Like EKFs, particle filters are frequently used for the tracking of mobile devices in wireless networks, e.g., in [MNR$^+$06, WLS$^+$07, GWJ$^+$11, CEL$^+$11].

Force-directed localization approaches. Originally, force-directed methods have been used in the context of VLSI design [QB79]. However, it was soon realized that they are also very well suited to compute embeddings of graphs and networks. The introduction of force-directed methods to the area of graph drawing is usually accounted to an early work by Eades [Ead84], and even today force-directed methods are still frequently used in the area of graph drawing, e.g., [DEG$^+$12].

As graph drawing problems and network embedding problems are closely related, it is not surprising that force-directed methods are frequently used to compute or to improve embeddings of sensor networks. For example, in the anchor-free localization approach by Priyantha *et al.* [PBDT03], a force-directed relaxation method is used to refine an initial embedding of the network. Similarly, in [MLRT04] a force-directed relaxation was proposed as an optional refinement after an initial cluster localization. In [EFI$^+$10], Efrat *et al.* compare several force-directed approaches to find embeddings of sensor networks based on signal strengths and angular information. A distributed force-directed algorithm that computes a network embedding based on local distance and angular information is introduced in [CKKK10].

All these approaches consider the embedding of whole sensor networks. To the best of our knowledge, the application of force-directed methods to the problem of RSS-based tracking has not yet been studied.

9.1.2 Problem Definition

We assume that a wireless network consisting of n *beacon nodes* (b_1, ..., b_n) with known positions is distributed in the 2-dimensional plane. Let $p_i = (x_i, y_i)$ denote the position of beacon node b_i. Each beacon node sends a message containing its ID and position once per time step. A person carrying a mobile device m is moving through the network. The true position of m in time step t is denoted by $p^t_m = (x^t_m, y^t_m)$. The mobile device is used to receive the broadcasts of the stationary beacon nodes, and the distances to the beacon nodes are estimated based on the received signal strengths. For time step t, the true distance between m and beacon b_i is denoted by $d_i^t = \text{dist}(p_m^t, p_i)$ and the distance estimate that is based on the received signal strength is denoted by $\hat{d}_i^t$. If m did not receive a message from b_i in time step t, then $\hat{d}_i^t$ is undefined.

The goal is to estimate the position p_m^t of m for each time step $t \in \{1, \dots, T\}$, with T being the total number of time steps. To estimate the position at time t, all RSS measurements that have been received up to that time can be used. Additionally, it is allowed to use additional movement knowledge. In this work we consider the use of assumptions about the movement of pedestrians and the use of step recognition data from an inertial measurement unit (IMU).

To compare the localization accuracy of different tracking approaches, we again use the *average localization error*. In the context of tracking, this is the average of the distances between the true positions and the computed positions. Let $\hat{p}_m^t$ denote the estimate for the position of m at time t. The average localization error $\mathcal{E}$ is then computed as

$$\mathcal{E} := \sum_{t=1}^{T} \text{dist}(\hat{p}_m^t, p_m^t)/T \quad . \tag{9.1}$$

9.2 Force-Directed Tracking Algorithm

Although the basic principle of the force-directed tracking approach is very similar to the one of the force-directed position estimation approach that was presented in Section 8.2, some changes are necessary in order to model the dynamic nature of the tracking problem. Like in the force-directed position estimation, we introduce for each beacon node b_i of the real network an *anchor node* a_i in the spring network. The positions of the anchor nodes are fixed and correspond to the positions of the original beacon nodes. However, instead of introducing a single *position node* z that represents the position of the mobile node m, we introduce a separate position node z_t for each time step $t \in \{1, \dots, T\}$. The reason for the introduction of multiple position nodes is that this allows to model additional forces between different position estimates, e.g., to represent movement knowledge. Note that it usually is not necessary to consider more than two position nodes at once, one for the current and one for the previous time step. Knowledge from earlier time steps is not necessary, as the current position only depends on the previous position and the movement that occurred meanwhile.

For each time step t, all RSS-based distance estimates $\hat{d}_i^t$ to some beacon node b_i are represented by a corresponding spring between position node z_t and anchor node a_i in the spring network. In this work, we use for RSS-based distance estimates the force definition $\vec{F}_3$ as defined in Equation 8.4.

Figure 9.2 visualizes the described modeling. Note that the modeling described so far results in exactly the same tracking results as using the simple force-directed position estimation approach for each time step separately. The reason is that we do not yet combine information between different time steps. How such a combination can be established is the topic of the following sections.

9.2.1 Forces from Movement Model Assumptions

Due to the high fluctuations in RSS-based distance estimations, the position estimate can jump in the range of meters, even if the mobile node does not move at all. To avoid this, we use the fact that the time difference between consecutive measurements is rather

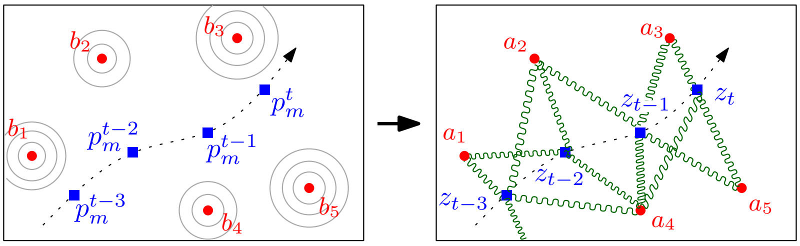


Figure 9.2: Basic modeling in the force-directed tracking approach. *Left:* Real-world network. Circles correspond to beacons, squares to the positions at which the mobile device computes distance estimates to the beacons. *Right:* Spring graph. Beacons are represented by anchor nodes, measurement positions by position nodes. Distance estimates are represented by springs.

small (1/4 second in our experiments and simulations), meaning that the person carrying the mobile device cannot move far within this time. Assuming a maximum velocity $v_{\max}$ of the tracked object (i.e., the person), a maximum distance $\Delta t \cdot v_{\max}$ is assumed for the movement of the receiver during one time period Δt. To model this, we introduce additional spring forces between consecutive position nodes (cf. Fig. 9.3).

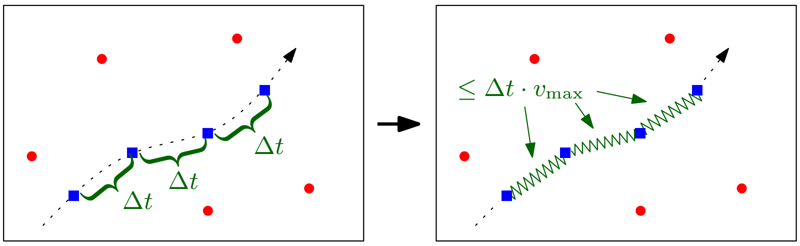


Figure 9.3: Time differences are translated into distance estimates between measurement nodes.

For example, a pedestrian usually does not move faster than $v_{max} = 5\,\text{km/h}$. From this we can infer that consecutive measurement positions should be no further apart than $d_{max} = \Delta t \cdot v_{max}$, with Δt being the time difference between two measurements.

Let $(\hat{x}_m^{k-1}, \hat{y}_m^{k-1})$ be the final position estimate for m, i.e., the final position of position node z_{k-1} in time step $k-1$, and $\vec{w} = (\hat{x}_m^{k-1} - \hat{x}_m^k, \hat{y}_m^{k-1} - \hat{y}_m^k)^T$ the vector pointing from the current position of node z_k to the position of z_{k-1}. By introducing the force

$$\vec{F}_{\text{move}} = \begin{cases} \vec{0} & \text{if } |\vec{w}| \leq d_{\max} \\ (|\vec{w}| - d_{\max}) \cdot \vec{w}/|\vec{w}| & \text{if } |\vec{w}| > d_{\max} \end{cases}, \tag{9.2}$$

on position node z_k, with $|\vec{w}|$ being the Euclidean length of vector $\vec{w}$, we model an attracting force that acts on z_k as soon as it is located more than distance $d_{\max}$ away from the previous position node z_{k-1}. This force is proportional to the amount distance $d_{\max}$ is exceeded. If the position nodes are closer than $d_{\max}$, no force is introduced. This simple example shows how easily further knowledge can be modeled by additional forces.

9.2.2 Combination with Step Recognition

Today, most mobile devices have a built-in accelerometer, which can be used to identify steps by recognizing peaks in the acceleration pattern. As an example of how to extend the system model with additional information, we study the possibility of incorporating such step information. At this point, we do not want to go into the technical details of step detection, but we just assume that we know when the person takes a step forward.

Until now, we assumed that the position of the mobile node is re-estimated every time step, using the new RSS measurements that arrived since the last time step. For each time step, a new position node was introduced and the oldest position node was removed from the network. If movement step information is available, we use a slightly different approach. Now, new position nodes are only introduced if steps are detected. If new distance measurement are obtained, the corresponding forces are simply attached to the current position node. Again, a connection of the current step node with the previous step node is incorporated by an additional force similar to $\vec{F}_{\text{move}}$ from the last paragraph (cf. Figure 9.4)

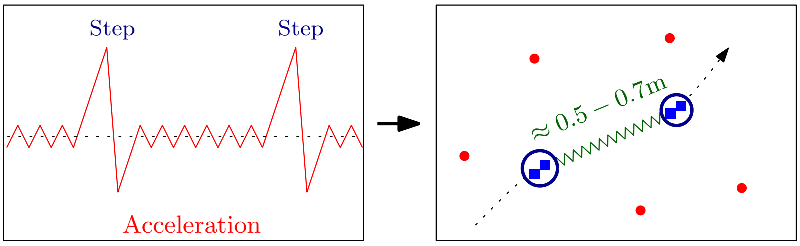


Figure 9.4: Step detection is used to detect and combine co-located measurements. The natural length of the spring that connects consecutive position nodes is set to a typical step length.

There are two differences to the model without step recognition: $d_{\max}$ is now set to an average step length and, as we now believe that the person actually moved, one can also make the force between consecutive position nodes repelling if the distance between the position nodes is shorter than a usual step.

If the person does not move for a while, it can happen that many measurements accumulate between two steps. In this case one does not have to use all measurements. Instead, it is usually sufficient to use only the k latest measurements, with k being chosen appropriately for the considered application.

9.2.3 Approximating a State of Equilibrium

After determining all forces that are exerted by the introduced springs, a state of force equilibrium, i.e., a position where all forces are in equilibrium, has to be found. In this work, all forces are considered to be 2-dimensional vectors. For each kind of force, we define an additional weighting that allows to control the importance of the forces relative to each other. By adding the vectors multiplied by the corresponding weight, we get the direction to which to move the position estimate. The movement is done in small increments of 5 cm to 10 cm. After each movement, the acting forces are recomputed to get the new movement direction. The number of necessary iterations depends on the scenario. For tracking without step detection we use 7 iterations with 10 cm movements, followed by 3 iterations with 5 cm movements. In the model with step detection, we recompute the position only once per step. Accordingly, we use a slightly higher number of iterations: 12 iterations with 10 cm movements followed by 3 iterations with 5 cm movements. More dynamic position changes could be enabled by greater displacement distances or a higher number of iterations. Of course, iterations could also be done as long as an improvement is achieved and interrupted otherwise.

9.2.4 Possible Extensions

There are many possibilities to extend the presented force-directed tracking approach with additional knowledge, e.g., to incorporate data from additional sensors. In this section, we briefly mention some possibilities for natural extensions. However, the ideas presented in this section will not be further analyzed in this work.

Using absolute orientations. Today, many mobile devices are equipped with electronic compasses. Data from such a compass could be used to get the absolute orientation of the moving object. In combination with step recognition, this kind of information could be easily incorporated in the force-directed tracking approach, by assuming that the next position node is preferably located one step distance in the direction of the estimated movement direction. To allow for uncertainties, the forces could be designed such that they attract the position node to the preferred new location and that the force is increasing with increasing distance to that location.

Using relative orientations. Instead of using absolute orientations from a compass, a gyroscope sensor can be used to estimate relative orientation changes between time steps. Lately, gyroscopes also became increasingly popular in consumer hardware such as cell phones. The advantage of gyroscopes in comparison to compasses is that they are not influenced by magnetic fields, which often disturb electronic compasses in indoor environments. Figure 9.5 illustrates how information from a gyroscope can be used in a force-directed tracking approach. The previous orientation is updated with the estimated rotation and the expected position is placed at a distance that corresponds to one step length. An attracting force is added, which pulls the position node towards the expected position.

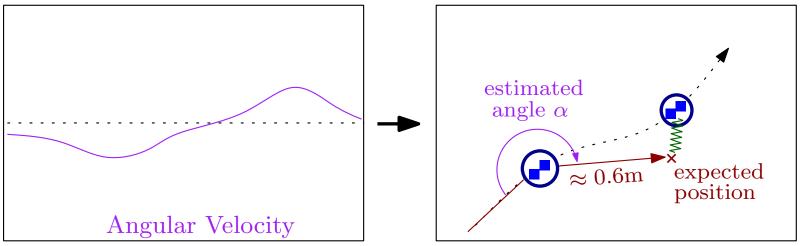


Figure 9.5: Gyroscope measurements can be used to compute relative orientations between consecutive steps.

Note that in order to infer the previous orientation from the position estimates of the force-directed tracking approach, it is not sufficient to consider only the previous position node and the position node before the previous node. Due to the high fluctuation of position estimates, the estimated orientation could be significantly off. Instead, the orientation of position nodes that are several time steps apart from each other should be considered.

Keeping position estimates variable. In this work, we assume that in each time step t only the position of the newest position node z_t is flexible. However, in some scenarios it might be desirable that measurements of later time steps are also used to improve the position estimates of earlier time steps. In the force-directed approach, this can be realized by considering in each time step more than only the last two position nodes. For example, at time t one could keep the position of node z_{t-k} fixed, and release the positions of nodes $\{z_{t-k+1}, \ldots, z_t\}$, with k being an appropriate constant. For nodes z_i with $t-k < i < t$, an additional force similar to F_{move} is introduced that attracts z_i to z_{i+1} if the corresponding distance is larger than expected. New positions for nodes $\{z_{t-k+1}, \ldots, z_t\}$ are then computed by iteratively finding force equilibria for all nodes.

9.2.5 Time and Space Complexity

Both time and space demand of the force-directed tracking approach are very low. Depending on the scenario, the algorithm can either store the coordinates of the beacons in terms of anchor nodes, or, if the beacons include their positions in the broadcast, it is not even necessary to store anchor nodes. For position nodes as well, only coordinates have to be stored. At no point in time more than two position nodes have to be kept in memory—one for the current position estimate and one for the previous position estimate, to allow the use of movement forces. In each iteration of a measurement update, the movement direction is determined by simply adding up 2-dimensional force vectors. This can be done very efficiently in time $O(f)$, with f being the number of forces. As the number of iterations is fixed (10-15 in our experiments and simulations), the complexity of the whole measurement update also is in $O(f)$. Summarizing, both time and space complexity of measurement updates are linear in the number of involved forces. This makes the force-directed approach very attractive even for application on devices with very low computational power or little memory.

9.3 Extended Kalman Filters

Extended Kalman filters are frequently used for RSS-based location tracking. In this work, we use the EKF as a reference approach to the force-directed tracking and examine similarities and differences between both approaches. For the EKF, we keep the description to a minimum and refer to the standard literature on this topic, e.g., [May79]. The descriptions and implementations of our EKFs are based on the tutorial on Kalman filters by Welsh and Bishop [WB01].

9.3.1 EKF with Movement Model

Similar to the model established in Section 9.2.1, we use the knowledge that a person can only move a limited distance between two consecutive measurements. In this section, we use a notation that is slightly different from the one introduced in Section 9.1.2, to stay closer to the notation that usually is used in connection with Kalman filters. The process state at time step k consists of the two-dimensional coordinates (x_k, y_k), which represent the position of the mobile node at time k, and movement is introduced in form of process noise. This results in the following process model

$$\vec{x}_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} x_{k-1} + w_1 \\ y_{k-1} + w_2 \end{pmatrix}, \tag{9.3}$$

with noise terms w_1 and w_2 that are assumed to be normally distributed and zero-mean. The measurement $\vec{z}_k$ at time step k is then given by

$$\vec{z}_k = \begin{pmatrix} \sqrt{(x_k - x_{a_1})^2 + (y_k - y_{a_1})^2} + v_1 \\ \vdots \\ \sqrt{(x_k - x_{a_m})^2 + (y_k - y_{a_m})^2} + v_m \end{pmatrix}, \tag{9.4}$$

where (x_{a_i}, y_{a_i}) is the position of the i-th beacon node used at time k and v_i is the measurement noise for the distance estimation to this beacon.

If not stated otherwise, we use the $m = 10$ measurements with highest RSS in the measurement update, as this offers a good compromise between computational effort and achievable accuracy. This means that the set of beacons that are used for the localization changes from one time step to another. The variance of v_i depends on the received signal strength, as distance estimations based on weak signals involve higher uncertainties than distance estimations based on strong signals. As described in Section 7.3, we inferred the uncertainties of the distance estimations directly from a set of training data. Further information on the definition of the process noise will be given in Section 9.5.1.

9.3.2 EKF with Step Detection

Like in Section 9.2.2 for the force-directed approach, the EKF can also be extended with step information in the system model. We now separate process updates and measurement updates. While measurement updates still occur every time when new RSS measurements arrive, process updates are only performed when a new step has been detected. Additionally, the variance of the process noise in equation (9.3) is adjusted to reflect the expected position change during one step. Further information on the used process noise is given in Section 9.5.1.

Both the EKF and the force-directed approach offer the possibility to design more sophisticated models, for instance by including velocities in the process model. We refrained from using more complex models in both approaches, as the goal of this work is to demonstrate similarities and differences of both approaches under simple and comparable conditions.

9.4 Force-Directed Tracking vs. EKF Tracking

Overall, the modeling of the force-directed approach and the described EKF is done in a similar way. In the force-directed approach, the forces that are exerted by the springs between consecutive position nodes ensure that the position estimate does not jump too far between consecutive time steps. In the EKF, the uncertainty of the process state controls how much the position estimate can change. However, there also are some distinctions between both approaches.

Concerning the measurements, the EKF uses a linearization around the current position estimate. This linearization can introduce an additional error, which depends on the quality of the current position estimate. In the force-directed approach, distances can be modeled directly, using non-linear terms.

Both the force-directed approach and the EKF combine a measurement with a prediction from a movement model. In the EKF, prediction and measurement update are initially computed separately and then fused based on the covariance estimates that describe the uncertainties of prediction and measurement. In contrast, in the force-directed approach, both measurement and movement forces are considered concurrently. The position estimate is computed by finding a force equilibrium for all kinds of forces at the same time.

As there is no directional information available, the EKF uses a normal distribution with mean at the previous position as prediction for the position of the next time step. This means that the last position is the most likely one and the likelihood that the mobile device is in a certain position decreases with distance to the last position estimate. In the force-directed approach, the movement force only acts if the distance to the previous position estimate exceeds some distance $d_{\max}$. This means that all positions within distance $d_{\max}$ are equally likely with respect to the movement model. If step recognition is used, the EKF assumes after a step is detected that the last position estimate is still the most likely position, but with an increased uncertainty. In the force-directed approach, one can model that an attractive force acts if the positions are too far apart and an additional repulsive force if they are too close together. This would result in a complex probability density function, which could only be modeled using a more complex Bayesian filter.

9.5 Experiments

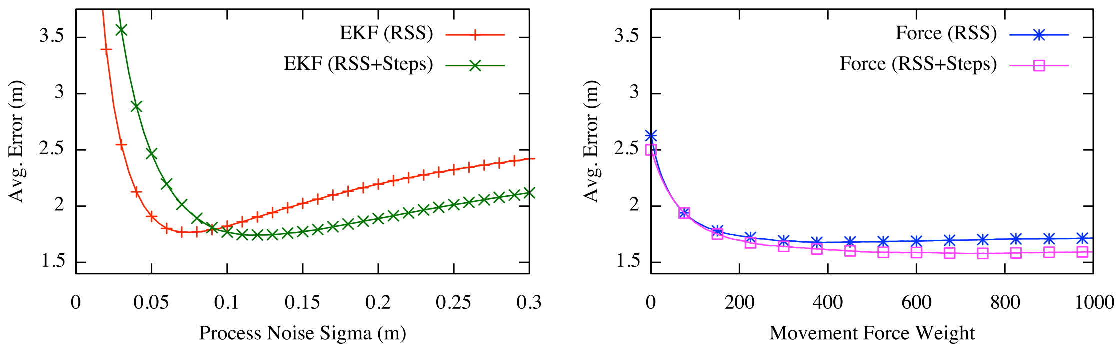
To evaluate the performance of the force-directed tracking approach, we present an experimental comparison with the described EKF implementation. The comparison is based on the sensor network experiments as described in Section 7.1. For both approaches, we analyze a variant that uses RSS measurements in combination with a movement model and another variant that combines the RSS measurements with step recognition data. The EKF that uses only RSS measurements for the localization is labeled "EKF (RSS)" in the following. The variant that additionally uses step information is labeled "EKF (RSS+Steps)". Similarly, "Force (RSS)" and "Force (RSS+Steps)" denote the variants of the force-directed approach that use only received signal strengths or both signal strengths and step data, respectively.

9.5.1 Parameter Influence

In the considered EKF model, movement is modeled by process noise. The higher the uncertainty of the process state, the stronger is the influence of measurement updates on the final position estimate. Figure 9.6a shows how the overall localization accuracy of our EKF implementations, averaged over all nine experiments, depends on the assumed standard deviation σ_w of process noise w_1 and w_2 (cf. Equation 9.3). We first analyze the extreme cases. If σ_w is chosen very low, measurement updates have little influence and the position estimate can not keep up with the movement of the tracked object. In this case, the localization error can become arbitrarily high. If σ_w is chosen too high, the prediction looses influence and the position estimate is mainly determined by the measurement update. This situation is similar to the one analyzed in Section 8.3.3 on the EKF measurement update.

According to Figure 9.6a, for the considered experiments the optimum values for σ_w are 7.5 cm for the model without step detection and 11.5 cm for the model with step detection. We will use this optimum parameters for the following analysis. However, one has to keep in mind that good values for σ_w strongly depend on factors such as walking velocity or the uncertainties of the measurements. Under realistic circumstances, these factors are not fully known, so one will usually not be able to use optimum parameters. Thus, from a practical point of view, determining the optimal parameters could be considered as an overfitting on the data. We use this approach nevertheless, as it is somewhat difficult to compare different approaches when suboptimal parameters are used. As the same set of parameters is used for all 9 experiments, the overfitting should be kept within reasonable limits.

In the force-directed approach, the behavior of the movement model is determined by two parameters: the expected movement distance $d_{\max}$ between two position estimations and the weighting factor w_{move} for the movement force $\vec{F}_{\text{move}}$ (we assume that the measurement forces are weighted with weight 1). Figure 9.6b shows the influence of w_{move} for the force-directed approach without step detection ($d_{\max} = 0.2\,\text{m}$) and with step detection ($d_{\max} = 0.5\,\text{m}$). Note that normally one would choose the values of $d_{\max}$ a little higher, e.g., $d_{\max} = 0.35\,\text{m}$ per 0.25 seconds or $d_{\max} = 0.65\,\text{m}$ per step. We chose the values



(a) Influence of σ_w in the EKF.

(b) Influence of w_{move} in the force-directed approach.

Figure 9.6: Influence of model parameters on the localization accuracy.

deliberately a little smaller to be able to demonstrate the influence of w_{move}. If w_{move} is chosen very low, the movement force plays almost no role for the final movement direction. In this case, we are back at the situation without movement forces, which was analyzed in the chapter on force-directed position estimation. For high values of w_{move}, a very strong movement force acts as soon as the position estimate leaves the area with radius d_{max} around the last position estimate. However, no matter how large w_{move} is chosen, the position estimate still can move distance d_{max} per step. This explains why the localization error in Figure 9.6b does not increase significantly for high values of w_{move}.

From a practical perspective, the modeling of the force-directed approach with a radius d_{max} within which no movement forces act offers some advantages. There are many scenarios for which it is easy to define a meaningful value for d_{max}. For example, a single step can not be arbitrary wide. In contrast, in the analyzed EKF model, the actual movement distance is determined by a combination of several complex factors, such as uncertainties of single measurements for example. This makes it harder to model such kinds of knowledge. Of course, it can happen that d_{max} is chosen significantly too small for the considered application. In this case, the outlined force-directed approach behaves similar to the EKF, meaning that the position estimate cannot catch up with the true position if w_{move} is chosen too large.

According to Figure 9.6b, the optimum values for w_{move} are $w_{\text{move}} = 385$ for the model without step recognition and $w_{\text{move}} = 710$ for the model with step recognition. As for the EKF, we will use this optimal parameters in the following studies.

9.5.2 Influence of Beacon Number

In this section, we examine the influence that the number of beacon nodes has on the localization results. To determine the subset of used beacons, we use the same approach as in the chapter on RSS-based position estimation: First, the two beacons with maximum distance from each other are selected. Subsequently, in each step the beacon that maximizes the minimum distance to any already selected beacon is chosen until the intended number of beacons is reached.

Figure 9.7 shows the average error over all experiments in dependence of the number of beacons. Again, in each step the 10 best RSS measurements were used. Altogether, the differences between the different approaches are rather small. In the considered experiments, the force-directed tracking approach that uses additional step information performs best, followed by the variant that uses only RSS measurements. Apparently, the EKFs with and without step detection show nearly the same average error. It seems that as long as the person walks somewhat steadily and the parameters are chosen accordingly, it makes no big difference whether the position update is done once per time step of 0.25 seconds or once every time when the person moves one step. Also, it can be seen that depending on the intended application even a small network with only 15 to 20 beacons might be sufficient to achieve a mean localization accuracy of about 2.5 m.

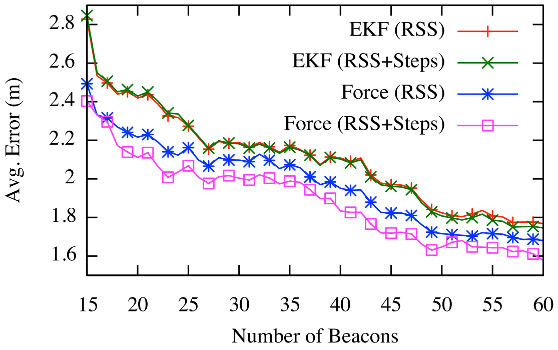


Figure 9.7: Average localization error in dependence on the number of used beacon nodes.

9.5.3 Tracking Examples

So far, all presented results were averaged over all experiments. To give a rough impression of how the algorithms performed in the single experiments, Table 9.1 gives an overview of the localization errors for each experimental run. The results were obtained using all 60 beacon nodes and the 10 best RSS measurements per time step. Additionally, the average localization errors of the force-directed position estimation technique from the last chapter are given. Apparently, using the additional movement forces between consecutive position nodes in the force-directed tracking approach lowered the error by more than one meter on average.

Algorithm / Scenario	ITI				ITIV				
	1	2	3	4	1	2	3	4	5
EKF (RSS)	1.69	1.54	1.44	1.32	1.67	2.11	1.83	2.25	2.06
EKF (RSS+Steps)	1.56	1.59	1.48	1.34	1.68	2.03	1.77	2.18	2.08
Force (RSS)	1.88	1.41	1.39	1.06	1.37	2.22	1.95	2.37	1.47
Force (RSS+Steps)	1.47	1.41	1.63	1.24	1.30	1.85	1.71	2.10	1.49
Force (No Model)	2.81	2.22	2.41	2.60	3.00	3.56	3.37	3.60	3.86

Table 9.1: Average localization errors (in m) for the single experiments.

To conclude the experimental part, two tracking results are presented in Figures 9.8 and 9.9. The path estimate in Fig. 9.8 was computed using the force-directed method without step detection. In comparison with Fig. 9.1, which is based on the same experimental run, we see that the movement model significantly improved the localization quality.

An example for a tracking result of the force-directed tracking method with step detection is shown in Figure 9.9. In both figures, we refrained from including the corresponding EKF trajectories, as the figures would become very confusing otherwise. Overall, the trajectories that are computed by the EKFs look very similar.

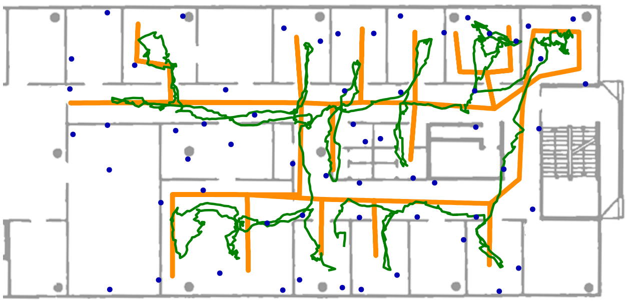


Figure 9.8: ITI: Tracking result using the force-directed approach without step recognition. The orange path shows the reference trajectory, the green path shows the computed trajectory.

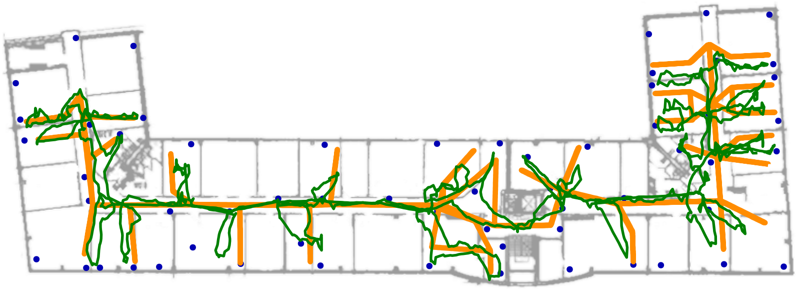
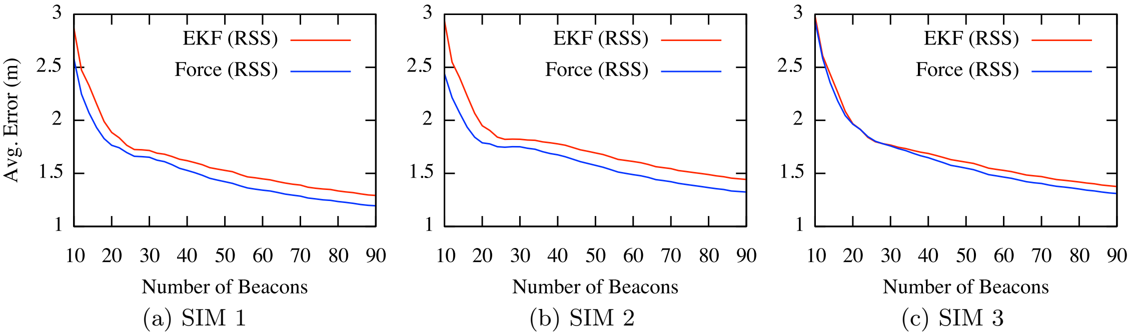


Figure 9.9: ITIV: Tracking result using the force-directed approach with step recognition. The orange path shows the reference trajectory, the green path shows the computed trajectory.

9.6 Simulations

In this section, we present some additional simulation results to supplement the results of the experiments. For the simulations, we used the simulation setup as described in Section 7.4. In the simulations, we restrict ourselves to tracking based on RSS measurements and movement model assumptions. To evaluate the approaches that additionally use step detection, a reasonable simulation of step data would be necessary, which is beyond the scope of this work. The simulations are based on the scenarios SIM 1 - SIM 3 as described in Section 7.4.2.

Figure 9.10 shows the dependence of the localization accuracy on the number of beacon nodes. For each walking scenario and each considered number of beacon nodes, 50 networks with different beacon node positions were used.



(a) SIM 1

(b) SIM 2

(c) SIM 3

Figure 9.10: Average localization errors in dependence of the number of beacon nodes for different simulation scenarios.

The overall picture in Figure 9.10 is very similar to the one that was observed in the real-world experiments, with slightly lower localization errors for the force-directed approach than for the EKF. On average, the mean localization error of both approaches is a little lower than in the real-world experiments. One reason for this difference is that in the experiments, due to hardware limitations, only 24 RSS values were recorded per time step. In the simulations, we omitted this restriction, as this limitation is very specific to the hardware used in the experiments.

Between the different simulation scenarios, SIM 1 resulted in slightly lower localization errors than the other simulation scenarios. The reason probably is that for SIM 1 less positions of the simulated trajectory lie close to the border of the sensor network. This is advantageous, as the localization usually is better when the mobile node is surrounded by beacon nodes. An example of a localization result with the force-directed tracking approach for simulation scenario SIM 2 with 50 beacon nodes is depicted in Figure 9.11.

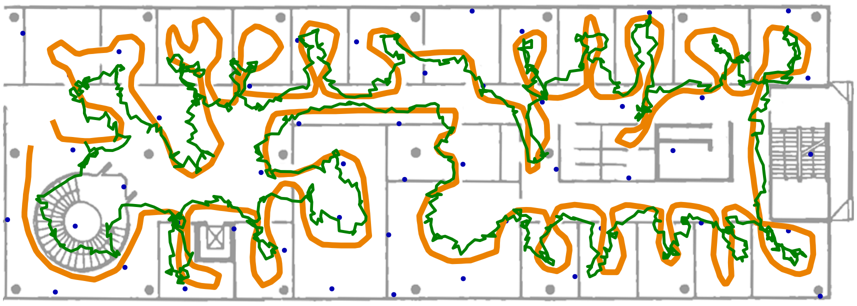


Figure 9.11: Localization result for simulation scenario SIM 2. A network of 50 beacon nodes was used (blue points). The orange path shows the reference trajectory, the green path shows the trajectory computed by the force-directed tracking method. In this simulation, the average localization error is about 1.5 meters.

9.7 Summary

In this chapter, we considered the application of force-directed approaches to the problem of RSS-based tracking in indoor environments. Different possibilities to model knowledge from movement models or inertial data have been described. The force-directed tracking approach was then compared to implementations of extended Kalman filters (EKFs). In our experiments and simulations, in which we analyzed the influence of parameters and beacon number, the force-directed tracking approach resulted in slightly lower localization errors than the EKFs. One possible cause for this difference is that, unlike EKFs, the force-directed approach does not require linearization.

Outlook. This work can only serve as a starting point, and further research is necessary to investigate the possibilities force-directed approaches offer in the context of tracking in wireless networks. In our opinion, the main benefit of the force-directed approach in comparison to other approaches is that it offers an immense freedom in building up system models. Process state and measurement inputs are not limited to normal distributions, and additional knowledge can easily be introduced with new kinds of forces. As runtime and space complexity of the outlined force-directed tracking algorithm depend linearly on the number of concurrent forces, it is well suited even for usage on hardware with restricted resources.

Chapter 10

Indirect Network Localization

In the previous chapters, we examined the use of sensor networks with known node positions to localize mobile devices based on signal strength measurements. In this chapter we consider the problem of localizing the sensor network itself. We present an indirect localization approach in which a mobile node is used to compute an embedding of the network, using solely information about signal strength measurements. In contrast to most existing work, we assume that neither information about the positions of anchor nodes is available, nor about the movement of the mobile node. The proposed approach puts extremely low requirements on the stationary nodes. It is not even necessary that they are able to communicate with each other. All computations are outsourced to one or few mobile devices.

The presented indirect localization approach is evaluated based on simulations and real-world experiments. Additionally, a comparison with two approaches that additionally use knowledge about the true movement of the mobile node is presented. As a possible application of the presented approach, the RSS-based tracking using the computed sensor node positions is examined.

10.1 Introduction

Most applications of wireless sensor networks require at least rough knowledge about the positions of the sensor nodes. Especially when the number of network nodes is high, a manual initialization of node positions is out of questions. Moreover, globally available positioning systems such as GPS only work outdoors. For this reason, many approaches for automatic localization of sensor networks have been proposed.

The existing localization approaches strongly differ with respect to the assumptions they make about the available knowledge. In some approaches, it is assumed that for some sensor nodes, so-called *anchor nodes*, the exact position or at least a position estimate is available. This information is then used to estimate the positions of the remaining nodes. Frequently, it is also assumed that angular information about the relative positions of nodes is available, e.g., using multiple antennas or antenna arrays.

A whole family of approaches uses communication between the stationary nodes to estimate pairwise distances between the nodes. Usually it is assumed that the networks are really large and the true distances are approximated by hop distances. However, these approaches work similarly well with RSS-based distance estimates. Due to the nature of communication-based distance estimation, one usually only obtains distances of nodes that are at most one maximum communication radius away from each other. The standard approach to infer the distances of nodes that are further apart is to use shortest path computations in the network that is induced by the available distance estimates. Finally, an embedding technique such as multidimensional scaling or a force-directed spring embedder is used to compute an embedding of the network. The most prominent such approach is MDS-MAP, which has been proposed by Shang *et al.* in [SRZF03].

In the area of indirect localization, some approaches assume that a mobile device is available whose position is known, e.g., by means of a GPS receiver. The mobile device either sends or receives wireless messages at known positions around the nodes that are to be located. Based on the received signal strengths in combination with the known positions, the positions of the stationary devices are determined.

Another kind of approach works by alternately localizing the mobile device and the stationary beacon devices. These so-called *SLAM* approaches (simultaneous localization and mapping) usually assume that either some initial information about reference positions is available, or information about the movement of the mobile device, e.g., in terms of odometry data.

In this work, we follow an approach similar to MDS-MAP. But instead of using communication between stationary nodes to estimate inter-node distances, we use a mobile node, e.g., a person that walks through the network with a wireless receiver. The mobile device collects wireless messages that are broadcasted by the stationary sensor nodes, and the task is to localize the sensor network based on the corresponding received signal strengths. In the considered scenario, we assume that there exists no information about any reference positions or the movement of the mobile device.

Contributions. Indirect localization of sensor networks has already been studied in several variants. However, usually either knowledge about the movement of the mobile device is assumed [SR04b, XCZ08], or the distance estimation method is significantly better, e.g., using time difference of arrival (TDoA) or other accurate techniques [PBDT05, ZL11]. In this work, we focus on additional difficulties that arise in connection with RSS-based distance estimation in indoor environments. The considered setting thus differs in several ways from most existing work on MDS-based localization: Our simulations are adjusted to measurements from real-world experiments and they also include systematic errors such as the attenuation of signals at walls. Additionally, we consider effects that are caused by the limited sensitivity of wireless receivers, e.g., distances to nodes that are far away are systematically underestimated as only those signals that randomly exceed the receiver sensitivity are detected. We describe the resulting additional difficulties and present simple but efficient possibilities to cope with them.

To evaluate the presented indirect localization approach, we use extensive simulations. A comparison with two approaches for indirect localization is presented, which additionally use full knowledge about the movement trajectory of the mobile device. As an example application, the tracking based on the computed sensor node positions is studied, and achievable tracking accuracies are examined based on simulations. Finally, some real-world examples are presented and additional problems that can arise in real-world application are discussed.

Applications. The presented approach can be used in all applications that require rough knowledge about the positions of the sensor nodes. As one possible example, we consider in this work the ad-hoc deployment of a sensor network for the subsequent localization of mobile devices. The presented indirect localization approach makes it possible to use a heterogeneous network with many low-cost stationary nodes and only few powerful mobile nodes. The stationary nodes only have to send messages with their unique node ID. Thus, they can be produced very small and cheap. For some scenarios, it might even be possible to use disposable nodes. Only the mobile nodes, which are small in number, have to process messages and to compute an embedding of the network. As they are carried on body, they do not get lost during operation. Thus, they can easily be reused. Additionally, the necessary computations are not computationally expensive, so average modern cell phones should be completely sufficient.

In the considered scenario, the stationary nodes are distributed throughout the building or area of interest. After deployment, they regularly send messages with their unique ID. These messages are received by mobile devices, which are carried by people. The measured signal strengths are then used to infer information about distances between stationary nodes. Given the calculated distances, a standard approach such as multidimensional scaling is used to find an embedding of the network. This is done centralized at the mobile nodes.

Usually, centralized approaches are undesirable in sensor networks, as they require a lot of communication and, thus, they normally do not scale well with network size. In the considered scenario, the situation is different. The RSS data is collected directly in the mobile nodes and can also be processed there. It is sufficient when the mobile devices exchange the computed distances from time to time. If the mobile devices are conventional cell phones, this could be easily realized by connecting to some internet service. Alternatively, one could also use the stationary network nodes to pass the messages. For realistic network dimensions, the amount of data that has to be exchanged should be quite manageable. Either way, using a heterogeneous network where all the processing is done by few mobile nodes should make it possible to use larger stationary networks with lower costs.

10.1.1 Related Work

Existing approaches for the sensor network localization problem can be classified according to the kind of information that is used, ranging from solely connectivity information to knowledge about node distances and positions of anchor nodes. In the following, we give a brief overview on the different possibilities for locating a sensor network.

Anchor-based localization. If anchor nodes or reference points with known positions are available, one can use distance or angular information to infer the positions of nodes with unknown locations by using trilateration or triangulation [NL02, SRL02]. Another example for this approach is the distributed ad hoc positioning system (APS) proposed by Niculescu and Nath in [NN01]. It either uses hop distances (DV-hop) or RSS-based distance estimates (DV-distance) to infer distances to known anchor nodes. Using these distance estimations, a node can compute its own position using a trilateration method similar to the one used in the Global Positioning System (GPS). The method proposed by Savarese *et al.* [SRL02] consists of two steps. The first step, Hop-TERRAIN, is similar to DV-hop and computes initial position estimates. The second step is an iterative refinement where each node uses least-squares trilateration to update its position. Like DV-hop, the method needs to know positions of at least three anchor nodes. In [DPEG01], Doherty *et al.* use a convex constraint satisfaction approach to formulate the problem as a feasibility problem. The convex constraint problem is then solved by semi-definite programming. If directional information is available, it is even possible to use linear programming. This method requires the availability of anchor nodes with known positions, preferably on the outer boundary of the network. Given a high number of anchor nodes, the collaborative multilateration approach proposed by Savvides *et al.* in [SPS02] can be used. Nodes that are connected to at least three anchor nodes determine their position and in turn become new anchor nodes. This is iterated until every node has a position estimate. In the RF-based localization system of Bulusu *et al.* [BHE00], a grid of beacon nodes with known positions is used. A node localizes itself to the centroid of a set of reference points that are within communication distance. Again, this method needs a high number of beacons to achieve reasonable accuracy.

Anchor-free localization. A method that works without anchor nodes and orientation information is MDS-MAP [SRZF03] (cf. Section 2.2.3). In MDS-MAP, an all-pairs-shortest-path algorithm is used to roughly estimate distances between each possible pair of nodes. Subsequently, classical multidimensional scaling (MDS) [Tor65] is used to derive node locations that fit the estimated distances. Optionally, if anchor nodes with known positions are available, this information is used to normalize the resulting coordinates. As MDS-MAP has difficulties to deal with irregular node placements, several modifications and extensions have been proposed. MDS-MAP(P) [SR04a] is a distributed version of MDS-MAP which divides the network into overlapping subnetworks. For these subnetworks, embeddings are computed using MDS. Subsequently, the single embeddings are stitched together by aligning them as good as possible. An optional least-squares refinement is used to improve the embedding. Similar approaches have been proposed in [YW08, SDK08], where the network is separated into small clusters which are embedded using MDS and finally merged. A slightly different approach has been proposed in [MSZ+06], where clustering is used to compute a subset of nodes that are embedded using MDS. The remaining nodes are then localized based on the computed positions using some cheaper technique such as trilateration.

In [CPI06], a distributed weighted-MDS method (dwMDS) was introduced by Costa *et al.*, which weights single measurements according to the expected accuracy.

In the approach by Basu *et al.* [BGMS06], distance and compass information are used to compute an embedding. In [KW07], Katz and Wagner considered the anchor-free localization of sensor networks based on noisy measurements of distances and relative directions. They showed that the problem to recover the nodes' positions is NP-hard in this setting, and presented a distributed algorithm that aims at robustness to erroneous measurements and scalability. Another work that deals with direction-based localization was presented by Katz *et al.* in [KGW07].

In [PBDT03], Priyantha *et al.* proposed AFL, a distributed, anchor-free localization approach that is based on force-directed methods.

In [EEF+06, EFI+10], Efrat *et al.* examined force-directed approaches for centralized sensor network localization. For the case that angular measurements are available, they proposed a force-directed approach that uses ideas from dead-reckoning to achieve improved estimates for the relative positions between distant sensor nodes. A distributed version of this multi-scale dead-reckoning (MSDR) algorithm was presented by Coogan *et al.* in [CKKK10].

The indirect localization of RFID tags has been considered by Shi and Wong in [SW11]. Some RFID readers are deployed around the area of interest and distances between RFID tags are estimated indirectly based on signal strength measurements. An embedding of the RFID tags is then found based on MDS-MAP. Optionally, an additional refinement can be used to improve the localization results.

Indirect localization with mobile beacons. Most approaches that use a mobile beacon node for the localization of sensor networks assume that the position of the mobile beacon node is known, e.g., by means of a GPS receiver or an inertial measurement unit (IMU). This information can then be used during deployment to initialize the positions of the sensor nodes. Such an approach is for example used in [GGK+03, SHS04, SVG+10].

Another possibility is that the mobile beacon node broadcasts its known position in regular intervals. These broadcasts are then received by the stationary sensor nodes and the position is determined using trilateration or similar techniques. Examples for this kind of approach are [SR04b, XCZ08].

In [HE04], Hu *et al.* introduced the sequential Monte Carlo Localization method, a particle filter approach that localizes a heterogeneous network of static and mobile nodes.

In their approach, it is assumed that for some of the nodes, so-called *seeds*, the positions are known. Both static nodes or mobile nodes can be seed nodes.

The results of [HE04] were generalized in [RD07] by Rudafshani and Datta. They also considered a network with mobile nodes, where some of the nodes know their positions, e.g., by means of GPS. In contrast to [HE04], they assume that even non-seed nodes broadcast their position estimates and that these estimates are used by other nodes to locate themselves.

If information about the relative movement of the mobile device is known, this information can also be used for localization. An example is [RB00], where a group of robots shares odometric data to locate themselves. Approaches in the context of simultaneous localization and mapping (SLAM), where both the position of a mobile device and the positions of reference points are estimated at the same time, frequently also utilize odometry data. An example is [HMS01], where one or multiple mobile robots are localized simultaneously with a sensor network.

There also exist some approaches that work without knowledge about the location and the relative movement of the mobile device. In [PBDT05], a mobile device is used to measure distances between node pairs until the resulting distance constraints form a globally rigid structure that guarantees a unique localization. The approach is based on the Cricket location-support system [PCB00], which estimates distances based on time differences of arrival (TDoA) between RF signals and ultrasound pulses. This TDoA approach allows distance estimation accuracies in the range of few centimeters. Thanks to the high accuracy of the distance estimations, only few measurements are necessary to get rigid structures. In [ZL11], sensor nodes are deployed manually. Every time a sensor node is placed, a mobile beacon, which sends RF signals and ultrasonic pulses, is switched on for a short period. By measuring TDoA between the different kinds of signals, distances between the mobile node and nodes that are already deployed are estimated. These distance estimates are then used as distance estimates between the newly deployed node and the nodes that have been deployed earlier. The embedding of the sensor network is finally computed using MDS-MAP.

Further information on localization of wireless sensor networks can be found in several surveys on this topic, e.g., [MFA07, WGD10].

10.2 Indirect Localization

In the standard MDS-MAP approach, the embedding of the sensor network is found as follows: Sensor nodes which are within communication range of each other exchange wireless messages. The distances between nodes that can communicate directly are then either estimated based on the received signal strengths, or they are simply set to 1 if hop distances are used. Distance estimates for pairs of nodes that are further away from each other are determined using shortest path computations in the network that is induced by the available distance estimates. Finally, the pairwise distances are used to compute an embedding of the network using multidimensional scaling (MDS). In some variants of MDS-MAP, an additional refinement is used to improve the embedding.

In this section, we introduce a modified version of MDS-MAP that enables the indirect localization of wireless sensor networks. To indicate that the embedding is computed indirectly, we will label the approach MDS-IND. If an additional force-directed refinement is used after the embedding with MDS, the resulting approach will be labeled MDS-IND(R).

10.2.1 Indirect Distance Estimation

We considered three strategies to indirectly estimate distances between static sensor nodes. The first variant uses a two-way measurement: Each time when the mobile node m receives RSS-measurements to more than one static sensor node, for each pair (u, v) of sensor nodes the RSS-based estimates $\hat{d}_{m,u}$ and $\hat{d}_{m,v}$ of the distances between nodes m and u and nodes m and v are used to get an upper bound $\hat{d}_{m,u} + \hat{d}_{m,v} \geq \hat{d}_{u,v}$ on the estimated distance $\hat{d}_{u,v}$ between u and w (cf. Figure 10.1). The idea then was to use the smallest upper bound as distance estimate, maybe increased by some percent to account for uncertainties in the RSS-based distance estimation process.

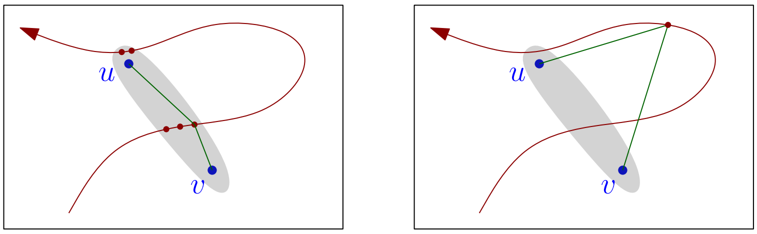


Figure 10.1: Two-way distance estimation. The blue points correspond to static sensor nodes u and w with unknown positions, the red line shows the (unknown) trajectory of the mobile node m, and the red points mark positions where the mobile node takes RSS measurements. *Left:* For measurements in the gray area, the average distance estimation approximates the true distance reasonably well. *Right:* Example for a measurement position that results in strongly overestimated distances.

Unfortunately, it turned out that this kind of distance estimation is extremely unreliable. One reason is that if, for two static nodes u and v, the mobile node does not get close to either u or v, or alternatively to the straight line connecting u and v (this area is highlighted in Figure 10.1 in gray color), then the distance between u and v is significantly overestimated, and there is no simple way to detect whether this happens. Second, even when using the lower quartile of the "upper bounds" instead of the minimum, large distances are still significantly underestimated due to the high fluctuation of signal strengths, whereas short distances are already significantly overestimated. For this reason, we will not go further into the details of this approach.

The second strategy that we considered was the approach for indirect distance estimation proposed in [SW11] for the indirect localization of RFID tags. Although the work in [SW11] considers several static RFID readers instead of a mobile device, the underlying problem of indirect distance estimation is quite similar. One can simply consider the positions of the different RFID readers as different positions where a single mobile device takes RSS measurements. The approach works as follows: Using triangular inequalities, each pair of distance estimates $\hat{d}_{m,u}$ and $\hat{d}_{m,v}$ from mobile node m to two static nodes u and v is used to get an upper bound and a lower bound on the distance $d_{u,v}$ between u and v, i.e.,

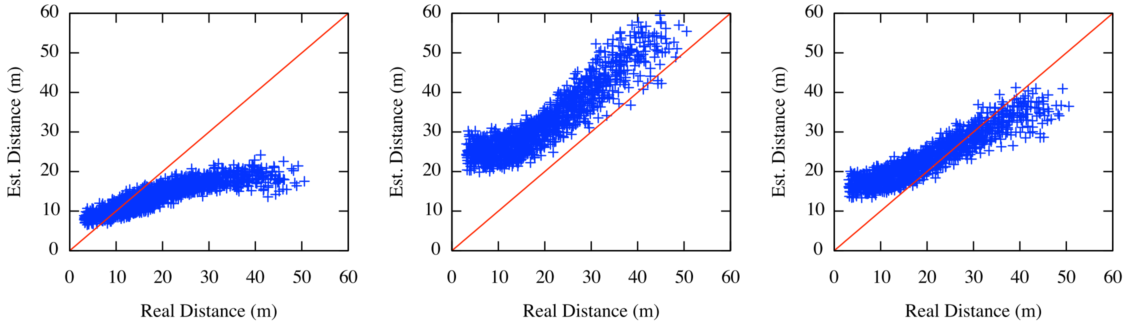
$$|\hat{d}_{m,u} - \hat{d}_{m,v}| \leq d_{u,v} \leq \hat{d}_{m,u} + \hat{d}_{m,v} \quad . \tag{10.1}$$

Then, to account for uncertainties of error-prone distance estimates, the final upper bound $d^u_{u,v}$ and lower bound $d^l_{u,v}$ on the distance $d_{u,v}$ are computed by averaging over all computed upper and lower bounds. The distance $d_{u,v}$ is then estimated as the mean of the upper bound $d^u_{u,v}$ and the lower bound $d^l_{u,v}$, i.e.,

$$\hat{d}_{u,v} = (d^u_{u,v} + d^l_{u,v})/2 \quad . \tag{10.2}$$

We implemented this strategy, but it turned out that the high uncertainty of the RSS-based distance estimation resulted in a severe overestimation of short distances. The reason is that even if two nodes u and v are close, there might be some measurements where one of the distance estimates $\hat{d}_{m,u}$ and $\hat{d}_{m,v}$ is much larger than the other one. In such a case, $|\hat{d}_{m,u} - \hat{d}_{m,v}|$ can exceed $d_{u,v}$ significantly. As a result, for small distances $d_{u,v}$ the computed lower bounds $d^l_{u,v}$ often exceeded $d_{u,v}$ in our simulations.

Figure 10.2 shows for a single simulation run how the computed upper and lower bounds $d^u_{u,v}$ and $d^l_{u,v}$ on the distance and the final distance estimate $\hat{d}_{u,v}$ correlated with the true distances $d_{u,v}$ between the corresponding sensor nodes. We observe that both upper and lower bounds result in a significant overestimation of short distances. In consequence, the final distance estimate also is far too large for short distances.



(a) Lower bounds $d^l_{u,v}$. (b) Upper bounds $d^u_{u,v}$. (c) Distance estimates $\hat{d}_{u,v}$.

Figure 10.2: Indirect estimation of pairwise distances using the method proposed in [SW11]. The plots are based on a single run of simulation scenario SIM 2 with 50 random sensor nodes. (a) Each data point shows for a single pair of static nodes how the lower bound on the distance correlates to the true distance. (b) Correlation between upper bounds and true distances. (c) Correlation between estimated distances and true distances.

The third and last strategy that we considered utilizes the fact that one can relatively accurately detect whether the mobile node m is close to some static sensor node u, by simply checking whether the received signal strength exceeds some specific *proximity threshold* θ_{RSS}. If this happens, the distance between u and some other node v can be approximated by the distance between m and v, i.e., $d_{u,v} \approx d_{m,v}$. Thus, to estimate the distance between two nodes u and v, the mobile node estimates its distance to u or w when it assumes to be close to the respective other node. All such distance estimates are collected, and the final estimate for the distance between u and v is computed as the mean or median of all distance estimates. This approach, which is illustrated in Figure 10.3, has the nice property that it is about equally likely that the distance between a pair $\{u, v\}$ of nodes is overestimated or underestimated. Thus, one can expect that the mean or median of all distance estimates approximates the true distance reasonably well. In the experiments and simulations of this thesis, we use the median of the estimates. Furthermore, to achieve additional robustness, we required a minimum of 20 estimates per pair of nodes, otherwise the distance estimate was ignored.

Apparently, if m gets neither close to u nor to w, the described one-way distance estimation will give no estimate for the distance between u and w. Thus, one has to make sure that most sensor nodes are visited (i.e., approached up to a distance of few meters) at some point by a mobile node. However, this restriction also has its positive aspects. While in the two-way distance estimation approach it was not easily possible to check whether the sum of the distances from m to u and from m to w gets close to the distance between

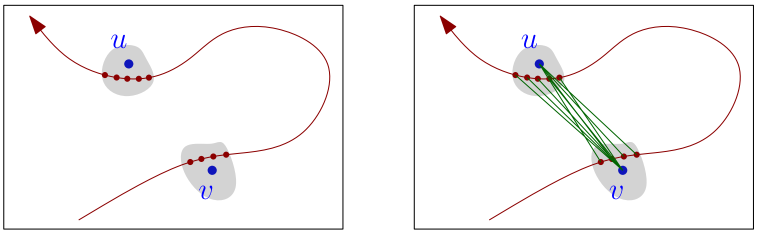


Figure 10.3: One-way distance estimation. The blue points correspond to static sensor nodes u and v with unknown positions, the red line shows the (unknown) trajectory of the mobile node m, the red points mark positions where the mobile node takes RSS measurements. At positions where m receives very strong signals from one of the nodes (gray areas), it estimates its distance to all other nodes which are in communication range based on received signal strengths. The green lines symbolize single distance estimates. The final distance estimate between u and v is computed as the average of all distance estimates from m to nodes u or v.

u and w, the selection rule of the one-way distance estimation makes sure that only those measurements are used for which one can expect that the distance between m and u or between m and w is comparable to the distance between u and w.

One question that remains open at this moment is how the threshold θ_{RSS} on the signal strength, which indicates proximity to a sensor node, should be chosen. A good choice for the value of θ_{RSS} depends on the used hardware and the trajectory of the mobile node, i.e., how closely the mobile node approaches the static sensor nodes. In the scenario of indoor localization, as considered in this thesis, a value of θ_{RSS} that corresponds to an estimated distance of 2 m to 5 m seems to work well. A simulation-based study on how the localization accuracy depends on the value of θ_{RSS} and on the walked trajectory is presented in Section 10.5.1. Some examinations of distance estimations based on the one-way approach will be shown later for simulation scenario SIM 2 in Figure 10.4a, for a real-world experiment at the ITI in Figure 10.12a, and for an experiment at the ITIV in Figure 10.14a.

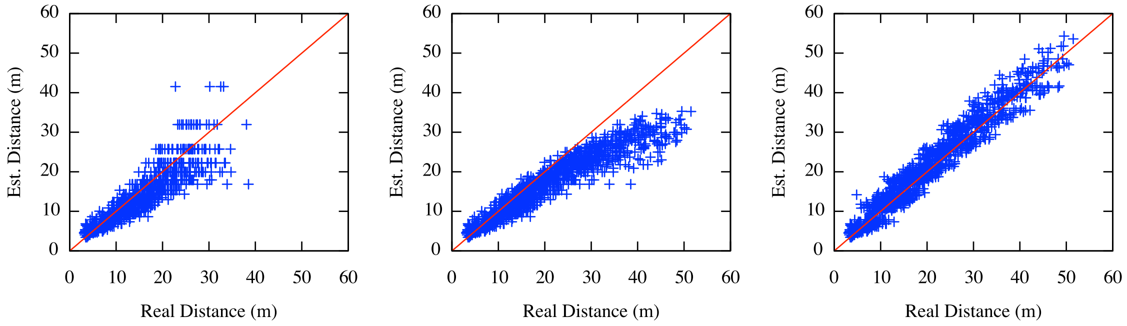
10.2.2 Completing Pairwise Distances

After the one-way distance estimation, only distance estimates between nodes which are not much more than one communication distance away from each other are available. Similar to the original MDS-MAP approach, we use shortest path computations in the graph that is induced by the available distance estimates to approximate distances of nodes which are further away from each other. However, the characteristics of RSS-based distance estimation make some slight modifications necessary.

Figure 10.4a shows for a simulation based on scenario SIM 2 with 50 sensor nodes how the estimated distances correlate to the real distances. Each data point belongs to a single pair of static sensor nodes. We observe that only for few nodes that are further than 30 m away from each other a distance estimate could be established.

If all distance estimates of the one-way distance estimation are used in the shortest path computation, one gets the situation depicted in Figure 10.4b. Wo observe that all long-range distances are significantly underestimated. This behavior is caused by the high fluctuation of signal strengths, which leads to the effect that the distances between single pairs of nodes are significantly underestimated. In combination with shortest path computations, a single underestimated distance can cause that most long-range distances are underestimated.

To alleviate this problem, we introduce a second threshold, *distance threshold* θ_{dist}. All



(a) One-way distance estimates. (b) Compl. distances, no cutoff. (c) Compl. distances, $\theta_{\text{dist}} = 8\,m$.

Figure 10.4: Estimation of pairwise distances between static nodes. Each data point shows for a single pair of static nodes how the estimated distance correlates to the true distance. The plots are based on a single run of simulation scenario SIM 2 with 50 random sensor nodes. (a) Situation after the initial one-way distance estimation. (b) Situation after shortest path distance completion without threshold. (c) Situation after shortest path distance completion with $\theta_{\text{dist}} = 8\,m$.

distance estimates of the one-way distance estimation that exceed the cutoff value θ_{dist} are not used in the shortest path computation. The motivation behind this approach is that the distance estimation error increases significantly for larger sender-receiver distances (cf. Section 7.2). Thus, considering only short distances in the shortest path computations promises to reduce the error. Again, there is a tradeoff. If θ_{dist} is chosen too small, the network that is induced by the remaining one-way distance estimates might fall apart into unconnected components. In our experiments and simulations, values of θ_{dist} in the range from 7 m to 10 m resulted in reasonable results. Figure 10.4c shows the effect of using only distance estimates up to $\theta_{\text{dist}} = 8$ m in the shortest path computations. We observe that the systematic underestimation of long distances is successfully avoided.

To convey an impression of the knowledge about pairwise distances before the shortest path computations, Figure 10.5 presents an example for which pairs of nodes the one-way distance estimation established an estimate. Distance estimates which exceed $\theta_{\text{dist}} = 8$ m are filtered out, as they are not used in the shortest path computation. The color of the edges indicates whether the corresponding distance estimate overestimates (red) or underestimates (green) the true distance.

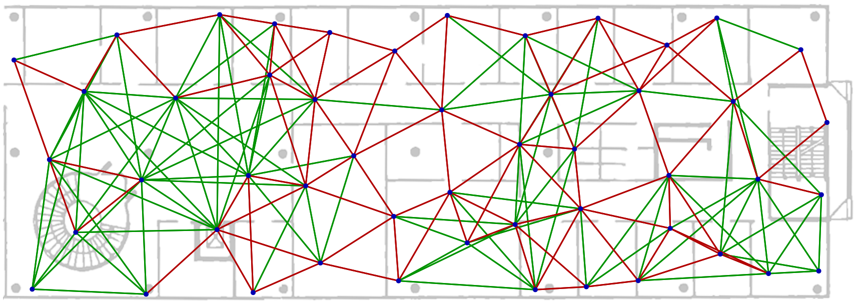


Figure 10.5: Pairwise distance estimates for scenario SIM 2 with 50 sensor nodes. Pairs of static nodes for which the distance estimate is less than $\theta_{\text{dist}} = 8$ m are connected by an edge. Green edges indicate that the actual distance is underestimated, red edges indicate that the distance is overestimated.

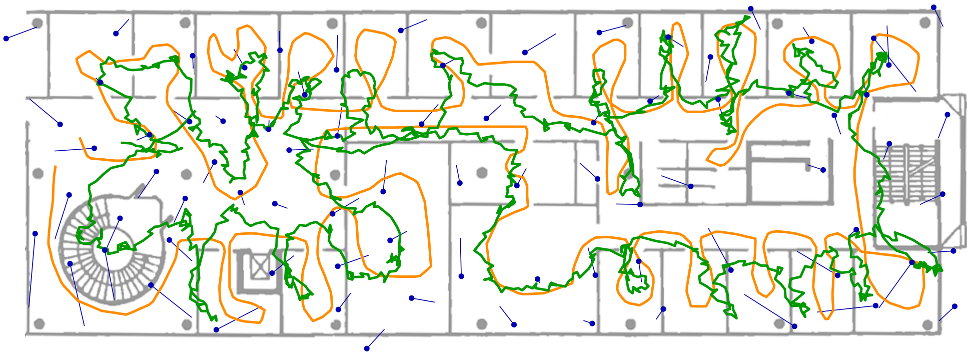


Figure 10.6: Example for an embedding of simulation scenario SIM 2 using MDS-IND(R). The simulated sensor network consists of 80 sensor nodes. Blue points show the computed sensor node positions. Blue lines emerging from the computed sensor node positions point to the actual positions of the sensor nodes. The orange line shows the simulated trajectory of the mobile node. The green line shows an estimate of the walked trajectory, using the force-directed tracking approach in combination with the estimated beacon node positions.

10.2.3 Network Embedding and Refinement

After the shortest path computation, pairwise distance estimates are available for all pairs of static sensor nodes. Using this information, an arbitrary embedding approach can be used to find an embedding of the network. In this work, we compute the embedding using multidimensional scaling (MDS) in combination with a subsequent force-directed refinement. Most variants of MDS, e.g., multilevel variants or the well-known MDS-MAP(P) [SR04a] approach, aim at solving problems that occur in very large networks, with hundreds of nodes, or in networks that span several communication radii. However, the networks that are considered in this work usually consist of less than a hundred nodes and they also cover a relatively small area in terms of maximum communication distances. Thus, we use in this work classical MDS [Tor65]. For the refinement, we use a Fruchterman-Reingold spring embedder (cf. Section 2.2.2).

An example of a sensor network embedding that was computed using the described MDS-IND approach is presented in Figure 10.6. To align the computed node positions with the real node positions, an optimal linear transformation that minimizes the sum of the squared errors between the true positions and the estimated positions was used. For the transformation, only translation and rotation were used. In particular, the scale of the estimated embedding was not changed.

Note that the described MDS-IND approach does not assume that a map of the building is available. In the considered scenario, one only gets positions of sensor nodes relative to each other. The alignment of the computed node positions to the map of the building was done afterwards, with the intention to convey an impression of the localization accuracy that is achievable with MDS-IND.

10.2.4 Using Multiple Mobile Nodes

The presented approach can be easily extended to the use of multiple mobile nodes, which cooperatively collect RSS data that is used later to compute an embedding of the network. The input of MDS-IND consists solely of sets of RSS measurements which were taken more or less simultaneously, thus data sets that are collected by different nodes simply have to be transferred to the node or device that computes the embedding of the network. If the devices that collect the data use different hardware, an additional adjustment might be necessary to make the RSS measurements comparable.

10.3 Application: Tracking

The presented MDS-IND approach can be used in connection with any application that needs position estimates of the sensor network. In this work, we consider as one of many examples the tracking of mobile devices based on signal strength information. For the tracking, we use the force-directed tracking approach that was presented in Chapter 9. But instead of the true positions of the sensor nodes, the position estimates of MDS-IND or MDS-IND(R) are used to reconstruct the walked trajectory.

Figure 10.7 shows a simulation output of MDS-IND(R) followed by force-directed tracking using the computed node positions. The reconstructed trajectory provides a reasonable impression of the real trajectory that was used in the simulation (cf. Figure 10.6). Note that in general the computed embedding and, thus, the computed trajectory can be arbitrarily rotated or mirror-inverted. If additionally positions of some anchor nodes are known, the computed trajectory can be easily displayed on a map, similar to Figure 10.6. Alternatively, compass data could be used to adjust the orientation of the computed node positions and the estimated trajectory.

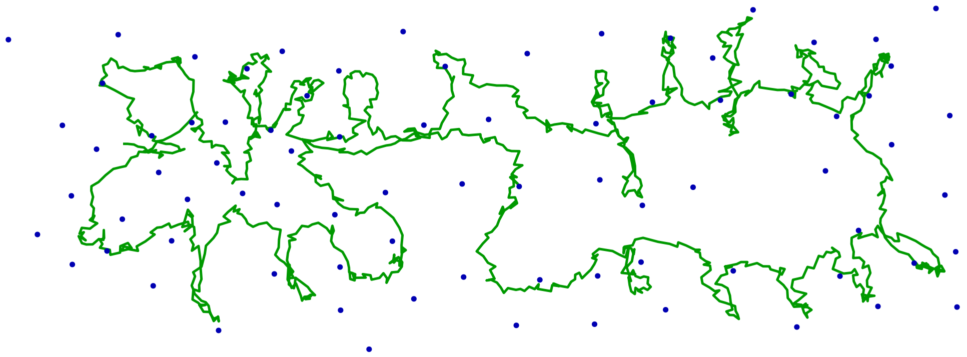


Figure 10.7: Output after an embedding of the network using MDS-IND(R) and subsequent tracking with the force-directed tracking approach using the estimated node positions. The blue points show the estimated node positions, the green line shows the estimated walking trajectory. This figure is based on the same data as Figure 10.6, but all information that is not part of the output of the algorithms is omitted.

10.4 Reference Approaches

As a reference approach for indirect network localization, we use the probabilistic indirect localization approach that was presented in [SR04b]. In the considered algorithm, a mobile beacon node with known position repeatedly broadcasts its current position. The broadcasts are received by the static nodes, which use the broadcasted position information in connection with the received signal strength of the message to estimate their own positions.

The scenario of [SR04b] differs in two ways from the scenario considered in this work: First and most important, in [SR04b] the position of the mobile node is known by means of GPS or similar techniques. This offers a significant advantage compared to the situation considered in this work, where no knowledge about positions and relative movement of the mobile node is available. Second, in [SR04b] the mobile node sends the broadcasts which are then received by the static nodes, whereas in this work the static nodes broadcast their IDs and the mobile node measures the received signal strengths. This second distinction should make no real difference, as on average one can expect similar received signal strengths in both cases. Altogether, it seems natural that additional knowledge about the trajectory of the mobile node allows for much better embeddings of the sensor network. In our simulations, we examine how big this difference is.

The probabilistic approach of [SR04b] is computationally very expensive. For each static sensor node, a map of probabilities has to be stored for a relatively large area. To examine whether similar results can be achieved with less effort, we adjust the force-directed RSS-based position estimation technique of Section 8.2 to the problem of indirect localization using a mobile node with known trajectory. The necessary modification is straightforward: Each broadcasted message of the mobile node is simply considered to be coming from a different static beacon node whose position corresponds to the position of the mobile node at the time the broadcast was sent.

It would also be very interesting to compare the presented indirect MDS-IND(R) approach to the traditional MDS-MAP approach that uses RSS measurements between static sensor nodes to find an embedding of the network. It is not directly clear, which of both approaches results in more accurate embeddings. The original MDS-MAP approach has the advantage that the RSS measurements are taken directly by the nodes whose distance is to be determined, whereas the MDS-IND approach uses indirect measurements, which potentially introduce additional kinds of errors. On the other hand, received signal strengths strongly depend on the exact position of the measurement and on the orientations of senders and receivers. For the static sensor nodes, both position and orientation do not change, thus, it is possible that the fixed positions and orientations of the nodes results in extraordinary high or low signal strengths. On the contrary, the mobile node takes measurements at many different positions, so there might be a chance that on average the measured signal strengths resemble the signal strength that is expected based on the distance between the two static nodes. Unfortunately, in our experiments it was not possible to collect the necessary data, as only the mobile node was able to store received signal strengths. Additionally, it is also not directly clear how much fluctuation of the RSS values should be used in the simulations to simulate communication between static sensor nodes. For this reasons, the described comparison will only be subject of future research.

10.5 Simulations

For the simulation-based evaluation of the proposed MDS-IND approach, we use the simulation scenarios SIM 1 to SIM 3 as described in Section 7.4.2. However, departing from the previous chapters on RSS-based position estimation and tracking, we now assume that the positions of the static beacon nodes are completely unknown.

Quality Measurement. Several quality measures have been used in the literature to evaluate the quality of positioning algorithms. For anchor-free localization approaches, the localization result can be rotated or flipped. To deal with this, often quality measures are used that measure the *stress* of the solutions, i.e., the deviation of the distances $\hat{d}_{ij}$ between nodes in the computed embedding from their true distances d_{ij}. Examples for such quality measures are the *global stress root-mean-square error* (GSR), which has been used in [KW07],

$$\mathrm{GSR} := \sqrt{\frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \left(\frac{d_{ij} - \hat{d}_{ij}}{d_{ij}} \right)^2} \quad , \tag{10.3}$$

and the *Frobenius metric* (FROB) [GVL96],

$$\mathrm{FROB} := \sqrt{\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(d_{ij} - \hat{d}_{ij} \right)^2} \quad , \tag{10.4}$$

which has been used in [EEF$^+$06, KW07, EFI$^+$10]. In both definitions, n states the number of static sensor nodes.

The problem with these measures, as well as with similar measures that are also based on distances between sensor nodes, is that they are hard to interpret and they sometimes hide systematic errors such as different scales [Kat09]. For this reason, we use an approach similar to the one used in [Kat09] to measure the quality of computed embeddings: For each computed embedding, a best-fit affine transformation [TCPK01] using rotation, translation, and reflection is used to align the estimated positions with the true positions. Subsequently, the quality of the embedding is determined as the average distance between the computed positions and the true positions. This measure is analogous to the one used in this work for the RSS-based position estimation and tracking, and it resembles the visual impression very well.

10.5.1 Proximity Threshold θ_{RSS}

We first analyze how the node localization quality depends on the value of θ_{RSS}, the threshold that controls for which signal strengths it is assumed that the mobile node is close to some static node. Figure 10.8 shows for various values of θ_{RSS} the resulting *average node localization error*, i.e., the average deviation between the true node position and the position estimate computed by MDS-IND(R). Each data point represents an average of 50 runs, using different random placements of 50 sensor nodes. For all embeddings, $\theta_{\mathrm{dist}} = 8\,\mathrm{m}$ was used.

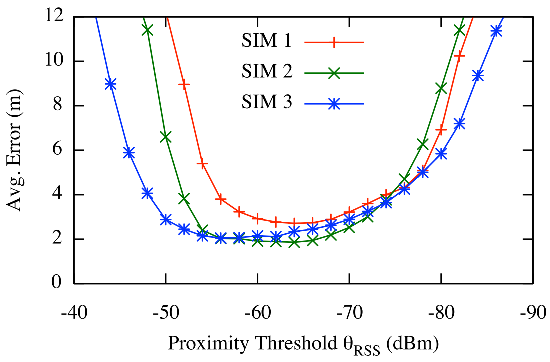


Figure 10.8: Average node localization error of MDS-IND(R) in dependence of the used proximity threshold θ_{RSS}.

The U-shape of the functions, which is apparent for all simulation scenarios, clearly shows that both very small and very large values of θ_{RSS} result in bad localization results. If θ_{RSS} is chosen too large, i.e., if a sensor node is only assumed to be nearby when the received signal strength is very strong, there are many pairs of nodes for which no distance estimate can be established. On the other hand, if θ_{RSS} is chosen too small, even distance estimates that are taken rather far from a sensor node are included in the distance estimations of that node.

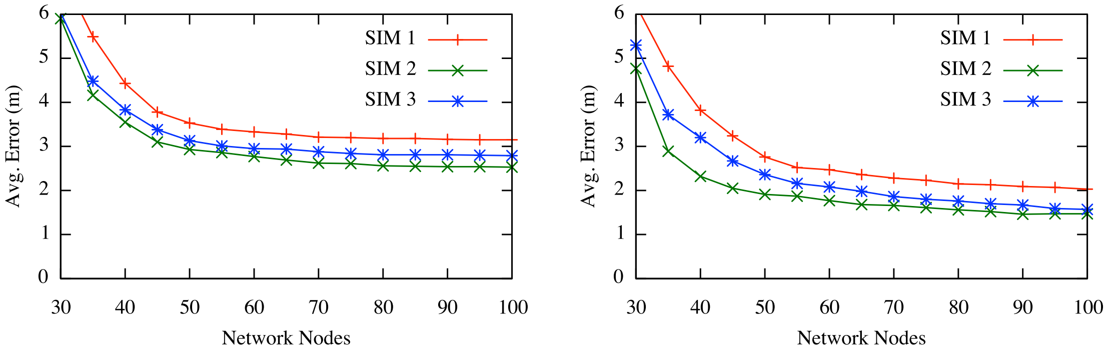
Considering the differences between the simulation scenarios, it turns out that for scenario SIM 1, the scenario with the short trajectory that only covers parts of the floor, the value of θ_{RSS} that results in the best localization results is lower than for the other scenarios. This seems reasonable, as in scenario SIM 1 the mobile node does on average not approach the sensor nodes as closely as in the other scenarios. Thus, even measurements at larger distances from a node have to be used in the distance estimations. It is also apparent in Figure 10.8 that the longer trajectories of SIM 2 and SIM 3 result in broader intervals of good values for θ_{RSS}, i.e., values that result in good localization results. Again, the reason is that lower values of θ_{RSS} can be used if the static nodes are approached more closely.

For the trajectories of the simulation scenarios SIM 2 and SIM 3, the achievable average localization error is in the range of about 2 m. The average error using SIM 1 is slightly higher. For the following experiments, we use the same parameter $\theta_{\mathrm{RSS}} = -65\,\mathrm{dBm}$ for all simulation scenarios. Note that this threshold is not optimal for scenario SIM 3, which explains why in some experiments the observed average localization error of SIM 3 will be slightly higher than the one of SIM 2.

10.5.2 Influence of Node Number and Refinement

In this section, we examine the influence of the number of static nodes on the localization accuracy. The node number influences the localization result, as a larger number of nodes also entails a higher node density per area, which in turn means that more pairwise distance estimates are available to compute the embedding of the network.

To assess the influence of the force-directed refinement, the average node localization error using MDS-IND and the average error using MDS-IND(R) are considered separately in Figures 10.9a and 10.9b, respectively. Each data point represents an average of 50 simulation runs, using different random node placements. In each run, the parameters $\theta_{\mathrm{RSS}} = -65\,\mathrm{dBm}$ and $\theta_{\mathrm{dist}} = 8\,\mathrm{m}$ were used.



(a) MDS-IND

(b) MDS-IND(R)

Figure 10.9: Average node localization error in dependence of the number of static sensor nodes.

Considering the result of MDS-IND, all three simulation scenarios approach an average localization error of roughly 3 m when more than 60 sensor nodes are used. For less than 50 sensor nodes, the localization accuracy decreases quickly. Among the different simulation scenarios, SIM 1 results in the worst results. The reason is that SIM 1 is based on a short walking trajectory, which covers only parts of the floor. SIM 2 and SIM 3 achieve comparable results, with SIM 2 showing slightly lower localization errors. As mentioned in the last section, this is partly caused by the parameter setting $\theta_{\mathrm{RSS}} = -65\,\mathrm{dBm}$, which is not optimal for scenario SIM 3. However, Figure 10.8 suggests that even with a better choice of θ_{RSS}, the significantly longer walking trajectory of scenario SIM 3 does not result in improved localization results.

Figure 10.9b reveals that the additional force-directed refinement results in significantly reduced localization errors. The rough tendency of the average localization error is similar to the results without refinement, but for network sizes of 60 nodes and above, the refinement results in accuracy improvements of almost 1 m. Additionally, for network sizes approaching 100 nodes, the differences between scenarios SIM 2 and SIM 3 seem to diminish.

10.5.3 Localization with Additional Movement Knowledge

Some existing work on indirect network localization assumes that information about the movement of the mobile device is available. To examine how much improvement this additional knowledge enables, we consider in this paragraph the most extreme case by assuming that the exact positions of the mobile device at each measurement are known. To utilize the additional movement knowledge, we use the probabilistic indirect localization approach that was presented in [SR04b] and the force-directed position estimation approach of Chapter 8.2. Further details on both algorithms are given in Section 10.4.

Figure 10.10 shows the average localization errors of both approaches for the three simulation scenarios of Section 7.4.2. Each data point is based on 20 different runs with varying node placements.

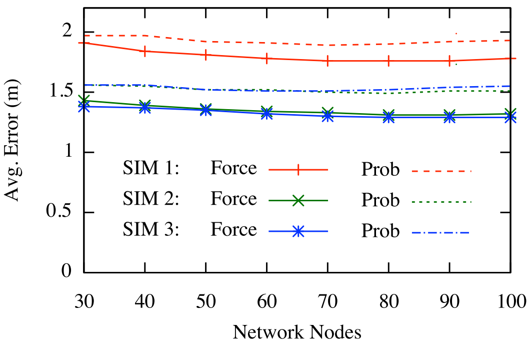


Figure 10.10: Average node localization error for the probabilistic approach (Prob) and the force-directed approach (Force), which both use full knowledge about the trajectory of the mobile device.

It is not surprising that in this scenario the number of nodes has no significant impact on the average localization accuracy, since the localization process of one node is not influenced by other nodes. However, the trajectory of the mobile device that is used for the measurements makes a difference. In general, more measurements result in improved localization results. Additionally, it is advantageous for the localization result if measurements are taken from different directions around a node. This is also reflected in the simulation results. The trajectory of scenario SIM 1 resulted in the worst localization results, whereas SIM 3 produced the best results. However, the difference between SIM 2 and SIM 3 is extremely small, so it is not likely that longer walking trajectories would result in much better localization results.

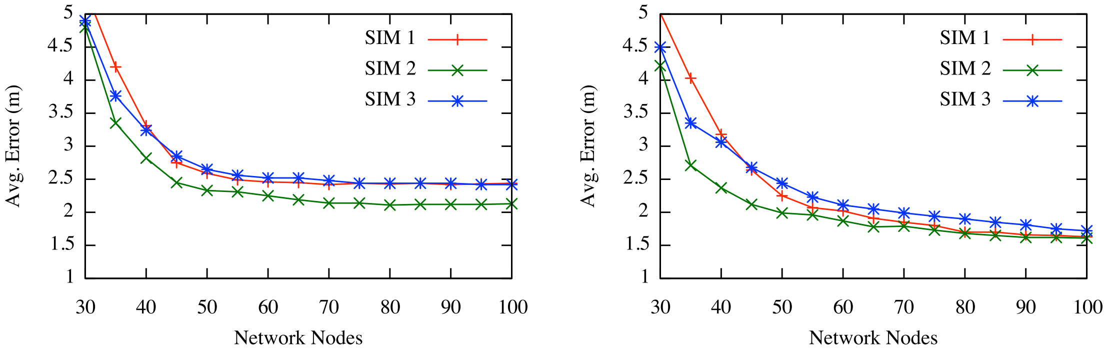
Comparing the considered approaches with each other, the difference in localization accuracy is small, with the force-directed approach producing slightly smaller errors on average. Taking into account that the force-directed approach also is significantly cheaper in terms of computational effort, this result is rather unexpected.

Next, we take a look at the performance of MDS-IND(R) in comparison to the approaches considered in this section. For MDS-IND(R), the average node localization error depends on the number of network nodes. For network sizes of 70 nodes and above, the average localization error approaches about 2 m to 2.5 m in scenario SIM 1 and about 1.5 m to 2 m in scenarios SIM 2 and SIM 3 (cf. Figure 10.9). This is about 50 cm higher than the force-directed approach that is considered in this section. Considering that the approaches of this section use exact knowledge about the walked trajectory, whereas MDS-IND(R) does not require any such knowledge, this difference is surprisingly small.

10.5.4 Application: Tracking

When the embedding of MDS-IND(R) is used for a subsequent tracking of mobile devices, the resulting localization accuracy depends in two ways on the number of network nodes: First, as observed in the previous section, the uncertainty in the computed positions of the static beacon nodes is influenced by the overall number of nodes in the network. Second, independent of the accuracy of the beacon node positions, a higher density of beacon nodes usually results in improved tracking accuracies.

To evaluate how well the node position estimates of MDS-IND(R) are suited for the tracking of mobile devices, we use the following approach: For each simulation scenario, the measured signal strengths are first used to compute an embedding of the beacon nodes. Subsequently, the computed node positions are used in combination with the tracking approach of Section 9.2. Like in Section 9.2, the quality of the computed trajectory is then evaluated using the mean distance between the true trajectory positions and the computed trajectory positions. Figure 10.11 shows the achieved accuracies for the force-directed tracking after the beacon node positions have been computed using either MDS-IND or MDS-IND(R). Every data point represents an average of 50 runs with varying sensor node placements.



(a) MDS-IND (b) MDS-IND(R)

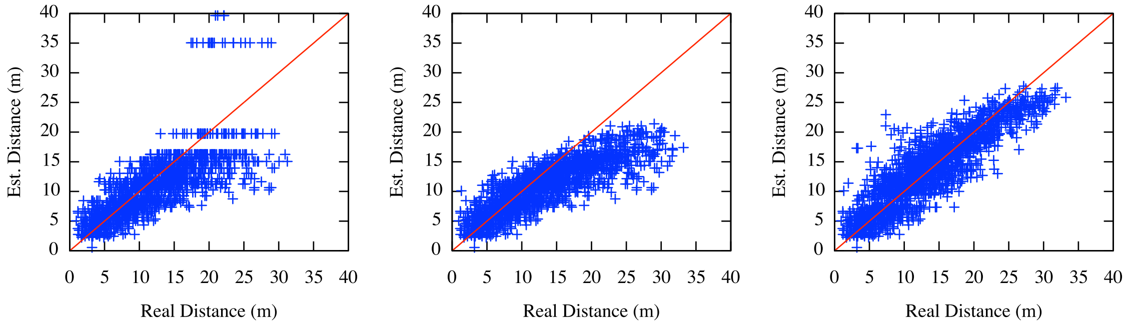
Figure 10.11: Average tracking error, using estimated node positions instead of true node positions. For the RSS-based tracking, the force-directed tracking approach was used.

For the MDS-IND approach without refinement, the average tracking error levels off in the range of 2 m to 2.5 m. Again, using the additional refinement of the sensor node positions pays off in terms of achievable accuracy. For the tracking after computing the node positions with MDS-IND(R), the average tracking error in the simulations approaches the range of 1.5 m to 2 m for sufficiently high numbers of sensor nodes. As before, using less than 50 sensor nodes results in significantly worse localization results. Overall, the simulations suggest that MDS-IND(R) is able to achieve sufficiently accurate estimates to use the computed positions for subsequent RSS-based tracking of mobile nodes.

10.6 Application in Real-World Networks

To conclude this chapter, we present some examples of how MDS-IND(R) works in connection with real-world measurements. Again, we use the experiments as described in Section 7.1, but this time we assume that the trajectory of the mobile node is completely unknown.

The first two examples show how the result of MDS-IND(R) looks if everything works as intended. The first example is based on experimental run ITI 2. Similarly to Figure 10.4, Figure 10.12 shows how the estimates of pairwise node distances correlate with the true distances. Apparently, the experiment shows the same general behavior as the simulations: After the one-way distance estimation, there are many pairs of nodes which are separated by a large distance, but which are estimated to be only 10 m to 15 m apart. If the shortest path computation is used without threshold, large distances are systematically underestimated. By using only distances up to $\theta_{\text{dist}} = 7.5$ m in the shortest path computations, this systematic underestimation of large distances can be avoided. Note that the final result is significantly influenced by an adequate choice of θ_{dist}. For this example, we found a reasonable value using a trial-and-error approach. Determining θ_{dist} automatically is an interesting topic for future research.



(a) One-way distance estimates. (b) Compl. distances, no cutoff. (c) Compl. distances, $\theta_{\text{dist}} = 7.5$ m.

Figure 10.12: Estimation of pairwise distances between static nodes. Each data point shows for a single pair of static nodes how the estimated distance correlates to the true distance. The plots are based on experimental scenario ITI 2 with 60 sensor nodes. (a) Situation after the initial one-way distance estimation. (b) Situation after shortest path distance completion without threshold. (c) Situation after shortest path distance completion with $\theta_{\text{dist}} = 7.5$ m.

The final embedding using MDS-IND(R) is shown in Figure 10.13. Additionally, a force-directed reconstruction of the walked trajectory, which is based on the estimated beacon positions, is shown. Taking into account that some of the estimated beacon positions differ significantly from the true positions, the computed walking trajectory is unexpectedly close to the real trajectory. Note that both the map and the true node positions are only shown to provide a visual impression of the errors that are introduced in the estimation process. The real output of the algorithms only consists of the estimated node positions and the reconstructed trajectory (cf. Figure 10.7).

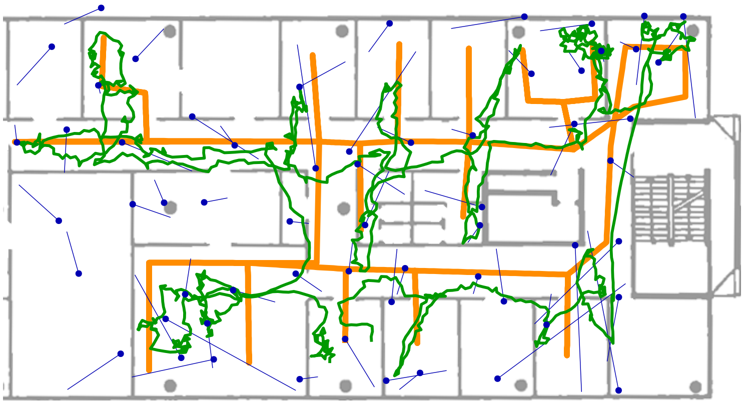
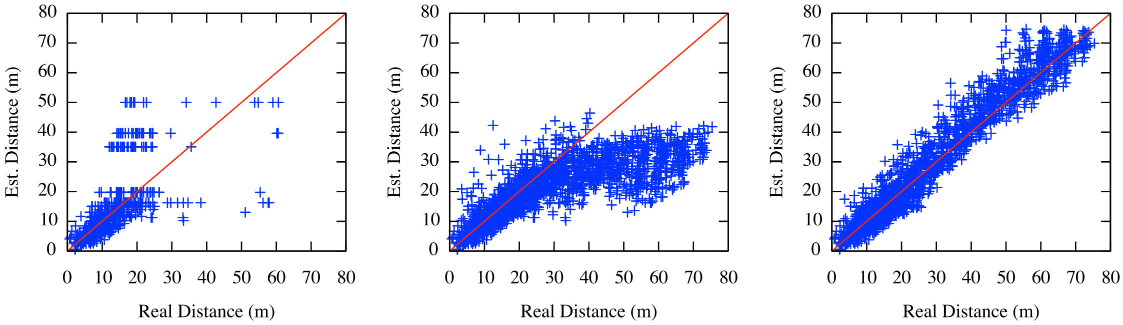


Figure 10.13: Example for the result of MDS-IND(R) on the data of experimental run ITI 2. Blue points show the estimated sensor node positions, blue lines emerging from the computed sensor node positions point to the true position of the sensor node. The orange line shows the trajectory of the mobile node that was used to collect the RSS data. The green line shows an estimate of the walked trajectory using the force-directed tracking algorithm with the estimated beacon node positions.

The second real-world example is based on an experiment at the ITIV. The corresponding building presents a good example to demonstrate two additional difficulties that can arise in connection with RSS-based localization of sensor networks. Both issues are caused by the U-shape of the building (cf. Figure 10.13). First, the U-shape enables line-of-sight connections even over large distances. For instance, if a mobile node makes RSS measurements in one wing of the building, there is a good chance that some packets of sensor nodes in the other wing of the building are received with relatively high signal strength. This effect is caused by line-of-sight connections through windows from one side of the building to the other side. In the process of RSS-based distance estimation, this can result in extremely underestimated distances. Results of this effect can be seen clearly in Figure 10.14a. Even for some node-pairs which are separated by more than 50 m, the one-way distance estimation returns distance estimates of less than 15 m. Without a threshold on the distance estimates that are used in the shortest path distance computation, the shortest path computation causes a severe underestimation of large distances, as can be seen in Figure 10.14b. However, by using a threshold of $\theta_{\text{dist}} = 8\,\text{m}$, the problem can be



(a) One-way distance estimates. (b) Compl. distances, no cutoff. (c) Compl. distances, $\theta_{\text{dist}} = 8\,\text{m}$.

Figure 10.14: Estimation of pairwise distances between static nodes. The plots are based on an experimental run in the ITIV floor with 60 sensor nodes.

successfully avoided in the considered experiment (cf. Figure 10.14c).

The second problem that can be caused by buildings in U-shape can actually result in an overestimation of distances. As discussed, one usually should try to filter those distance estimates that correspond to node pairs which have direct line-of-sight through windows of the building. As a result, there should be no direct distance estimates of node pairs in which one node is situated in one wing of the building and the other node is situated in the other wing of the building. In consequence, the distance between such pairs of nodes has to be estimated based on shortest path computations. However, as the shortest path will go mostly through the inner part of the building, it is likely that the distances between nodes in different wings of the building are systematically overestimated. To a certain degree, this effect can also be seen in Figure 10.14c.

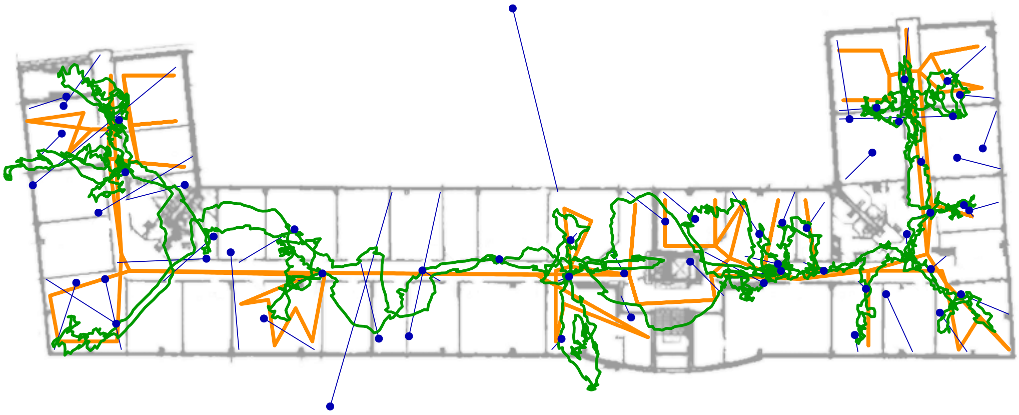


Figure 10.15: Example for a result of MDS-IND(R) on the data of an experiment at ITIV. A reconstruction of the walked trajectory using the force-directed tracking approach in combination with the estimated node positions is shown by the green line.

This being said, the embedding corresponding to the plots in Figure 10.14 is shown in Figure 10.15. For the most part, the reconstructed trajectory in Figure 10.15 reasonably resembles the truly walked trajectory. However, apparently there are also some parts where the deviation between the computed path and the ground truth is significant.

Finally, we also want to show an example of an embedding, where the node positions could not be reconstructed so nicely. Figure 10.16 shows an embedding of an experiment

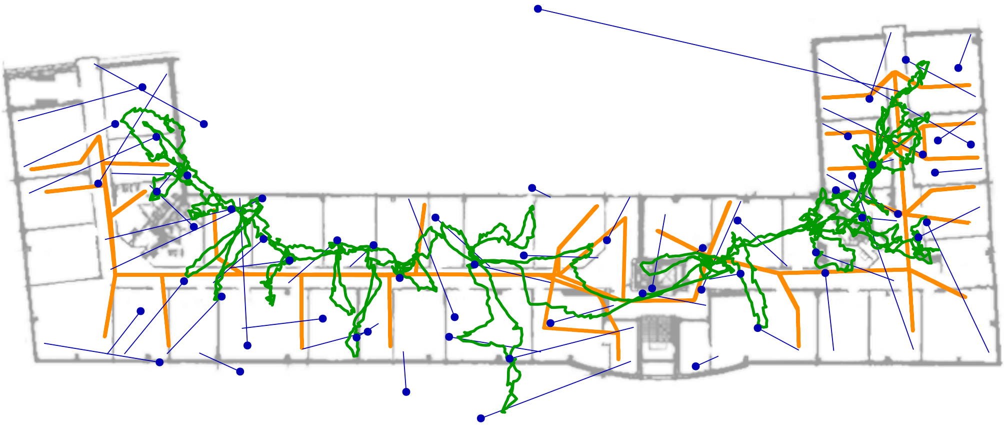


Figure 10.16: Example for a result of MDS-IND(R) based on the data of an experiment at ITIV. A reconstruction of the walked trajectory using the force-directed tracking approach in combination with the estimated node positions is shown by the green line.

in which the used threshold $\theta_{\text{dist}} = 8\,\text{m}$ was not sufficient to filter all distance estimates that were significantly too low. As a result, the embedding is compressed and the U-shape is somewhat stretched out. However, for scenarios in which a rough approximation of the network embedding and the walked trajectory is sufficient, the result might be still sufficient. Unfortunately, embeddings such as the one in Figure 10.16 were not uncommon for embeddings based on our real-world data. Accordingly, improved methods for the detection of strongly underestimated distances might be necessary in order to enable a robust and reliable application in reality.

10.7 Summary

In this chapter, we considered the indirect and anchor-free localization of wireless sensor networks based on signal strength measurements in indoor environments. In contrast to most existing work on indirect localization, we assumed that no knowledge about the movement of the mobile device or about reference positions is available.

In the considered scenario, the high fluctuations of received signal strengths in indoor environments, the systematic attenuation of signals at walls, and the limited sensitivity of wireless receivers, introduce additional difficulties for the indirect estimation of distances, which, in existing work, are often not relevant (e.g., because distance estimation approaches with significantly higher accuracy are used, such as TDoA) or not considered (e.g., because simplifying simulations are used). We described these difficulties and presented simple and efficient solution strategies.

To evaluate the achievable localization accuracy, we used simulations modeling communication in an indoor environment. In the simulations, the influence of the number of network nodes and the influence of the movement trajectory of the mobile device were studied. Additionally, the improvements that are achieved by a force-directed refinement were analyzed. In our simulations, the refinement improved the accuracy of the sensor node localization by almost 1 m on average.

To assess the localization quality, a comparison with two approaches for indirect network localization, which additionally used full knowledge about the movement trajectory of the mobile device, was presented. The difference in localization accuracy between the indirect approach without such knowledge and the two other approaches was surprisingly small—the average node localization error differed in the simulations only by about 50 cm.

As one possible application of the presented approach, the RSS-based tracking of mobile devices based on the estimated beacon node positions was examined. The achievable tracking accuracy was examined in simulations, also considering the influence of the number of beacon nodes.

Finally, some real-world examples were presented, and additional difficulties that might occur in a real-world application of the presented approach were discussed.

To summarize the results, in the simulations the indirect embedding usually worked very well, with an average node localization error in the range between 2 m and 2.5 m after the force-directed refinement. The average localization error for the tracking based on the computed node positions was in the range of about 2 m, only about 0.5 m to 0.8 m larger than for tracking based on the true node positions. However, our real-world experiments indicated that for a reliable and robust application in reality, an additional outlier detection for strongly underestimated distances might be necessary.

Chapter 11

Conclusion

From an algorithmic point of view, wireless sensor networks offer many new and interesting challenges. Among these challenges, this work addressed different aspects of communication and localization in wireless sensor networks.

Communication in wireless sensor networks. In the area of communication, new data structures for the efficient computation of optimum transmission powers were introduced. The focus in the design of the data structures was on enabling efficient recomputation of optimum powers when there are only slight changes in the input. The presented data structures differed in the kinds of supported operations and the corresponding complexities. The first data structure was optimized for situations in which wireless transmissions are activated one after another. In this scenario, the data structure allows to compute very efficiently how optimum transmission powers change if inactive transmissions are additionally activated. An application for this data structure are time division multiple access scheduling algorithms that fill time slots iteratively. The second data structure additionally saves information about the operations that resulted in the current state of the underlying matrices. Using this additional information, the data structure can provide further operations efficiently, e.g., the deactivation of transmissions. Finally, a modification of the second data structure was described, which allows to improve the complexities of some operations at the cost of a less efficient prediction of power changes for the case that transmissions are activated.

The second contribution in the area of communication dealt with the computation of efficient time division multiple access schedules. New heuristics were presented, which aim at minimizing the transmission powers that are necessary to schedule a set of transmissions within a given number of time slots. In contrast to existing approaches, the new heuristics base their decisions on the resulting effects on optimum transmission powers. To compute power changes efficiently, the aforementioned data structures for dynamic power control were used. In a simulation-based comparison with existing approaches, it turned out that this scheduling strategy not only reduces the necessary transmission powers, but it also results in throughput improvements. As a second difference to existing scheduling algorithms, two of the presented heuristics do not fill time slots greedily one-by-one, but instead they consider several time slots in parallel and assign transmissions to slots where they fit best. This allowed the computation of more balanced schedules, which required significantly lower average transmission powers. In addition, this strategy can be used to find a compromise between throughput and power consumption.

The last contribution dealing with communication in wireless networks considered the complexity of scheduling with power control in the geometric SINR_G model. NP-hardness proofs for the problems of scheduling with power control and one-shot scheduling with power control were presented. The presented proofs are applicable in scenarios where the allowed transmission powers are bounded, including the situation where the powers can be chosen from a finite set of allowed powers.

Localization in wireless sensor networks. The first problem that was considered in the area of localization in wireless sensor networks was the problem of connectivity-based boundary recognition. A new algorithm was presented, EC-BR, which enables each node to decide autonomously, based solely on connectivity information of its local 2-hop neighborhood, whether it is situated in close proximity to an area of low node density or an outer boundary of the network. Several simulation scenarios were considered to assess the performance of EC-BR in comparison to existing algorithms for boundary recognition. At this, different node densities, node placement strategies, and communication models were examined. In comparison to the other algorithms, EC-BR produced surprisingly good classification results, with low misclassifications rates for both boundary nodes and inner nodes.

The remainder of this thesis dealt with localization based on received signal strengths (RSS). First, the real-world sensor network experiments that are used in this thesis were described, and a preliminary analysis of the experimental data was presented. The observations of this analysis were then used to motivate the simulation setup that was used in the following chapters for the simulation-based evaluation of localization algorithms. Additionally, a comparison between simulations and experiments was presented.

The next two chapters dealt with force-directed localization in wireless networks. Force-directed approaches have been used frequently before to compute or refine embeddings of whole sensor networks. In this thesis, the focus was on force-directed localization and tracking of single mobile nodes, using additional information from movement models or inertial data.

In the first chapter, the static localization problem was considered, where only RSS-based distance estimates to beacon nodes with known positions are used to localize single nodes. Several possibilities to represent RSS-based distance estimates by spring forces were compared experimentally. In addition, further existing localization approaches were analyzed, to enable an assessment of the quality of the force-directed position estimation.

The second chapter on force-directed localization dealt with the problem of RSS-based tracking in wireless sensor networks. Different possibilities to represent movement knowledge with additional forces have been presented. Additionally, similarities and differences to tracking with extended Kalman filters (EKF) have been discussed. The force-directed tracking approaches and comparable implementations of extended Kalman filters have then been analyzed based on the real-world data. In the experiments, the influence of the beacon number and the influence of the number of RSS measurements that are used per time step were examined. Additionally, simulations were used to verify the experimental results. Both the experiments and the simulations showed slightly lower localization errors for the force-directed tracking approach. In addition to the slightly improved localization results, the main advantage of the force-directed tracking approach probably lies in its ability to represent all kinds of knowledge very easily by the introduction of additional forces. Force-definitions are not limited to simple functions, and both time and space complexity of the approach are linear in the number of used forces. This makes the force-directed tracking approach well suited even for application on hardware with limited resources.

The final chapter of this thesis dealt with the automatic localization of wireless sensor networks. An extension of the well-known MDS-MAP approach was presented, which allows the indirect localization of wireless sensor networks. While in the original MDS-MAP approach communication between static sensor nodes is used to estimate pairwise distances between nodes, the presented approach uses a mobile device to indirectly estimate distances in the sensor network. In contrast to most existing work on indirect localization, the presented approach uses no knowledge about node locations or the movement of the mobile device. Additionally, special focus was put on the difficulties that arise due to the high uncertainties of RSS-based distance estimations in indoor environments. The examined approach was then evaluated based on simulations. Three different walking scenarios have been considered to evaluate the influence of the walked trajectory on the resulting localization accuracy. Additionally, a comparison with two other approaches, which additionally used full knowledge about the true trajectory of the mobile node, has been presented. As a possible application of the indirect localization approach, the subsequent RSS-based tracking based on the computed beacon positions has been studied. The chapter was concluded with examples of embeddings that were computed based on real-world data, and with a discussion of additional difficulties that might arise when the presented approach is used in practice.

Bibliography

[AC09] M. Azizyan and R.R. Choudhury. SurroundSense: Mobile Phone Localization using Ambient Sound and Light. *ACM SIGMOBILE Mobile Computing and Communications Review*, 13(1):69–72, 2009.

[ACZ01] J. Albowicz, A. Chen, and Lixia Zhang. Recursive position estimation in sensor networks. In *Proceedings of the Ninth International Conference on Network Protocols*, pages 35–41, 2001.

[ACZ05] Aaron Ault, Edward Coyle, and Xuan Zhong. K-Nearest-Neighbor Analysis of Received Signal Strength Distance Estimation Across Environments. In *Proceedings of the 1st Workshop on Wireless Network Measurements (WiNMee 2005)*, 2005.

[Aei73] J. M. Aein. Power balancing in systems employing frequency reuse. *COMSAT Technical Review*, 3(2), 1973.

[ARY94] J. B. Anderson, T. S. Rappaport, and S. Yoshida. Propagation Measurements and Models for Wireless Communications Channels. *IEEE Communications Magazine*, 33(1), November 1994.

[ASSC02] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.

[BBS06] Gurashish Singh Brar, Douglas M. Blough, and Paolo Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pages 2–13, 2006.

[Ber87] R.C. Bernhardt. Macroscopic Diversity in Frequency Reuse Systems. *IEEE Journal on Selected Areas in Communications*, 5:862–878, June 1987.

[BGMS06] Amitabh Basu, Jie Gao, Joseph S. B. Mitchell, and Girishkumar Sabhnani. Distributed localization using noisy distance and angle information. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '06)*, pages 262–273, 2006.

[BHE00] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, October 2000.

[BK04] R. Bhatia and M. Kodialam. On power efficient communication over multi-hop wireless networks: joint routing, scheduling and power control. In *Proceedings of the 23rd Conference on Computer Communications (INFOCOM '04)*, volume 2, pages 1457–1466. IEEE, 2004.

[BOGVB10] M. Bshara, U. Orguner, F. Gustafsson, and L. Van Biesen. Fingerprinting Localization in Wireless Networks Based on Received-Signal-Strength Measurements: A Case Study on WiMAX Networks. *IEEE Transactions on Vehicular Technology*, 59(1):283–294, January 2010.

[Bou09] A. Boukerche. *Algorithms and protocols for wireless sensor networks.* Wiley series on parallel and distributed computing. Wiley, 2009.

[BP00] P. Bahl and V.N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, volume 2, pages 775–784. IEEE, 2000.

[BR03] A. Behzad and I. Rubin. On the performance of graph-based scheduling algorithms for packet radio networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*, volume 6, pages 3432–3436, December 2003.

[BR05] A. Behzad and I. Rubin. Impact of power control on the performance of ad hoc wireless networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pages 102–113, March 2005.

[BTG$^+$06] Kun Bi, Kun Tu, Naijie Gu, Wan Lin Dong, and Xiaohu Liu. Topological Hole Detection in Sensor Networks with Cooperative Neighbors. In *Proceedings of the International Conference on Systems and Networks Communication (ICSNC '06)*, pages 31–35, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[BVY04] Patrik Björklund, Peter Värbrand, and Di Yuan. A column generation method for spatial TDMA scheduling in ad hoc networks. *Ad Hoc Networks*, 2(4):405–418, 2004.

[CC01] Trevor F. Cox and Michael A. A. Cox. *Multidimensional Scaling.* Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, second edition, 2001.

[CEL$^+$11] Xi Chen, A. Edelstein, Yunpeng Li, M. Coates, M. Rabbat, and A. Men. Sequential Monte Carlo for simultaneous passive device-free tracking and sensor localization using received signal strength measurements. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN '11)*, pages 342–353, April 2011.

[Che06] Qi Cheng. On Comparing Sums of Square Roots of Small Integers. In *Mathematical Foundations of Computer Science 06*, volume 4162 of *LNCS*, pages 250–255. Springer, 2006.

[CKKK10] Kevin Coogan, Varun Khare, Stephen G. Kobourov, and Bastian Katz. Multi-Scale Dead-Reckoning Algorithm for Distributed Force-Directed Sensor Network Localization. In *Proceedings of the 6th International Conference on Algorithms for Sensor Systems, Wireless Adhoc Networks, and Autonomous Mobile Entities*, pages 148–160, 2010.

[CKM$^+$07] Deepti Chafekar, V.S. Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan. Cross-layer latency minimization in wireless networks with SINR constraints. In *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'07)*, pages 110–119. ACM, 2007.

[CLRS09] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* The MIT Press, 3rd edition, 2009.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on the Theory of Computing*, pages 151–158, 1971.

[CPI06] Jose A. Costa, Neal Patwari, and Alfred O. Hero III. Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks*, 2(1):39–64, 2006.

[CS03] R. L. Cruz and A. V. Santhanam. Optimal routing, link scheduling and power control in multihop wireless networks. In *Proceedings of the 22nd Conference of Computer Communications (INFOCOM '03)*, volume 1, pages 702–711. IEEE, 2003.

[DBKR90] D.M.J. Devasirvatham, C. Banerjee, M.J. Krain, and D.A. Rappaport. Multi-frequency radiowave propagation measurements in the portable radio environment. In *Proceedings of the IEEE International Conference on Communications (ICC '90)*, volume 4, pages 1334–1340, April 1990.

[DDHG05] Jitender S. Deogun, Saket Das, Haitham S. Hamza, and Steve Goddard. An Algorithm for Boundary Discovery in Wireless Sensor Networks. In *12th International Conference on High Performance Computing (HiPC '05)*, pages 343–352, 2005.

[DEG⁺12] Christian A. Duncan, David Eppstein, Michael T. Goodrich, Stephen G. Kobourov, and Martin Nöllenburg. Lombardi Drawings of Graphs. *Journal of Graph Algorithms and Applications*, 16(1):85–108, 2012.

[Die10] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, 2010.

[DLL09] Dezun Dong, Yunhao Liu, and Xiangke Liao. Fine-grained boundary recognition in wireless ad hoc and sensor networks by topological methods. In *Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09)*, pages 135–144, New York, NY, USA, 2009. ACM.

[DPEG01] L. Doherty, K. Pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, volume 3, pages 1655–1663, 2001.

[DSG06] V. De Silva and R. Ghrist. Coordinate-free Coverage in Sensor Networks with Controlled Boundaries via Homology. *International Journal of Robotics Research*, 25(12):1205–1222, 2006.

[Ead84] P. Eades. A Heuristic for Graph Drawing. *Congressus Numerantium*, 42:149–160, 1984.

[EE02] T. ElBatt and A. Ephremides. Joint scheduling and power control for wireless ad-hoc networks. In *Proceedings of the 21st IEEE International Conference on Computer Communications (INFOCOM '02)*, volume 2, pages 976–984. IEEE, 2002.

[EE04] T. ElBatt and A. Ephremides. Joint scheduling and power control for wireless ad hoc networks. *IEEE Transactions on Wireless Communications*, 3(1):74–85, January 2004.

[EEF⁺06] Alon Efrat, Cesim Erten, David Forrester, Anand Iyer, and Stephen G. Kobourov. Force-Directed Approaches to Sensor Localization. In *Proceedings of the Eighth Workshop on Algorithm Engineering and Experiments (ALENEX '06)*, pages 108–118, 2006.

[EFI⁺10] Alon Efrat, David Forrester, Anand Iyer, Stephen G. Kobourov, Cesim Erten, and Ozan Kilic. Force-Directed Approaches to Sensor Localization. *ACM Transactions on Sensor Networks (TOSN)*, 7(3):1–25, 2010.

[FGG04] Qing Fang, Jie Gao, and Leonidas J. Guibas. Locating and Bypassing Routing Holes in Sensor Networks. In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pages 2458–2468, 2004.

[FK06] Stefan Funke and Christian Klein. Hole detection or: "how much geometry hides in connectivity?". In *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry (SCG '06)*, pages 377–385, USA, 2006. ACM.

[FKKL05] S. P. Fekete, M. Kaufmann, A. Kröller, and N. Lehmann. A new approach for boundary recognition in geometric sensor networks. In *Proceedings of the 17th Canadian Conference on Computational Geometry (CCCG '05)*, pages 82–85, 2005.

[FKP⁺04] Sándor P. Fekete, Alexander Kröller, Dennis Pfisterer, Stefan Fischer, and Carsten Buschmann. Neighborhood-Based Topology Recognition in Sensor Networks. In *Proceedings of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS '04)*, pages 123–136, 2004.

[FKV09] Alexander Fanghänel, Thomas Kesselheim, and Berthold Vöcking. Improved Algorithms for Latency Minimization in Wireless Networks. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP'09)*, pages 447–458, 2009.

[FM93] G. J. Foschini and Z. Miljanic. A simple distributed autonomous power control algorithm and its convergence. *IEEE Transactions on Vehicular Technology*, 42(4):641–646, Nov 1993.

[FR91] Thomas M. J. Fruchterman and Edward M. Reingold. Graph Drawing by Force-directed Placement. *Software - Practice and Experience (SPE)*, 21(11):1129–1164, 1991.

[Fun05] Stefan Funke. Topological hole detection in wireless sensor networks and its applications. In *Proceedings of the 2005 Joint Workshop on Foundations of Mobile Computing (DIALM-POMC '05)*, pages 44–53, USA, 2005. ACM.

[GC01] Pei Guangyu and C. Chien. Low power TDMA in large wireless sensor networks. In *Proceedings of the IEEE Military Communications Conference (MILCOM '01)*, pages 347–351. IEEE, 2001.

[Gez08] Sinan Gezici. A Survey on Wireless Position Estimation. *Wireless Personal Communications: An International Journa*, 44(3):263–282, February 2008.

[GGK$^+$03] Aram Galstyan, Aram Galstyan, Bhaskar Krishnamachari, Kristina Lerman, and Sundeep Pattem. Distributed Online Localization in Sensor Networks Using a Moving Target. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pages 61–70, 2003.

[GH01] Jimmi Grönkvist and Anders Hansson. Comparison between graph-based and interference-based STDMA scheduling. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '01)*, pages 255–258. ACM, 2001.

[GHWW09] Olga Goussevskaia, Magnus Halldorsson, Roger Wattenhofer, and Emo Welzl. Capacity of Arbitrary Wireless Networks. In *Proceedings of the 28th Annual IEEE Conference on Computer Communications (INFOCOM '09)*, April 2009.

[Gis08] Drew Gislason. *Zigbee Wireless Networking.* Newnes, 2008.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[GK00] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[GM05] Robert Ghrist and Abubakr Muhammad. Coverage and hole-detection in sensor networks via homology. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pages 254–260, USA, 2005. IEEE Press.

[GOW07] Olga Goussevskaia, Yvonne Anne Oswald, and Roger Wattenhofer. Complexity in Geometric SINR. In *Proceedings of the 8th ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc '07)*, pages 100–109, New York, NY, USA, 2007. ACM.

[GPW10] Olga Goussevskaia, Yvonne-Anne Pignolet, and Roger Wattenhofer. Efficiency of Wireless Networks: Approximation Algorithms for the Physical Interference Model. *Foundations and Trends in Networking*, 4(3):313–420, 2010.

[GVG94] S.A. Grandhi, R. Vijayan, and D.J. Goodman. Distributed power control in cellular radio systems. *IEEE Transactions on Communications*, 42(234):226–228, 1994.

[GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, MD, 3rd edition, October 1996.

[GWJ+11] Qinghua Gao, Jie Wang, Minglu Jin, Hongyang Chen, and Hongyu Wang. Target tracking by lightweight blind particle filter in wireless sensor networks. *Wireless Communications and Mobile Computing*, 2011.

[Hal09] Magnús M. Halldórsson. Wireless Scheduling with Power Control. In *Proceedings of the 17th European Symposium on Algorithms (ESA'09)*, pages 361–372. Springer, 2009.

[HE04] Lingxuan Hu and David Evans. Localization for mobile sensor networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pages 45–57, 2004.

[HHB+03] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pages 81–95, New York, NY, USA, 2003. ACM.

[HM11] M. M. Halldórsson and P. Mitra. Nearly Optimal Bounds for Distributed Wireless Scheduling in the SINR Model. In *Proceedings of the 38th International Conference on Automata, Languages and Programming (ICALP '11)*, April 2011.

[HMS01] A. Howard, M.J. Mataric, and G. Sukhatme. Relaxation on a Mesh: a Formalism for Generalized Localization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1055 –1060, 2001.

[HW09] Magnús M. Halldórsson and Roger Wattenhofer. Wireless Communication Is in APX. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP '09)*, pages 525–536, Berlin, Heidelberg, 2009. Springer-Verlag.

[HWBL+08] Hofmann-Wellenhof, Bernhard, Lichtenegger, Herbert, Wasle, and Elmar. *GNSS – Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more.* Springer, 2008.

[HWLC97] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice.* Springer-Verlag, Berlin, 4th edition, 1997.

[JPPQ03] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MOBICOM '03)*, pages 66–80. ACM, 2003.

[JSPG10] A.R. Jiménez, F. Seco, J.C. Prieto, and J. Guevara. Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU. In *Proceedings of the 7th Workshop on Positioning Navigation and Communication (WPNC '10)*, pages 135–143, March 2010.

[JUDW95] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the 1995 American Control Conference*, pages 1628–1632, 1995.

[Jun08] Dieter Jungnickel. *Graphs, Networks and Algorithms.* Springer-Verlag, 3rd edition, 2008.

[Kal60] Rudolph Emil Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[Kar72] R. M. Karp. Reducibility Among Combinatorial Problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[Kat09] Bastian Katz. *Positioning and Scheduling of Wireless Sensor Networks - Models, Complexity, and Scalable Algorithms.* PhD thesis, Universität Karlsruhe (TH), 2009.

[Kay93] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice Hall PTR, 1993.

[Kes11] Thomas Kesselheim. A Constant-Factor Approximation for Wireless Capacity Maximization with Power Control in the SINR Model. In *Proceedings of the 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA '11)*, pages 1549–1559, 2011.

[KFPF06] A. Kröller, S. P. Fekete, D. Pfisterer, and S. Fischer. Deterministic boundary recognition and topology extraction for large sensor networks. In *Proceedings of the 17th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '06)*, pages 1000–1009, 2006.

[KGW07] Bastian Katz, Marco Gaertler, and Dorothea Wagner. Maximum Rigid Components as Means for Direction-Based Localization in Sensor Networks. In *Proceedings of the 33rd conference on Current Trends in Theory and Practice of Computer Science (SOFSEM '07)*, pages 330–341, 2007.

[KKP99] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: mobile networking for Smart Dust. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '99, pages 271–278, New York, NY, USA, 1999. ACM.

[KKT04] Ulas C. Kozat, Iordanis Koutsopoulos, and Leandros Tassiulas. A Framework for Cross-layer Design of Energy-efficient Communication with QoS Provisioning in Multi-hop Wireless Networks. In *Proceedings of the 22nd Conference of Computer Communications (INFOCOM '04)*, pages 1446–1456. IEEE, 2004.

[KMPS04] V. S. Anil Kumar, Madhav V. Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan. End-to-end packet-scheduling in wireless ad-hoc networks. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '04)*, pages 1021–1030, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[KMW08] Bastian Katz, Steffen Mecke, and Dorothea Wagner. Efficient Scheduling of Data-Harvesting Trees. In *Proceedings of the 4th International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS '08)*, pages 43–56, 2008.

[Knu76] Donald E. Knuth. Big Omicron and big Omega and big Theta. *SIGACT News*, 8:18–24, April 1976.

[KPV07] Azadeh Kushki, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. Kernel-Based Positioning in Wireless Local Area Networks. *IEEE Transactions on Mobile Computing*, 6(6):689–705, June 2007.

[KV10] Thomas Kesselheim and Berthold Vöcking. Distributed contention resolution in wireless networks. In *Proceedings of the 24th International Conference on Distributed Computing (DISC'10)*, pages 163–178. Springer-Verlag, 2010.

[KVW08] Bastian Katz, Markus Völker, and Dorothea Wagner. Link Scheduling in Local Interference Models. In *Proceedings of the 4th International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS '08)*, pages 57–71. Springer, 2008.

[KVW10] Bastian Katz, Markus Völker, and Dorothea Wagner. Energy Efficient Scheduling with Power Control for Wireless Networks. In *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt 2010)*, pages 160–169. IEEE Computer Society, 2010.

[KW01] Michael Kaufmann and Dorothea Wagner, editors. *Drawing Graphs: Methods and Models*, volume 2025 of *Lecture Notes in Computer Science*. Springer, 2001.

[KW07] Bastian Katz and Dorothea Wagner. Multiscale Anchor-Free Distributed Positioning in Sensor Networks. In *Proceedings of the International Workshop on Theoretical and Algorithmic Aspects of Sensor and Ad-hoc Networks (WTASA'07)*, 2007.

[LE05] Yun Li and Anthony Ephremides. Joint Scheduling, Power Control, and Routing Algorithm for Ad-Hoc Wireless Networks. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*. IEEE, 2005.

[Li06] X. Li. RSS-Based Location Estimation with Unknown Pathloss Model. *IEEE Transactions on Wireless Communications*, 5(12):3626–3633, 2006.

[LK05] Gang Lu and B. Krishnamachari. Energy efficient joint scheduling and power control for wireless sensor networks. In *2nd Annual Conference on Sensor and Ad Hoc Communications and Networks (SECON '05)*, pages 362–373. IEEE, 2005.

[LLS95] Tsern-Huei Lee, Jen-Cheng Lin, and Yu.T. Su. Downlink power control algorithms for cellular radio systems. *IEEE Transactions on Vehicular Technology*, 44(1):89–94, February 1995.

[LLW06] Bo-Chieh Liu, K.-H. Lin, and Jieh-Chian Wu. Analysis of hyperbolic and circular positioning algorithms using stationary signal-strength-difference measurements in wireless communications. *IEEE Transactions on Vehicular Technology*, 55(2):499–509, March 2006.

[LvRW08] Thomas Locher, Pascal von Rickenbach, and Roger Wattenhofer. Sensor Networks Continue to Puzzle: Selected Open Problems. In *Proceedings of the 9th International Conference on Distributed Computing and Networking (ICDCN'08)*, Kolkata, India, January 2008.

[LW07] K. Lorincz and M. Welsh. MoteTrack: A Robust, Decentralized Approach to RF-based Location Tracking. *Personal and Ubiquitous Computing*, 11(6):489–503, 2007.

[MAF07] G. Mao, B. Anderson, and B. Fidan. Path Loss Exponent Estimation for Wireless Sensor Network Localization. *Computer Networks*, 51(10):2467–2483, 2007.

[May79] Peter S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 141 of *Mathematics in Science and Engineering*. Academic Press, 1979.

[MFA07] Guoqiang Mao, Barış Fidan, and Brian D. O. Anderson. Wireless sensor network localization techniques. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 51(10):2529–2553, 2007.

[MJD08] Ritesh Maheshwari, Shweta Jain, and Samir R. Das. A measurement study of interference modeling and scheduling in low-power wireless networks. In *Proceedings of the 6th Conference on Embedded Network Sensor Systems (SenSys '08)*, pages 141–154. ACM, 2008.

[MLRT04] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust Distributed Network Localization with Noisy Range Measurements. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 50–61. ACM, 2004.

[MNR$^+$06] C. Morelli, M. Nicoli, V. Rampa, U. Spagnolini, and C. Alippi. Particle Filters for RSS-Based Localization in Wireless Sensor Networks: An Experimental Study. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, volume 4, 2006.

[MOW07] T. Moscibroda, Y. A. Oswald, and R. Wattenhofer. How Optimal are Wireless Scheduling Protocols? In *Proceedings of the 26th International Conference on Computer Communications (INFOCOM'07)*, pages 1433–1441. IEEE, May 2007.

[MS04] F. Martincic and L. Schwiebert. Distributed perimeter detection in wireless sensor networks. Technical Report WSU-CSC-NEWS/03-TR03, Wayne State University, 2004.

[MSZ$^+$06] Muralidhar Medidi, Roger Slaaen, Yuanyuan Zhou, Christopher Mallery, and Sirisha Medidi. Scalable Localization in Wireless Sensor Networks. In Yves Robert, Manish Parashar, Ramamurthy Badrinath, and Viktor Prasanna, editors, *High Performance Computing - HiPC 2006*, volume 4297 of *Lecture Notes in Computer Science*, pages 522–533. Springer Berlin / Heidelberg, 2006.

[MW05] Thomas Moscibroda and Roger Wattenhofer. Coloring unstructured radio networks. In *Proceedings of the Seventeenth Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '05)*, pages 39–48, New York, NY, USA, 2005. ACM.

[MW06] T. Moscibroda and R. Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pages 1–13, 2006.

[MWW06] Thomas Moscibroda, Roger Wattenhofer, and Yves Weber. Protocol Design Beyond Graph-Based Models. In *Proceedings of the 5th Workshop on Hot Topics in Networks (HotNets'06)*, 2006.

[MWZ06] Thomas Moscibroda, Roger Wattenhofer, and Aaron Zollinger. Topology control meets SINR: The Scheduling Complexity of Arbitrary Topologies. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'06)*, pages 310–321. ACM, 2006.

[NL02] Asis Nasipuri and Kai Li. A directionality based location discovery scheme for wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pages 105–111, New York, NY, USA, 2002. ACM.

[NN01] D. Niculescu and B. Nath. Ad Hoc Positioning System (APS). In *Proceedings of the Global Telecommunications Conference (GLOBECOM '01)*, volume 5, pages 2926–2931, 2001.

[NN03] Dragos Niculescu and Badri Nath. Ad Hoc Positioning System (APS) Using AoA. In *Proceedings of the 22nd Conference of the IEEE Computer and Communication Societies (INFOCOM '03)*, pages 1734–1743, 2003.

[OSW10] Georg Oberholzer, Philipp Sommer, and Roger Wattenhofer. The SpiderBat ultrasound positioning system. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*, pages 403–404, 2010.

[OSW11] G. Oberholzer, P. Sommer, and R. Wattenhofer. SpiderBat: Augmenting wireless sensor networks with distance and angle information. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN '11)*, pages 211–222, April 2011.

[otICS97] LAN MAN Standards Committee of the IEEE Computer Society. Wireless LAN medium access control (MAC) and physical layer (PHY) specification. *IEEE Std 802.11-1997*, November 1997.

[PAK+05] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero III, R.L. Moses, and N.S. Correal. Locating the Nodes: Cooperative Localization in Wireless Sensor Networks. *IEEE Signal Processing Magazine*, 22(4):54–69, 2005.

[PBDT03] Nissanka B. Priyantha, Hari Balakrishnan, Erik D. Demaine, and Seth J. Teller. Anchor-free distributed localization in sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pages 340–341, 2003.

[PBDT05] Nissanka B. Priyantha, Hari Balakrishnan, Erik D. Demaine, and Seth Teller. Mobile-Assisted Localization in Wireless Sensor Networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pages 172–183, 2005.

[PCB00] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket Location-Support System. In *Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MOBICOM '00)*, 2000.

[PCC+10] J. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an Organic Indoor Location System. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, pages 271–284. ACM, 2010.

[PMBT01] Nissanka Bodhi Priyantha, Allen K. L. Miu, Hari Balakrishnan, and Seth Teller. The Cricket Compass for Context-Aware Mobile Applications. In *Proceedings of the 6th ACM MOBICOM Conference*, 2001.

[PS07] Rong Peng and Mihail L. Sichitiu. Probabilistic localization for outdoor wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11:53–64, January 2007.

[PW09] A.S. Paul and E.A. Wan. RSSI-Based Indoor Localization and Tracking using Sigma-Point Kalman Smoothers. *IEEE Journal of Selected Topics in Signal Processing*, 3(5):860–873, 2009.

[QB79] N. Quinn and M. Breur. A force directed component placement procedure for printed circuit boards. *IEEE Transactions on Circuits and Systems*, 26(6):377–388, 1979.

[Rap91] T. S. Rappaport. The Wireless Revolution. *IEEE Communications Magazine*, pages 52–71, November 1991.

[Rap01] Theodore S. Rappaport. *Wireless Communications - Principles and Practice.* Prentice Hall, 2nd edition, 2001.

[RB00] S.I. Roumeliotis and G.A. Bekey. Synergetic localization for groups of mobile robots. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 4, pages 3477–3482, 2000.

[RD07] Masoomeh Rudafshani and Suprakash Datta. Localization in wireless sensor networks. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN '07)*, pages 51–60, 2007.

[RGHR10] A.R.J. Ruiz, F.S. Granja, J.C.P. Honorato, and J.I.G. Rosas. Pedestrian indoor navigation by aiding a foot-mounted IMU with RFID Signal Strength measurements. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN '10)*, pages 1–7, September 2010.

[RL93] Subramanian Ramanathan and Errol L. Lloyd. Scheduling Algorithms for Multihop Radio Networks. *IEEE/ACM Transactions on Networking*, 1:166–177, 1993.

[RM09] C. Rohrig and M. Muller. Indoor location tracking in non-line-of-sight environments using a IEEE 802.15.4a wireless network. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '09)*, pages 552–557, October 2009.

[Rob75] Lawrence G. Roberts. ALOHA packet system with and without slots and capture. *ACM SIGCOMM Computer Communication Review*, 5(2):28–42, 1975.

[RRP$^+$03] Ananth Rao, Sylvia Ratnasamy, Christos Papadimitriou, Scott Shenker, and Ion Stoica. Geographic routing without location information. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pages 96–108, New York, NY, USA, 2003. ACM.

[SBN$^+$11] J. Schmid, F. Beutler, B. Noack, U. Hanebeck, and K. Müller-Glaser. An Experimental Evaluation of Position Estimation Methods for Person Localization in Wireless Sensor Networks. In *Proceedings of the 8th International Conference on Wireless Sensor Networks (EWSN '11)*, pages 147–162. Springer, 2011.

[SCGL05] Guolin Sun, Jie Chen, Wei Guo, and K.J.R. Liu. Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs. *IEEE Signal Processing Magazine*, 22(4):12–23, July 2005.

[SDK08] Biljana Stojkoska, Danco Davcev, and Andrea Kulakov. Cluster-based MDS algorithm for nodes localization in wireless sensor networks with irregular topologies. In *Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology (CSTST '08)*, pages 384–389, New York, NY, USA, 2008. ACM.

[SGSMG11] J. Schmid, T. Gadeke, W. Stork, and K.D. Muller-Glaser. On the fusion of inertial data for signal strength localization. In *Proceedings of the 8th Workshop on Positioning Navigation and Communication (WPNC '11)*, pages 7–12, April 2011.

[SHS01] Andreas Savvides, Chih-Chieh Han, and Mani B. Srivastava. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In *Proceedings of the 7th Conference on Mobile Computing and Networking (MOBICOM '01)*, pages 166–179, 2001.

[SHS04] R. Stoleru, Tian He, and J.A. Stankovic. Walking GPS: a practical solution for localization in manually deployed wireless sensor networks. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, pages 480–489, 2004.

[SKL10] Chun-Jung Sun, Hong-Yi Kuo, and Chin E. Lin. A sensor based indoor mobile localization and navigation using Unscented Kalman Filter. In *Proceedings of the IEEE/ION Position Location and Navigation Symposium (PLANS '10)*, pages 327–331, May 2010.

[SMR$^+$09] Paolo Santi, Ritesh Maheshwari, Giovanni Resta, Samir Das, and Douglas M. Blough. Wireless link scheduling under a graded SINR interference model. In *Proceedings of the 2nd ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing (FOWANC '09)*, pages 3–12. ACM, 2009.

[SMS06] Gaurav Sharma, Ravi R. Mazumdar, and Ness B. Shroff. On the complexity of scheduling in wireless networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MobiCom '06)*, pages 227–238, New York, NY, USA, 2006. ACM.

[SPS02] Andreas Savvides, Heemin Park, and Mani B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pages 112–121, New York, NY, USA, 2002. ACM.

[SR92] S. Y. Seidel and T. S. Rappaport. 914 MHz path loss prediction models for indoor wireless communications in multifloored buildings. *IEEE Transactions on Antennas and Propagation*, 40(2):207–217, February 1992.

[SR04a] Yi Shang and W. Ruml. Improved MDS-Based Localization. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*, volume 4, pages 2640–2651, March 2004.

[SR04b] M.L. Sichitiu and V. Ramadurai. Localization of Wireless Sensor Networks with a Mobile Beacon. In *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 174–183. IEEE, 2004.

[SRL02] Chris Savarese, Jan Rabaey, and Koen Langendoen. Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks. In *Proceedings of the USENIX Technical Annual Conference*, pages 317–327, June 2002.

[SRZF03] Yi Shang, Wheeler Ruml, Ying Zhang, and Markus P. J. Fromherz. Localization from mere connectivity. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '03)*, pages 201–212, New York, NY, USA, 2003. ACM.

[SSG$^+$08] Olga Saukh, Robert Sauter, Matthias Gauger, Pedro José Marrón, and Kurt Rothermel. On Boundary Recognition without Location Information in Wireless Sensor Networks. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN '08)*, pages 207–218, Washington, DC, USA, 2008. IEEE Computer Society.

[SSGM10] Olga Saukh, Robert Sauter, Matthias Gauger, and Pedro José Marrón. On Boundary Recognition without Location Information in Wireless Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)*, 6(3):1–35, 2010.

[SVG$^+$10] Johannes Schmid, Markus Völker, Tobias Gädeke, Pascal Weber, Wilhelm Stork, and K.D. Müller-Glaser. An Approach to Infrastructure-Independent Person Localization with an IEEE 802.15.4 WSN. In *Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN '10)*, pages 906–914. IEEE Computer Society, 2010.

[SVW11a] D. Schieferdecker, M. Völker, and D. Wagner. Efficient Algorithms for Distributed Detection of Holes and Boundaries in Wireless Networks. Technical Report 2011-08, Karlsruhe Institute of Technology, 2011.

[SVW11b] Dennis Schieferdecker, Markus Völker, and Dorothea Wagner. Efficient Algorithms for Distributed Detection of Holes and Boundaries in Wireless Networks. In *Proceedings of the 10th International Symposium on Experimental Algorithms (SEA '11)*, pages 388–399. Springer, 2011.

[SW06] Stefan Schmid and Roger Wattenhofer. Algorithmic models for sensor networks. In *Proceedings of the 20th International Parallel and Distributed Processing Symposium (IPDPS '06)*, IPDPS'06, Washington, DC, USA, 2006. IEEE Computer Society.

[SW11] Wenbo Shi and V.W.S. Wong. MDS-Based Localization Algorithm for RFID Systems. In *2011 IEEE International Conference on Communications (ICC '11)*, pages 1–6, June 2011.

[TBC11] Paula Tarrío, Ana M. Bernardos, and José R. Casar. Weighted Least Squares Techniques for Improved Received Signal Strength Based Localization. *Sensors*, 11(9):8569–8592, 2011.

[TCPK01] Attila Tanács, Gábor Czédli, Kálmán Palágyi, and Attila Kuba. Affine matching of two sets of points in arbitrary dimensions. *Acta Cyberntica*, 15(1):101–106, August 2001.

[TK75] F. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part II–The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution. *IEEE Transactions on Communications*, 23(12):1417–1433, 1975.

[Tor52] Warren Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):401–419, 1952.

[Tor65] Warren Torgerson. Multidimensional Scaling of Similarity. *Psychometrika*, 30:379–393, 1965.

[UY98] S. Ulukus and R.D. Yates. Stochastic power control for cellular radio systems. *IEEE Transactions on Communications*, 46(6):784–798, June 1998.

[VKW09] Markus Völker, Bastian Katz, and Dorothea Wagner. On the Complexity of Scheduling with Power Control in Geometric SINR. Technical Report 2009-15, Fakultät für Informatik, Universität Karlsruhe, 2009. `http://digbib.ubka.uni-karlsruhe.de/volltexte/1000012389`.

[Völ08] Markus Völker. Scheduling and Topology Control in Wireless Sensor Networks. Master's thesis, Universität Karlsruhe (TH), October 2008.

[vRSWZ05] Pascal von Rickenbach, Stefan Schmid, Roger Wattenhofer, and Aaron Zollinger. A Robust Interference Model for Wireless Ad-Hoc Networks. In *Proceedings of the 5th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad-Hoc and Sensor Networks (WMAN '05)*. IEEE, 2005.

[VSG$^+$12] Markus Völker, Johannes Schmid, Tobias Gädeke, K.D. Müller-Glaser, and Dorothea Wagner. Force-Directed Tracking in Wireless Networks using Signal Strength and Step Recognition. In *Proceedings of the International Conference on Localization and GNSS (ICL-GNSS '12)*. IEEE Computer Society, 2012.

[WB01] Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. *Design*, 7(1):1–16, 2001.

[WCRP03] Kang Wang, Carla F. Chiasserini, Ramesh R. Rao, and John G. Proakis. A Distributed Joint Scheduling and Power Control Algorithm for Multicasting in Wireless Ad Hoc Networks. In *Proceedings of the International Communication Conference (ICC '03)*, pages 725–731. IEEE, 2003.

[WCRP05] Kang Wang, Carla-Fabiana Chiasserini, Ramesh R. Rao, and John G. Proakis. A Joint Solution to Scheduling and Power Control for Multicasting in Wireless Ad Hoc Networks. *Journal on Applied Signal Processing*, 2:144–152, 2005.

[Weg05] Ingo Wegener. *Complexity Theory: Exploring the Limits of Efficient Algorithms.* Springer Berlin / Heidelberg, 2005.

[WGD10] Jing Wang, R. K. Ghosh, and Sajal K. Das. A survey on sensor localization. *Journal of Control Theory and Applications*, 8(1):2–11, 2010.

[WGM06] Yue Wang, Jie Gao, and Joseph S.B. Mitchell. Boundary recognition in sensor networks by topological methods. In *12th Annual International Conference on Mobile Computing and Networking (MobiCom '06)*, pages 122–133, USA, 2006. ACM.

[Win03] Robert Winkler. *An Introduction to Bayesian Inference and Decision.* Probabilistic Publishing, 2nd edition, 2003.

[WLS$^+$07] Hui Wang, H. Lenz, A. Szabo, J. Bamberger, and U.D. Hanebeck. WLAN-Based Pedestrian Tracking Using Particle Filters and Low-Cost MEMS Sensors. In *Proceedings of the 4th Workshop on Positioning, Navigation and Communication (WPNC '07)*, pages 1–7, 2007.

[WW07] Dorothea Wagner and Roger Wattenhofer. *Algorithms for Sensor and Ad Hoc Networks: Advanced Lectures.* Lecture notes in computer science: Tutorial. Springer, 2007.

[XCZ08] Bin Xiao, Hekang Chen, and Shuigeng Zhou. Distributed Localization Using a Moving Beacon in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(5):587–600, 2008.

[Yat95] R.D. Yates. A framework for uplink power control in cellular radio systems. *IEEE Journal on Selected Areas in Communications*, 13(7):1341–1347, September 1995.

[YJGJ10] Jaegeol Yim, Seunghwan Jeong, Kiyoung Gwon, and Jaehun Joo. Improvement of Kalman filters for WLAN based indoor tracking. *Expert Systems with Applications*, 37(1):426–433, 2010.

[YW08] Gwo-Jong Yu and Shao-Chun Wang. A Hierarchical MDS-Based Localization Algorithm for Wireless Sensor Networks. In *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications (AINA '08)*, pages 748–754, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[YYAS03] Moustafa Youssef, Moustafa Youssef, Ashok Agrawala, and A. Udaya Shankar. WLAN Location Determination via Clustering and Probability Distributions. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom '03)*, 2003.

[YYN08] Jie Yin, Qiang Yang, and L.M. Ni. Learning Adaptive Temporal Radio Maps for Signal-Strength-Based Location Estimation. *IEEE Transactions on Mobile Computing*, 7(7):869–883, July 2008.

[Zan92a] J. Zander. Distributed cochannel interference control in cellular radio systems. *IEEE Transactions on Vehicular Technology*, 41(3):305–311, August 1992.

[Zan92b] J. Zander. Performance of optimum transmitter power control in cellular radio systems. *IEEE Transactions on Vehicular Technology*, 41(1):57–62, 1992.

[ZJ09] Jun Zheng and Abbas Jamalipour. *Wireless Sensor Networks: A Networking Perspective.* John Wiley & Sons, 2009.

[ZL11] Yifeng Zhou and Louise Lamont. A Mobile Beacon Based Localization Approach for Wireless Sensor Network Applications. In *Proceedings of the 5th International Conference on Sensor Technologies and Applications*, pages 243–248, 2011.

Index