Deployment Strategies for Target Monitoring and Coverage Improvement in Mobile Sensor Networks

Hamid Mahboubi

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montréal, Québec, Canada

September 2013

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis proposal prepared

By: **Hamid Mahboubi**

Entitled: **Deployment Strategies for Target Monitoring and Coverage Improvement in Mobile Sensor Networks**

and submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Dr. R. Sedaghati, Chair

_____ Dr. J. Gu, External Examiner

_____ Dr. W. Xie, External to Program

_____ Dr. S. Hashtrudi Zad, Examiner

_____ Dr. M. R. Soleymani, Examiner

_____ Dr. A. G. Aghdam, Thesis Supervisor

Approved by _____
Chair of the Department of Electrical and Computer Engineering

_____
Dean of the Faculty of Engineering and Computer Science

ABSTRACT

Deployment Strategies for Target Monitoring and Coverage Improvement in Mobile Sensor Networks

Hamid Mahboubi,

Concordia Unviersity, 2013

Efficient sensor deployment strategies are developed in this work for target monitoring and coverage improvement in collaborative wireless mobile sensor networks. The objective of the target monitoring problem is to compute the desired sensing and communication radii of sensors as well as their location at every time instant such that a set of prescribed specifications such as connectivity preservation and low energy consumption are satisfied. An energy-efficient strategy is also proposed for tracking a moving target in a sensing field, using a grid of sufficiently small rectangular cells. The grid is converted to a graph with properly weighted edges. A shortest-path algorithm is subsequently used to route information from target to destination by a subset of sensors. In the problem of coverage improvement in mobile sensor networks, on the other hand, the objective is to place each sensor in the field using available local information about its neighbors in such a way that the area covered by sensors is as large as possible, while some important criteria are taken into consideration. Both cases of identical and nonidentical sensors (in terms of sensing radii) are considered, and different iterative algorithms are developed which are shown to be convergent. The relocation algorithms are based on the relative position of each sensor w.r.t. the boundaries of its cell or the corresponding corner point. The algorithms are extended to the case of limited communication range of sensors (leading to inaccurate Voronoi cells), an environment with prioritized sensing (mathematically characterized by a weighting function for different points), and an environment with obstacles (leading to some invisible areas). Simulation results are provided to validate the effectiveness of the proposed algorithms.

To my parents

for their love and sacrifice

ACKNOWLEDGEMENTS

I am indebted to many people for making the time working on my Ph.D. an unforgettable experience. My appreciation first goes to my supervisor, Prof. Amir G. Aghdam who has oriented and supported me with care and patience from the preliminary to the concluding stages of my Ph.D. studies. I am deeply grateful for his encouragement, motivation, experience, and immense knowledge, which makes him an exceptional supervisor.

I would also like to thank Dr. Kamran Sayrafian-Pour for very helpful discussions and insightful suggestions throughout this thesis. My gratitude extends to my colleagues at Concordia University, who provided me with some fruitful discussions. In particular, I would like to thank Ahamdreza Momeni, Kaveh Moezzi Madani, Walid Masoudi, Jalal Habibi and Farid Sharifi with whom I co-authored a number of papers.

I ultimately would like to thank my beloved parents and sisters for their love and warm support.

I would like to acknowledge the financial support from the National Institute of Standards and Technology (NIST), and also from the Natural Sciences and Engineering Research Council of Canada (NSERC) throughout the course of this study.

TABLE OF CONTENTS

List of Figures

Chapter 1

Introduction

Sensor networks have been envisioned as a means for gathering, monitoring, processing and delivering information about the physical environment to the intended recipient(s) [1], [2]. This area of research has attracted much attention in both control and communication literature in recent years [3], [4], [5], [6]. A mobile sensor network (MSN) is typically comprised of wireless mobile nodes equipped with battery-powered sensors. Such networks are known to be very effective in detecting, monitoring, and tracking dynamic targets, and have important civilian and military applications [7], [8], [9], [10]. Examples of such applications include robot-assisted sensor networks for data collection [11], security and surveillance [12], [13], [14], environmental monitoring [15], [16], [17], target tracking [18], [19], and structural health monitoring (SHM) [20], [21], to name only a few.

In an MSN, each sensor communicates with a subset of sensors in the network, and uses a proper movement strategy in order to achieve certain objectives such as covering a sensing field, monitoring or tracking a moving target with a trajectory which is not known *a priori*. The information exchange between the sensors and a proper algorithm to use the collected information in order to effectively relocate the mobile sensors are the two important components of any MSN control scheme.

These components along with the capabilities of the individual sensors (in terms of battery power, communication range and displacement flexibility) determine the efficacy of the MSN in achieving any desired objective [22].

Recent developments in MEMS technology have provided a wealth of cheap, customizable, and embedded ad hoc wireless sensor systems [23], [24], [25]. There has been a burst of research activities in cross-layer network optimization in recent years, involving routing, flow and power control, and packet scheduling [26], [27]. The mathematical framework for such an optimization problem is based on the concept of elastic users and the corresponding aggregate utility maximization; for instance, see the framework given in [28] in the context of network management. Price-based distributed algorithms concerning utility maximization for a wire-line network were developed in [29]. These algorithms assume that elastic users respond to congestion pricing signals by modifying their bandwidth requirements. More recent papers such as [30], [31], extended the price-based algorithms to a wireless environment. Note that wireless networks have numerous advantages in sensor applications, due mainly to the distributed nature of this type of system.

Target tracking is one of the most important problems concerning mobile sensor networks [32]. In this type of problems, it is desired to track a moving target by properly moving some or all of the sensors in the field to create a route from the target to destination, where the network information is collected. There has been considerable progress recently in developing efficient deployment algorithms for mobile sensor networks [33], [34]. On the other hand, communication, sensing and movement are sources of energy consumption in mobile sensors [7]. Hence, limited energy resources of the sensors need to be taken into consideration in designing sensor deployment algorithms in real-world applications. Furthermore, due to the distributed structure of the network, a decentralized decision-making configuration is often more desirable. It is important to note that a strategy which takes these

limitations into account and achieves the design specifications does not necessarily exist.

The problem of collaborative tracking of mobile nodes in wireless sensor networks is studied in [35]. It combines target tracking and node selection procedures to identify the effective sensors for an energy efficient strategy. In [36], an algorithm is provided to estimate the position of the target, while optimizing the quantization level for the minimum transmission power. A distributed energy optimization technique is proposed in [18] for target tracking in wireless sensor networks. Sensor nodes are clustered properly, and the sensing area is partitioned for parallel sensor deployment optimization.

The other main area of interest in wireless sensor networks is concerned with the development of efficient sensor deployment strategies to improve both coverage and resource management in the network [37]. There are a number of practical constraints which need to be taken into account in designing control algorithms for sensor networks. For instance, in many real-world applications no *a priori* knowledge is available about the initial position of the sensors. Furthermore, it is often desirable to have some form of decentralization due to the distributed nature of the system. In other words, each sensor is required to make a decision based on its limited communication and sensing capabilities, as well as its limited knowledge obtained from other sensors [38].

The Voronoi diagram is often used for coverage analysis in sensor networks. In [39], a Voronoi-based technique is proposed to improve coverage in a sensor network with no requirement of global location assurance condition for the sensors. Distributed gradient-descent algorithms are given in [40] to increase sensing coverage using the Delaunay graph. A class of aggregate objective functions is studied in [38] based on the geometry of the Voronoi cells and proximity graphs. In [41], an algorithm is developed for efficient sensor deployment and power assignment in

a sensor network. To this end, a multi-objective optimization problem is introduced which is reformulated as a group of single-objective scalar problems.

Non-smooth gradient flows are used in [42] to develop distributed control strategies for the problems of disk covering and sphere packing. In [43], the problem of positioning a group of sensors in a region for detection purposes is investigated by minimizing the maximum probability of non-detection. A decentralized adaptive control law is developed in [44] to properly place a group of sensors in an environment for optimal sensing coverage. Distributed control strategies are proposed in [45] to obtain a convex equi-partition configuration in an MSN. Effective deployment strategies are subsequently developed to improve the sensing coverage. The problem of covering an environment using a network of mobile robots with different sensor footprints is considered in [46]. An efficient deployment algorithm is proposed in [47] which finds the appropriate locations for the mobile sensors by minimizing the maximum error variance and extended prediction variance. In [48]-[52], efficient coverage strategies are developed which do not use simple sensing models or Voronoi partitions. Distributed gradient-based techniques are presented in [48], [49] for optimal coverage in an MSN. To this end, the sensors cooperatively optimize a probabilistic detection metric, as opposed to a simple geometric area metric. Distributed control strategies are introduced in [50] for optimal coverage in an environment with obstacles, where the sensors' field-of-view is limited. In [51], distributed convergence to a Nash equilibrium in an MSN is investigated. A coverage algorithm is provided in [52] for maximizing the probability of detection, where the communications cost is minimized in order to increase the network lifetime.

In [53], a coverage inference protocol is presented which can provide an accurate measurement of the connected coverage for the base station in an energy-efficient manner. Some of the important characteristics of the protocol include efficient routing, spatial aggregation, sleeping scheduling and topology control. The concept of

desired sensing coverage (DSC) is introduced in [54], and an energy-efficient scheme is developed to meet the desired DSC. To this end, a number of sensors are selected using the geometric probability theory and a randomized technique to operate as "on-duty data reporters". An algorithm is subsequently presented to transmit the collected data to the sink node by constructing a data gathering tree for each group of selected sensor. The authors in [55] propose a new type of network called partitioned synchronous network to address the coverage and connectivity problems at the same time. The work [55] is concerned with the detection of stochastic events for which the conventional definition of coverage is not applicable. In [56], a randomized scheduling algorithm is studied under quality of service (QoS) constraints such as network coverage intensity, detection probability, and bounded detection delay. The problem of network lifetime maximization is then analyzed under these constraints. The coverage property of clustered wireless sensor networks is studied in [57] and a foundation is provided to optimize the performance of the network. It is shown that the connectivity of the network changes by increasing the vacancy in random placement of sensors in a wireless sensor network. Furthermore, the probability of coverage in the network is determined by analyzing various levels of redundancy. The authors in [58] exploit the temporal and spatial correlation among the data sensed by different sensors and leverage prediction to maximize the network lifetime. The concept of entropy is then adopted to evaluate the information uncertainty concerning the sensing field. The problem is formulated as a minimum weight submodular set cover problem, and an efficient centralized truncated greedy algorithm is presented to solve it. The optimal deployment of sensors for achieving a full coverage and four-connectivity in a WSN is investigated in [59], and two new patterns called Diamond pattern and Double-strip pattern are introduced accordingly. A centralized heuristic technique is proposed in [60] to maximize the spatial-temporal coverage by scheduling sensors' activities after their deployment.

Also, a distributed parallel optimization protocol is given which converges to a local optimum.

All of the papers cited in the previous paragraph study static wireless sensor networks. Note that mobility is an inherent property of many real-world WSNs [61], and also it could enhance the coverage performance [62], [63]. In addition, in many practical settings sensors cannot be relocated manually (for example in disaster areas, toxic urban regions and remote harsh fields). In these cases, sensors are sometimes dropped from an aircraft over the field. Thus, it is important that the sensors move to appropriate positions in order to achieve the desired objective [64], [65].

Most of the existing sensor movement strategies in the literature use one of the following three types of techniques: coverage pattern [66], [67], [68], grid architecture [69], [70], and virtual forces [64], [71]. In [66], the so-called "adaptive triangular deployment (ATRI) algorithm" is proposed for maximizing coverage area and minimizing coverage gaps in an unattended mobile sensor network by adjusting the deployment layout of the nodes such that it becomes as close as possible to equilateral triangulation, which is shown to be the optimal layout for maximizing the no-gap coverage. Two related deployment problems, namely sensor dispatch and sensor placement, are investigated in [67], and the proposed solution can be applied to any arbitrary polygon-shape field in the presence of obstacles. In [68], an efficient obstacle-resistant deployment algorithm is provided which deploys a near-optimal number of sensors over the sensing field to achieve full coverage while avoiding obstacles. The algorithm integrates the deployment policy, the obstacle-resistant rules, the boundary handling rule, and serpentine movement to achieve the objective. In [69], a framework consisting of a generic system model and a generic objective function is given. A generic method based on bipartite matching is subsequently proposed to redeploy the mobile sensors and solve a problem with various design

objectives in a WSN. To this end, the sensing field is partitioned into a number of grids, and the difference between the number of sensors in the grid and the desired number of sensors is defined as the gap of each grid. The problem is then formulated as an optimization problem to minimize the sum of gaps as well as the movement cost of all sensors. A centralized algorithm is proposed in [70] to minimize the total moving distance of sensors for covering a sensing field. Various strategies are then proposed to achieve a balanced state by using scan and dimension exchange. The authors in [71] propose a virtual centripetal force-based algorithm for improving area coverage. The method aligns the direction of every sensor properly to decrease the coverage overlap according to the virtual centripetal force theory. Moreover, in order to maximize the network lifetime, the redundant sensors are shut off in certain time intervals. In [64], three algorithm, namely vector-based (VEC), Voronoi-based (VOR), and Minimax are proposed to determine the final destination of each sensor in the network, such that the coverage increases. In [64], [65], two different approaches called basic protocol and virtual movement protocol are introduced to place the sensors in appropriate positions in order to improve network coverage. Note that in the papers [62]-[71] cited above, it is assumed that there is no limit on the sensors' movement.

The authors in [72] investigate the coverage problem in a mobile sensor network for the case where each mobile sensor cannot move longer than a certain distance because of hardware limitations. The problem of determining a movement plan for sensors with limited mobility in a field clustered into multiple regions is studied in [73], where it is desired to minimize the variance in the number of sensors in different regions and also to minimize the movement of sensors. Two algorithms (one centralized and one distributed) are subsequently proposed to solve the problem. A network of sensors with limited mobility is investigated in [74], where the mobility of each sensor is restricted to a flip (hop). A minimum-cost maximum-flow solution is

proposed in [74] which maximizes the sensing coverage while minimizing the number of flips.

The papers [53]-[74] addressed above and most of the results reported in the literature assume that the sensing and communication capabilities of all sensors are identical. There are several applications, however, where the sensors have different sensing ranges. For example, when sensors from different manufacturers are utilized in the network, their sensing ranges are likely to be different. Furthermore, the variation of the sensing range of a sensor with time is unavoidable [75]. Note that the energy consumption due to sensing is proportional to the square of the sensing range of the sensor [75]. Also, the remaining energy of different sensors will not be the same, in general, after operating for a long period of time. Hence, it is desirable that sensors adjust their sensing ranges based on their remaining energies such that the sensing range of a sensor with lower energy is smaller than that of a sensor with higher energy. This sensing range adjustment can increase the lifetime of the network, and will obviously result in a heterogeneous sensor network. It is known that deployment and topology control in heterogeneous wireless sensor networks is more complex than that in homogeneous wireless sensor networks. However, with a proper degree of heterogeneity in terms of the number of low-end sensors (which have limited computation capability and lower communication and sensing ranges) and high-end sensors, one can address the trade-off between the performance and cost efficiency of the network (for example, see [75], [76]). In what follows, some related results on heterogeneous sensor networks are reviewed briefly.

Energy-efficient coverage algorithms are developed in [75] for a network of heterogeneous mobile sensors for both uniform and Poisson sensor deployment schemes. The notion of the equivalent sensing radius (ESR) is defined first, and then the necessary and sufficient conditions on ESR are obtained for achieving full coverage. Since in the above work the sensors are assumed to move randomly, there is no need

for communication between them. In [77], the problem of partitioning the sensing field in heterogeneous sensor networks is investigated. Although the partitioning introduced in [77] has some drawbacks compared to the MW-Voronoi diagram used in the present work, it addresses some of the shortcomings of conventional Voronoi partitioning (which is used for a network of identical sensors). A general integer linear programming formulation is proposed in [78] to minimize sensor deployment cost in a heterogeneous network composed of different types of sensors with different cost and sensing radii. In [78], the sensing field is mapped to a 2D or 3D grid, with the grid points representing the locations of the sensors. Then, a greedy algorithm is developed to place at least one sensor at each grid point. A novel algorithm is developed in [79] to prolong the lifetime of a heterogeneous WSN composed of two types of sensors: sensors with fixed sensing ranges and sensors with variable sensing ranges. The algorithm uses Voronoi-Laguerre diagram and reduces sensor coverage redundancy by joint sensor activation and sensing radius adaptation. The problem of minimum connected k-coverage is studied in [80] for both homogeneous and heterogeneous WSNs, where it is desired to minimize the number of sensors required to maintain network connectivity in such a way that every point in the field is sensed by at least k active nodes. The method provided in [80] consists of two steps: sensing range slicing and active node scheduling. In the slicing phase, the sensing range of each sensor is decomposed into smaller subregions that are guaranteed to be k-covered. In the scheduling phase, it is specified which sensors should become active, and at what time instants. Two protocols called self-scheduling driven k-coverage and triggered-scheduling driven k-coverage are also developed in [80]. The problem of relay node placement in a heterogeneous WSN composed of sensors, relays and base stations is investigated in [81]. A two-phase approach is also developed in [81] for the case where all nodes are energy limited. The problem of energy-efficient

m-coverage and n-connectivity under border effects in heterogeneous WSNs is investigated in [37], where a location-independent energy-efficient routing algorithm is proposed to maintain the network n-connectivity and m-coverage ratio simultaneously. The authors in [82] develop an ant colony optimization (ACO) based approach to maximize the lifetime of heterogeneous WSNs. The approach finds the minimum number of connected covers for maximizing the network lifetime. A two-phase distributed algorithm is developed in [83] to find the minimum set of sensors that form a connected network and also cover the query region. The sensors are assumed to be heterogeneous in terms of sensing range, communication range and also energy levels. In the first phase of the algorithm, each sensor establishes a connection with its neighbors, and in the second phase the coverage strategy of each sensor is specified according to its minimum-weight coverage cost. In [84], heterogeneous two-tier WSNs are studied, where the concepts of optimization theory and coverage are used to maximize both network lifetime and coverage. In such networks, it is assumed that one tier of nodes is more robust and computationally intensive than the other tier. The effect of random sensors' locations on the coverage and lifetime of heterogeneous sensor networks is investigated in [85]. It is assumed that there are two types of nodes in the network: ordinary and powerful. The powerful nodes can transmit their information to the sink directly, while the ordinary nodes transmit their information to the sink only through powerful nodes. It is shown that the network coverage at any time instant is a function of the initial coverage and the density of cluster heads. The authors in [86] propose two distributed algorithms to find the minimum number of heterogeneous sensors that need to be connected in order to cover the perimeter of the queried region. In the first algorithm, a sensor near the center of region is chosen as the coordinator which selects the sensors that cover the perimeter of the region. The second algorithm consists of two phases. In the first phase, each sensor collects the information of

its neighbors, and in the second phase it selects the minimum number of sensors that cover the region circumference. In [87], optimal solutions to the sleep-wake-up problems are proposed for the model of barrier coverage. The proposed solutions work for both homogeneous lifetime case (when all sensors have identical lifetime) and heterogeneous lifetime case (when sensors have different lifetime). The strategies are also effective for heterogeneous sensing regions. A large number of sensors with adjustable sensing ranges are considered in [88], where it is desired to maintain certain level of coverage by activating a small number of sensors and consuming a small amount of energy. A coverage control scheme is developed in [88], which is a non-dominated sorting genetic algorithm. Furthermore, a binary coding scheme is provided for representing sensor selection and sensing range adjustment. Several localized sensor area coverage protocols for a network of heterogeneous sensors with arbitrary sensing and communication ranges are developed in [89]. Since the prior knowledge of the neighbors is not required in [89], the communication overhead of the proposed approaches is relatively small. First, each sensor chooses a random time out and receives the information of other sensors in this time out. If the sensing area of a sensor is not covered completely or it is covered by a disconnected set of active sensors, then this sensor remains active and sends an activity message. Any sensor whose area is fully covered, on the other hand, sleeps with or without informing the neighbors about its status.

In Chapter 2, a routing strategy is presented for the relocation of mobile sensors in a network and the adjustment of their communication and sensing range, such that a certain cost function is minimized, while the end-to-end connectivity from a moving target to a fixed access point (also called the destination point) is maintained. Various cost functions concerning individual sensors and the entire network will be considered to evaluate the performance of the network in terms of power consumption. A technique is also provided to maximize the durability

of the whole network by monitoring the residual energy of individual sensors and adjusting their parameters accordingly. Simulation results elucidate the desirable characteristics of the proposed methods.

In Chapter 3, an energy-efficient routing technique is introduced to track a target in a mobile sensor network, while optimizing the energy consumption. It is assumed that the main sources of energy consumption in the network are communication, sensing, and movement. The field is first divided into a grid, and then each sensor is directed to a proper node in the grid. A graph is subsequently derived from this grid, and its edges are weighted properly based on the parameters of the energy consumption model. This graph is used to find the optimal route to transfer the information from the target to destination. Then, the graph is redrawn in such a way that the minimum energy problem is translated to the constrained shortest path from the target to destination. This is a well-known problem in network and routing, and several algorithms exist in the literature to handle it. Due to the simplicity and effectiveness of Dijkstra's algorithm it will be adopted in this chapter to solve the underlying problem.

In Chapter 4, new techniques are introduced to improve network coverage more efficiently. The proposed methods are mainly concerned with the distances of each sensor and the points inside its corresponding Voronoi polygon from the edges and vertices of the polygon. Four algorithms are developed: Maxmin-vertex, Maxmin-edge, Minmax-edge, and VEDGE algorithm. The main characteristic of these algorithms is that the sensor movement is performed iteratively. Once each destination is computed, the coverage area of the corresponding sensor w.r.t. the new destination inside the preceding constructed Voronoi polygon is compared with its previous local coverage area. If this coverage area is larger than the preceding one, the sensor moves to the new destination; otherwise, it remains in its current location. If, on the other hand, the coverage area inside its Voronoi polygon does not increase

by a certain amount, then the iterations stop. This termination condition guarantees that under the proposed algorithms the sensors arrive at the desired destinations (with a prescribed accuracy which is set by the threshold level) in finite time. Unlike existing coverage algorithms (e.g., the ones proposed in [64]), the proposed strategies are mainly concerned with the distance of each sensor from the edges of its Voronoi polygon (rather than its distance from the vertices of the polygon) in order to compute its next destination. The proposed strategies outperform the existing ones in the case when a sensor is required to move toward a sharp vertex. The VEDGE algorithm is more preferable as far as network coverage is concerned. The Maxmin-edge strategy, on the other hand, is more energy-efficient than the other algorithms when there is a large number of sensors in the network, while Maxmin-vertex strategy is more desirable when there is a relatively small number of sensors. Finally, the Minmax-edge strategy outperforms the other algorithms in terms of deployment time, when there is a relatively large number of sensors in the network.

The objective of Chapter 5 is to develop sensor deployment algorithms in a network of mobile sensors with different sensing capabilities, for effective network coverage. The multiplicatively weighted Voronoi (MW-Voronoi) diagram (where the weight of each sensor is assumed to be equal to its sensing radius) is used to discover coverage holes [90], [91]. Three algorithms are proposed: Weighted Vector Based (WVB), Minmax-curve and Maxmin-curve. The main idea behind these algorithms is to move each sensor iteratively in such a way that its sensing coverage is increased. Similar to the case of identical sensors considered in Chapter 4, once a new location for a sensor is computed, the corresponding coverage area w.r.t. the new location of the sensor inside the previously constructed MW-Voronoi region is compared to the preceding local coverage area. If this coverage area is larger than the preceding one, the sensor moves to the new location; otherwise, it remains in its current position. A pre-specified threshold is used to stop the algorithm when no sensor's coverage

area inside its MW-Voronoi region increases by this amount.

In Chapter 6, three distributed deployment algorithms are presented for a network of nonidentical sensors. The multiplicatively weighted Voronoi (MW-Voronoi) diagram is employed to find the coverage holes, where the weight assigned to each sensor is proportional to its sensing radius. Three proposed algorithms in this work are farthest point boundary (FPB), Maxmin-vertex and Minmax-vertex. These sensor placement algorithms are distributed and perform iteratively. Again, the movement of the sensors to the candidate locations and termination of the algorithm follow the scheme introduced in the previous two chapters (the same movement and termination schemes are used in the next five chapters, and the description of the scheme is not repeated for brevity).

In Chapter 7, new deployment strategies are introduced to increase coverage in a network of mobile sensors. Three algorithms are developed: vertex virtual forces (VVF) algorithm, edge virtual forces (EVF) algorithm, and vertex-edge virtual forces (VEVF) algorithm. The above-mentioned algorithms are then extended to the case of a network with nonidentical sensors using multiplicatively-weighted Voronoi (MW-Voronoi) diagrams. A virtual force is assumed to be applied to each sensor from the vertices and boundaries of the corresponding Voronoi cell, which tend to move the sensor. The movement of each sensor follows the scheme described in the preceding chapters.

In Chapter 8, two different distributed relocation algorithms are presented for a network of mobile sensors with different sensing capabilities and limited communication ranges. It is desired to improve network coverage by moving them to proper locations in the field. To this end, the notion of *limited communication multiplicatively weighted Voronoi* (LCMW-Voronoi) diagram is introduced and its useful characteristics are discussed. This diagram is subsequently used to discover the so called "coverage holes", and properly relocate the sensors to cover them as much

as possible. Two iterative algorithms are introduced where in each step the above-mentioned diagram is used to find a candidate point for each sensor. The sensor movement follows the scheme described before. It is shown that both algorithms are monotonically increasing and convergent.

In Chapter 9, two different distributed algorithms are presented for deployment and relocation of sensors in an MSN. It is assumed that the nodes in the network have different sensing capabilities; and also, they are operating on a flat field containing obstacles. The approach in our proposed algorithms is based on using the *visibility-aware multiplicatively weighted Voronoi* (VMW-Voronoi) diagram. The effect of obstacles on sensors' sensing capabilities (i.e. visibility area of each sensor) is taken into consideration in the proposed algorithms. The VMW-Voronoi diagram is used for discovering coverage holes and sensing radii of sensors are used as node weights in the construction of this diagram. In this work, it is assumed that if the line-of-sight between a point and a sensor is blocked with an obstacle, then the sensor is not able to sense any object located at that point. This is an acceptable assumption and has been used in the literature previously ([92], [93]). Using VMW-Voroni diagram, the following two algorithms are presented in this chapter: *Obstructed Farthest Point* (OFP) and *Obstructed Minmax Point* (OMP) algorithms. By iterative application of these algorithms, a gradual improvement in the overall coverage can be obtained. At each iteration, a new candidate coordinate for sensor relocation is calculated based on the current position of each sensor and its VMW-Voronoi region.

In Chapter 10, new distributed deployment strategies are introduced to increase coverage in a network of mobile sensors with a prescribed priority function for the sensing field. To this end, a priority function is assumed to be given which specifies the coverage priority of different points in the sensing area. The MW-Voronoi diagram is used to partition the sensing field. This partitioning is then used to discover coverage holes in the network and relocate the sensors accordingly

to minimize them, while taking into account the coverage priority of different points in the field. Three algorithms are developed: maximum weighted vertex (MWV), maximum weighted point (MWP), and maximum distance weight (MDW). The main idea behind the proposed algorithms is to move each sensor iteratively in such a way that its weighted coverage increases. Once each sensor's destination is computed, the weighted coverage area w.r.t. the new destination of the corresponding sensor in the previously constructed MW-Voronoi region is compared to its previous local weighted coverage to decide whether the sensor should move or not.

In Chapter 11, new distributed sensor deployment strategies are introduced for a network consisting of both static and mobile sensors. The multiplicatively weighted Voronoi (MW-Voronoi) diagram is utilized to find the coverage holes, where the weight assigned to each mobile sensor is proportional to its sensing radius. In the proposed strategies, namely, farthest weighted vertex (FWV) and Max-area, every static sensor broadcasts its sensing radius and location to all mobile sensors. Each mobile sensor subsequently assigns a proper virtual weight to every point in the field based on the received information. The algorithms are then performed iteratively to compute the candidate location for each mobile sensor.

The results of this dissertation are published (or submitted for publication) in a number of journals and conference proceedings ([94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106]). These publications are listed below for different chapters.

- Chapter 2

    1. H. Mahboubi , A. Momeni, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "An efficient target monitoring scheme with controlled node mobility for sensor networks," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 6, pp. 1522-1532, 2012.

    2. H. Mahboubi , A. Momeni, A. G. Aghdam, K. Sayrafian-Pour, and V.

Marbukh, "Optimal target tracking strategy with controlled node mobility in mobile sensor networks," in *Proceedings of American Control Conference*, 2010, pp. 2921-2928.

- Chapter 3

3. H. Mahboubi, W. Masoudimansour, A. G. Aghdam and K. Sayrafian-Pour, "An energy-efficient target tracking strategy for mobile sensor networks," 2013 (submitted to a journal).

4. H. Mahboubi, W. Masoudimansour, A. G. Aghdam and K. Sayrafian-Pour, "Cost-efficient routing with controlled node mobility in sensor networks," in *Proceedings of IEEE Multiconference on Systems and Control*, 2011, pp. 1238-1243.

- Chapter 4

5. H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour and V. Marbukh, "Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 163-174, 2014.

6. H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour and V. Marbukh, "Distributed deployment algorithms for improved coverage in mobile sensor networks," in *Proceedings of IEEE Multiconference on Systems and Control*, 2011, pp. 1244-1249.

- Chapter 5

7. H. Mahboubi, K. Moezzi, A. G. Aghdam and K. Sayrafian-Pour, "Distributed deployment algorithms for efficient coverage in a network of mobile sensors with nonidentical sensing capabilities," *IEEE Transactions on Vehicular Technology*, 2013 (to appear).

8. H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour and V. Marbukh, "Self-deployment algorithms for coverage problem in a network of mobile sensors with unidentical sensing range," in *Proceedings of IEEE Global Communications Conference*, 2010, pp. 1-6.

9. H. Mahboubi, K. Moezzi, A. G. Aghdam and K. Sayrafian-Pour, "Self-deployment algorithms for field coverage in a network of nonidentical mobile sensors," in *Proceedings of IEEE International Conference on Communications*, 2011, pp. 1-6.

- Chapter 6

10. H. Mahboubi, K. Moezzi, A. G. Aghdam and K. Sayrafian-Pour, "Distributed coverage algorithms for a network of nonidentical mobile sensors," 2013 (submitted to a journal).

11. H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour and V. Marbukh, "Self-deployment algorithms for field coverage in a network of nonidentical mobile sensors: vertex-based approach," in *Proceedings of American Control Conference*, 2011, pp. 3227-3232.

- Chapter 7

12. H. Mahboubi, and A. G. Aghdam, "Distributed deployment algorithms for coverage improvement in a network of wireless mobile sensors: relocation by virtual force," 2013 (submitted to a journal).

13. H. Mahboubi, and A. G. Aghdam, "Distributed deployment strategies to increase coverage in a network of wireless mobile sensors," in *Proceedings of American Control Conference*, 2013, pp. 5907-5912.

- Chapter 8

14. H. Mahboubi, and A. G. Aghdam, "Self-deployment Techniques for Coverage Improvement in a Network of Nonidentical Mobile Sensors with Limited Communication Ranges in the Presence of Obstacles," 2013 (submitted to a journal).

15. H. Mahboubi, and A. G. Aghdam, "Self-deployment algorithms for coverage improvement in a network of nonidentical mobile sensors with limited communication ranges," in *Proceedings of American Control Conference*, 2013, pp. 6898-6903.

- Chapter 9

16. H. Mahboubi, W. Masoudimansour, A. G. Aghdam and K. Sayrafian-Pour, "Self-deployment Algorithms for Coverage Improvement in Mobile Sensor Networks in Presence of Obstacles," 2013 (submitted to a journal).

- Chapter 10

17. H. Mahboubi , J. Habibi, A. G. Aghdam and K. Sayrafian-Pour, "Distributed deployment strategies for improved coverage in a network of mobile sensors with prioritized sensing field," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 451-461, 2013.

18. H. Mahboubi , J. Habibi, A. G. Aghdam and K. Sayrafian-Pour, "Cooperative self-deployment strategies in a mobile sensor network with prioritized coverage plan," in *Proceedings of IEEE Global Communications Conference*, 2011, pp. 1-6.

- Chapter 11

19. H. Mahboubi , J. Habibi, A. G. Aghdam and K. Sayrafian-Pour, "Distributed deployment algorithms for efficient coverage in a network of static and mobile sensors," 2013 (submitted to a journal).

20. H. Mahboubi , J. Habibi, A. G. Aghdam and K. Sayrafian-Pour, "Distributed coverage optimization in a network of static and mobile sensors," in *Proceedings of American Control Conference*, 2013, pp. 6893-6897.

The author has also published (or submitted) the following relevant papers ([107, 108, 109, 110] in the references).

21. H. Mahboubi, W. Masoudimansour, A. G. Aghdam and K. Sayrafian-Pour, "Maximum Lifetime Strategy for Target Tracking with Controlled Node Mobility in Sensor Networks with Obstacles," 2013 (submitted to a journal).

22. H. Mahboubi, F. Sharifi, A. G. Aghdam and Y. M. Zhang, "Distributed coordination of a network of agents with limited communication range in the presence of obstacles," 2013 (submitted to a journal).

23. H. Mahboubi, A. G. Aghdam and K. Sayrafian-Pour, "A joint relocation and sensing range control algorithm to improve coverage and lifetime in mobile sensor networks," 2013 (submitted to a journal).

24. H. Mahboubi, W. Masoudimansour, A. G. Aghdam and K. Sayrafian-Pour, "Maximum life span strategy for target tracking in mobile sensor networks," in *Proceedings of American Control Conference*, 2012, pp. 5096-5101.

25. H. Mahboubi, F. Sharifi, A. G. Aghdam and Y. M. Zhang, "Distributed coordination of multi-agent systems for coverage problem in presence of obstacles," in *Proceedings of American Control Conference*, 2012, pp. 5252-5257.

26. H. Mahboubi, F. Sharifi, A. G. Aghdam and Y. M. Zhang, "Distributed coordination of a network of nonidentical agents with limited communication capabilities in the presence of fixed obstacles," in *Proceedings of American Control Conference*, 2013, pp. 5418-5423.

27. H. Mahboubi, and A. G. Aghdam, "An energy-efficient strategy to improve coverage in a network of wireless mobile sensors with nonidentical sensing ranges," in *Proceedings of IEEE 77th Vehicular Technology Conference*, 2013, pp. 1-5.

Finally, the author collaborated with other researchers on the following relevant papers during the course of this study ([111, 112, 113, 114, 115, 116, 117] in the references).

28. J. Habibi, H. Mahboubi and A. G. Aghdam, "A gradient-based coverage optimization strategy for mobile sensor networks," 2012 (submitted to a journal).

29. F. Sharifi, A. Chamseddine, H. Mahboubi, A. G. Aghdam and Y. M. Zhang, "A distributed deployment strategy for a network of cooperative autonomous vehicles," 2012 (submitted to a journal).

30. J. Habibi, H. Mahboubi and A. G. Aghdam, "Distributed coverage control of mobile sensor networks subject to measurement error," 2013 (submitted to a journal).

31. W. Masoudimansour, H. Mahboubi, A. G. Aghdam and K. Sayrafian-Pour, "Maximum lifetime strategy for target monitoring in a mobile sensor network with obstacles," in *Proceedings of the 51th IEEE Conference on Decision and Control*, 2012, pp. 1404-1410.

32. J. Habibi, H. Mahboubi and A. G. Aghdam, "A nonlinear optimization approach to coverage problem in mobile sensor networks," in *Proceedings of 50th*

IEEE Conference on Decision and Control, 2011, pp. 7255-7261.

33. J. Habibi, H. Mahboubi and A. G. Aghdam, "Distributed coverage optimization in a network of mobile agents subject to measurement error," in *Proceedings of American Control Conference*, 2012, pp. 4510-4515.

34. M. Tousi, A. Ajorlou, H. Mahboubi, and A. G. Aghdam, "Decentralized pole-placement using generalized sampled-data hold functions," in *Proceedings of the 51st IEEE Conference on Decision and Control*, 2012, pp. 6921-6925.

35. V. Marbukh, K. Sayrafian-pour, H. Mahboubi, A. Momeni, and A. G. Aghdam, "Cooperative sensor relocation in a mobile sensor network by distributed subgradient algorithm," in *Proceedings of International Conference on Sensor Technologies and Applications*, 2011, pp. 91-96.

36. V. Marbukh, K. Sayrafian-Pour, H. Mahboubi, A. Momeni, and A.G. Aghdam, "Towards evolutionary-pricing framework for mobile sensor network self-organization," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2010, pp. 4394-4401.

37. F. Sharifi, Y. M. Zhang, H. Mahboubi, and A. G. Aghdam, "Coverage control in multi-agent systems subject to communication delays," in *Proceedings of IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications*, 2012, pp. 267-274.

38. F. Sharifi, H. Mahboubi, A. G. Aghdam and Y. M. Zhang, "A Coverage strategy with guaranteed collision avoidance in multi-agent systems using navigation functions," *in Proceedings of AIAA Guidance, Navigation, and Control Conference*, 2013, pp. 1-8.

Chapter 2

An Efficient Target Monitoring Scheme with Controlled Node Mobility for Sensor Networks

This chapter is concerned with target monitoring using a network of collaborative mobile sensors. The objective is to compute (online) the desired sensing and communication radii of sensors as well as their location at each time instant, such that a set of prescribed specifications are met. These specifications include end-to-end connectivity preservation from the target to a fixed destination, while durability of sensors is maximized and the overall energy consumption is minimized. The problem is formulated as a constrained optimization, and a procedure is presented to solve it. Simulation results demonstrate the effectiveness of the proposed techniques.

The plan of the rest of this chapter is as follows. The problem is formulated in Section 2.1, and some important assumptions and definitions are also provided which will be used later to develop the main results. An algorithm and some important theorems and lemmas are presented in Section 2.2, as the main contributions of the chapter for solving the corresponding constrained and unconstrained optimization

problems. Finally, simulations are given in Section 2.3.

2.1 Problem Formulation

Consider a group of n mobile sensors, each one representing a node in the sensor network, and let the coordinates of sensor i, $i \in \mathbf{n} := \{1, \ldots, n\}$, be denoted by x_i. Consider also a moving target and a fixed access point (also referred to as the destination point), where for convenience of notation they are labeled as nodes 0 and $n+1$, respectively, and their coordinates are represented by x_0 and x_{n+1}, accordingly. In order to ensure target monitoring at all times, it is required to maintain connectivity from the target to the access point continuously (in terms of sensing and communication). Furthermore, in order to accomplish the mission in the most efficient manner, it is desired that the *routing cost* defined as the sum of the costs (associated with the sensing and transmission power needed to communicate over the link) of any sensor involved in establishing a connected link from the target to the destination point is minimized.

One of the most desirable control objectives in MSNs is energy efficiency [22], [64]. In general, energy consumption of mobile sensors is due to communication, sensing, and movement. The optimal energy-efficient control action depends on which one of the above-mentioned energy-consuming factors is dominant [64], [118].

In order to formulate the optimization problem, it is assumed that a link $l = (i, j)$ from node $i \in \mathbf{n} \cup \{0\}$ to node $j \in \mathbf{n} \cup \{n+1\} \backslash \{i\}$ exists if and only if the corresponding signal-to-interference ratio SIR exceeds certain (strictly positive) threshold χ. This can be mathematically expressed as:

$$SIR_{ij} = \frac{P_{ij}\xi_{ij}}{\eta_j + \sum_{(n,k)\neq(i,j),n\neq i,j} P_{nk}\xi_{nj}} > \chi \tag{2.1}$$

where P_{ij} is the power required to transmit information from node i to node j, ξ_{ij} is the *path loss* (which is defined to be the reduction in the signal power as it travels

from transmitter to the receiver) from node i to node j, $\sum_{(n,k)\neq(i,j),n\neq i,j} P_{nk}\xi_{nj}$ is the overall interference power, and η_j is the noise power at node j. For simplicity, assume that the interference power is negligible, and that the noise power η_j is equal to 1, for all $j \in \mathbf{n}$. Then, using (2.1) the following minimum power consumption by node i is obtained for direct communication with node j:

$$P_{ij} = \frac{\chi}{\xi_{ij}}$$

The path loss is inversely proportional to some power of the distance d_{ij} between nodes i and j, i.e. d_{ij}^{λ}, for all $i, j \in \mathbf{n}$, $i \neq j$. The power λ is typically between 2 and 4, and is closer to 4 for low-lying antenna and near-ground channels, as in most sensor network applications [7], [119]. The communication radius of sensor i at the instant t, denoted by $R_{ci}(t)$, is equal to the radius of the largest circle around x_i, such that the corresponding SIR from x_i to any point inside the circle is greater than the threshold χ. Here, it is assumed that the coverage area of each sensor can be described by a disk of radius $R_{ci}(t)$. The power required for sensing, on the other hand, is typically greater than the power required for communication from the same distance. This power is also proportional to d^{γ}, where d is distance and $\gamma \geq 2$; in particular, in a radar system, $\gamma \geq 4$.

A power consumption model for sensor movement is given by $P_m(d_m) = kd_m$, where P_m is the power consumption due to movement, k is a known constant, and d_m is the traveling distance of the sensor [120]. While this is a realistic model in many practical settings (e.g., when the sensors are mounted on mini-wheel robots), for the case when the sensors move on a surface with approximately pure rolling [121], the energy consumption due to movement is small. As mentioned earlier, the power consumption due to communication over a distance d_c is proportional to d_c^{λ}. Consider the case where the target is located far from the destination and communication range of sensors is relatively large such that d_c is much greater than the typical moving distance d_m. In such cases, the power consumption due

to communication can be significantly higher than the power consumption due to movement, for large values of λ. A similar argument can be made for the case where the target motion is relatively slow with long pauses.

Based on the above discussion, the following assumption is made.

Assumption 2.1. *It is assumed in this chapter that the power consumption of sensors due to movement is negligible compared to that due to communication and sensing.*

Assumption 2.2. *In this chapter, it is assumed that one sensor is assigned to sense the target at any given time. This sensor will hereafter be referred to as the* monitoring sensor*. The information is transmitted from the monitoring sensor to the fixed destination point through a subset of the sensors in a collaborative fashion. The monitoring sensor is not fixed in general, and can be changed from time to time depending on the position of the sensors and their residual energies.*

Let the monitoring sensor be labeled as sensor 1 throughout the chapter. The sensing radius of this sensor at the time instant t, denoted by $R_{s1}(t)$, is defined as the radius of the largest circle around x_1, such that this sensor can sense the target anywhere inside the circle. Note that sensor i can transmit the information to sensor j at the instant t if and only if $R_{ci}(t) \geq d_{ij}(t)$, for all $i, j \in \mathbf{n}$. Note also that sensor 1 can sense the target at the instant t if and only if $R_{s1}(t) \geq d_{10}(t)$, where d_{10} denotes the distance between sensor 1 and the target.

Assumption 2.3. *It is assumed that the target is at a reachable distance from the destination point through other sensors at all times, i.e.* $x(t) \leq nR_{c,\max} + R_{s,\max}$*, where* $R_{c,\max}$ *is the maximum communication radius that can be covered by every sensor,* $R_{s,\max}$ *is the maximum sensing radius that can be detected by sensor 1, and* $x(t)$ *is the distance between the target and destination.*

Recall that the required powers for sensor i ($i \in \mathbf{n}$) to communicate information and for sensor 1 to detect information are proportional to R_{ci}^{λ} and R_{s1}^{γ}, respectively. On the other hand, by assumption the movement power is negligible compared to the above-mentioned communication and sensing powers. Thus, for any $i, j \in \{2, \ldots, n\}$:

$$\frac{\text{power consumption of sensor } i}{\text{power consumption of sensor } j} = \frac{R_{ci}^{\lambda}}{R_{cj}^{\lambda}} \tag{2.2}$$

and the following cost function (which reflects the overall instantaneous power consumed by all sensors) is to be minimized at any time $t > 0$:

$$J_P(t) = \alpha R_{s1}^{\gamma}(t) + \sum_{i \in \mathbf{k}(t)} R_{ci}^{\lambda}(t) \tag{2.3}$$

subject to the condition $R_{s1}(t) + \sum_{i \in \mathbf{k}(t)} R_{ci}(t) \geq x(t)$ for some set $\mathbf{k}(t) \subset \mathbf{n}$. Furthermore, the constraints given below need to be satisfied for all $t > 0$:

i) $0 \leq R_{ci}(t) \leq R_{c,\max} \quad \forall i \in \mathbf{k}(t)$

ii) $0 \leq R_{s1}(t) \leq R_{s,\max}$

where α is a constant coefficient used to normalize the sensing power with respect to the communication power.

While minimizing power consumption is of great importance in MSNs, in many applications it is more desirable that the sensor with the smallest residual energy consume the smallest amount of power at each instant, in order to maximize the life-span of the sensors, which in turn maximizes the durability of the entire network. In such cases, the following performance index is used instead of (2.3):

$$J_D(t) = \alpha \theta_{s1}(t) R_{s1}^{\gamma}(t) + \sum_{i \in \mathbf{k}(t)} \theta_i(t) R_{ci}^{\lambda}(t) \tag{2.4}$$

subject to the condition $R_{s1}(t) + \sum_{i \in \mathbf{k}(t)} R_{ci}(t) \geq x(t)$, for any time $t > 0$, where θ_i's are strictly positive weight functions which are to be chosen such that the power consumption of each sensor is consistent with the corresponding residual energy.

Moreover, θ_{s1} is the weight function for the sensing power of sensor 1 (which is, by assumption, assigned to sense the target).

Definition 2.1. *Given an MSN satisfying Assumptions 2.1 to 2.3:*

- *The minimization problem with the performance index J_D and no constraints will hereafter be referred to as the* unconstrained durability optimization problem *(UDOP), and the corresponding minimum cost will be denoted by J_D^*.*
- *The minimization problem with the performance index J_D and the constraints (i) and (ii) will hereafter be referred to as the* constrained durability optimization problem *(CDOP), and the corresponding minimum cost will be denoted by $\bar{J}_D^*$.*
- *The minimization problem with the performance index J_P and the constraints (i) and (ii) will hereafter be referred to as the* constrained power optimization problem *(CPOP), and the corresponding minimum cost will be denoted by $\bar{J}_P^*$.*

Definition 2.2. *Consider the weight functions $\theta_i(t)$'s in (2.4), and let $m \in \mathbf{n}$ be a given integer. The* inverse-weight coefficient *is defined as:*

$$\sigma_m(t) = \sum_{i=1}^{m} \left(\frac{1}{\theta_i(t)} \right)^{\frac{1}{\lambda - 1}}, \ \forall t \geq 0$$

and will prove convenient in the development of the main results.

Remark 2.1. *Note that since θ_i is strictly positive for all $i \in \mathbf{n}$, it is straightforward to conclude that $\sigma_{j+1}(t) > \sigma_j(t)$, for all $j \in \{1, \ldots, n-1\}$.*

Before proceeding to the next section, let *Holder's inequality* [122] (which will be used later to prove some of the results) be provided here.

Holder's Inequality: Given the strictly positive values p and q with the property $\frac{1}{p} + \frac{1}{q} = 1$, then the following inequality:

$$\sum_{i=1}^{n} |x_i y_i| \leq \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}} \left(\sum_{i=1}^{n} |y_i|^q \right)^{\frac{1}{q}} \tag{2.5}$$

holds for all $(x_1, x_2, \ldots, x_n), (y_1, y_2, \ldots, y_n) \in \mathbb{C}^n$.

In the next section, effective techniques are proposed to solve the optimization problems introduced earlier at any given time t. For convenience of notation, the time argument from the variables x, R_{s1}, R_{ci}, σ_i, θ_i and the set $\mathbf{k}$ will be omitted in the development of the main results.

2.2 Main Results

Consider an MSN satisfying Assumptions 2.1-2.3. The following two lemmas will be used to solve the UDOP.

Lemma 2.1. *Let J_D be the minimum cost in the UDOP after setting the sensing radius of sensor 1 to R_{s1}. Then:*

$$J_D = \frac{(x - R_{s1})^{\lambda}}{\sigma_n^{\lambda-1}} + \alpha\theta_{s1}R_{s1}^{\gamma}$$

Proof. Choose $x_i = \theta_i^{\frac{1}{\lambda}} R_{ci}$, $y_i = \theta_i^{-\frac{1}{\lambda}}$, $p = \lambda$ and $q = \frac{\lambda}{\lambda-1}$. Then, using Holder's inequality along with Remark 2.1 (and on noting that $\exists \mathbf{k} \subset \mathbf{n}$, such that $R_{s1} + \sum_{i\in\mathbf{k}} R_{ci} \geq x$), one can write:

$$\sum_{i\in\mathbf{k}} \theta_i R_{ci}^{\lambda} \geq \frac{\left(\sum_{i\in\mathbf{k}} R_{ci}\right)^{\lambda}}{\sigma_k^{\lambda-1}} \geq \frac{\left(\sum_{i\in\mathbf{k}} R_{ci}\right)^{\lambda}}{\sigma_n^{\lambda-1}} \geq \frac{(x - R_{s1})^{\lambda}}{\sigma_n^{\lambda-1}} \tag{2.6}$$

for some $\mathbf{k} \subset \mathbf{n}$. It follows immediately from (2.6) that $J_D \geq \frac{(x-R_{s1})^{\lambda}}{\sigma_n^{\lambda-1}} + \alpha\theta_{s1}R_{s1}^{\gamma}$, and that the inequality turns to an equality by choosing $R_{ci} = \frac{(x-R_{s1})}{\theta_i^{\frac{1}{\lambda-1}}\sigma_n}$. This completes the proof. ■

Lemma 2.2. *The function*

$$f(R_{s1}) = \sigma_n^{\lambda-1}\alpha\gamma\theta_{s1}R_{s1}^{\gamma-1} - \lambda\,(x - R_{s1})^{\lambda-1} \tag{2.7}$$

has exactly one real-positive root over $[0, x]$.

Proof. Since $f(R_{s1})$ is continuous, $f(0) = -\lambda x^{\lambda-1} < 0$ and $f(x) > 0$, thus $f(R_{s1})$ has at least one real-positive root over $[0, x]$. On the other hand, since $\lambda, \gamma \geq 2$ and also $\frac{df}{dR_{s1}} > 0$ over $[0, x]$, hence $f(R_{s1})$ is strictly increasing in the above interval. Therefore, $f(R_{s1})$ has exactly one real-positive root over $[0, x]$. ■

Let the sensing radius and communication radii obtained by solving the UDOP be denoted by R^*_{s1} and R^*_{ci}, $i \in \mathbf{n}$, respectively. The following theorem characterizes the solution of the UDOP.

Theorem 2.1. *The solution of the UDOP is unique, and is characterized by:*

$$f(R^*_{s1}) = 0, \quad R^*_{ci} = \frac{(x - R^*_{s1})}{\theta_i^{\frac{1}{\lambda-1}} \sigma_n}, \quad i \in \mathbf{n}$$

Proof. The proof follows by taking derivative of J_D and using the results of Lemmas 2.1 and 2.2. ■

Remark 2.2. *It is implied from Theorem 2.1 and the proof of Lemma 2.1 that in the solution of the UDOP,* $\mathbf{k} = \mathbf{n}$*. Furthermore, the sum of the sensing radius of sensor 1 and the communication radii of all sensors at any time instant is equal to the distance between the target and destination point at that time. This means that in the optimal strategy all sensors will be functional, and will be located on distinct points on the straight line connecting the target to the destination. It is also implied that* $\theta_j R^{\lambda-1}_{cj} = \theta_i R^{\lambda-1}_{ci}, \forall i, j \in \mathbf{n}$*. In other words, the solution of UDOP has the following property:*

$$\frac{\theta_j}{\theta_i} = \frac{R^{\lambda-1}_{ci}}{R^{\lambda-1}_{cj}}, \quad \forall i, j \in \mathbf{n} \tag{2.8}$$

Remark 2.3. *It is implied from Theorem 2.1 and Lemma 2.2 that:*

$$\sigma_n^{\lambda-1} \alpha\gamma\theta_{s1} \left(R^*_{s1}\right)^{\gamma-1} = \lambda \left(x - R^*_{s1}\right)^{\lambda-1} \tag{2.9}$$

$$\left(\sigma_{n-1}\right)^{\lambda-1} \alpha\gamma\theta_{s1} \left(R^*_{s1}\right)^{\gamma-1} = \lambda \left(x - R^*_{cn} - R^*_{s1}\right)^{\lambda-1} \tag{2.10}$$

$$\frac{(x - R^*_{s1})}{\theta_i^{\frac{1}{\lambda-1}}\sigma_n} = \frac{(x - R^*_{cn} - R^*_{s1})}{\theta_i^{\frac{1}{\lambda-1}}\sigma_{n-1}}, \quad \forall i \in \{1, \ldots, n-1\} \tag{2.11}$$

Denote by $\bar{R}^*_{s1}$ and $\bar{R}^*_{ci}$, $i \in \mathbf{n}$, respectively, the sensing radius and communication radii obtained by solving the CDOP.

Lemma 2.3. *The optimal parameters associated with the CDOP have the following properties at all times:*

i) if $\theta_i = \theta_j$, then $\bar{R}^*_{ci} = \bar{R}^*_{cj}$.

ii) if $\theta_i < \theta_j$, then $\bar{R}^*_{ci} \geq \bar{R}^*_{cj}$.

iii) if $\theta_i < \theta_j$, then $\theta_j \bar{R}^{*\lambda-1}_{cj} \geq \theta_i \bar{R}^{*\lambda-1}_{ci}$.

iv) if $\theta_i < \theta_j$ and $\theta_j \bar{R}^{*\lambda-1}_{cj} > \theta_i \bar{R}^{*\lambda-1}_{ci}$, then $\bar{R}^*_{ci} = R_{c,\max}$.

Proof.

i) Assume $\theta_i = \theta_j$, but $\bar{R}^*_{ci} < \bar{R}^*_{cj}$. Define $J(\beta) := \bar{R}^{*\lambda}_{ci} + \bar{R}^{*\lambda}_{cj} - \left(\bar{R}^*_{ci} + \beta\right)^\lambda - \left(\bar{R}^*_{cj} - \beta\right)^\lambda$ for any $\beta \in \left(0, \frac{1}{2}\left(\bar{R}^*_{cj} - \bar{R}^*_{ci}\right)\right)$. Since $J(0) = 0$ and $\frac{dJ}{d\beta}$ is positive for any β in $\left(0, \frac{1}{2}\left(\bar{R}^*_{cj} - \bar{R}^*_{ci}\right)\right)$, there exists β in this interval for which $J(\beta) > 0$. This contradicts the initial assumption of the lemma about the optimality; hence, $\bar{R}^*_{ci} = \bar{R}^*_{cj}$.

ii) Assume that $\theta_i < \theta_j$, but $\bar{R}^*_{ci} < \bar{R}^*_{cj}$. Since $(\theta_j - \theta_i)\left(\bar{R}^{*\lambda}_{cj} - \bar{R}^{*\lambda}_{ci}\right) > 0$, thus $\theta_i \bar{R}^{*\lambda}_{ci} + \theta_j \bar{R}^{*\lambda}_{cj} > \theta_i \bar{R}^{*\lambda}_{cj} + \theta_j \bar{R}^{*\lambda}_{ci}$. This contradicts the initial assumption of the lemma about the optimality; hence, $\bar{R}^*_{ci} \geq \bar{R}^*_{cj}$.

iii) Assume $\theta_i < \theta_j$, but $\theta_j \bar{R}^{*\lambda-1}_{cj} < \theta_i \bar{R}^{*\lambda-1}_{ci}$. This implies that $\bar{R}^*_{ci} > \bar{R}^*_{cj}$. On the other hand, since $\theta_i < \theta_j$, there exist $y \in \left(\theta_j \bar{R}^{*\lambda-1}_{cj}, \theta_i \bar{R}^{*\lambda-1}_{ci}\right)$ and $r_1, r_2 > 0$ such that $\theta_i \left(\bar{R}^*_{ci} - r_1\right)^{\lambda-1} = \theta_j \left(\bar{R}^*_{cj} + r_2\right)^{\lambda-1} = y$. Define $T := \frac{1}{2}\min\{r_1, r_2, \bar{R}^*_{ci}, \bar{R}_{c,\max} - \bar{R}^*_{cj}\}$ (note that T is positive). Define also $\Delta J(\beta) = \theta_i \bar{R}^{*\lambda}_{ci} + \theta_j \bar{R}^{*\lambda}_{cj} - \theta_i \left(\bar{R}^*_{ci} - \beta\right)^\lambda - \theta_j \left(\bar{R}^*_{cj} + \beta\right)^\lambda$. Since $\frac{d\Delta J}{d\beta} > 0$, $\forall \beta \in (0, T]$, and also $\Delta J(0) = 0$, one can conclude that $\Delta J(\beta) > 0$, $\forall \beta \in (0, T]$. This contradicts the initial assumption of the lemma about the optimality; hence, $\theta_j \bar{R}^{*\lambda-1}_{cj} \geq \theta_i \bar{R}^{*\lambda-1}_{ci}$.

iv) Assume that $\theta_i < \theta_j$ and $\theta_j \bar{R}_{cj}^{*\lambda-1} > \theta_i \bar{R}_{ci}^{*\lambda-1}$, but $\bar{R}_{ci}^* < R_{c,\max}$. The first two inequalities imply that there exist $y \in \left(\theta_i \bar{R}_{ci}^{*\lambda-1}, \theta_j \bar{R}_{cj}^{*\lambda-1}\right)$ and $r_1, r_2 > 0$ such that $\theta_i \left(\bar{R}_{ci}^* + r_1\right)^{\lambda-1} = \theta_j \left(\bar{R}_{cj}^* - r_2\right)^{\lambda-1} = y$. Define $T := \frac{1}{2}\min\left\{r_1, r_2, \bar{R}_{cj}^*, R_{c,\max} - \bar{R}_{ci}^*\right\}$ (note that T is positive). Define also $\Delta J(\beta) = \theta_i \bar{R}_{ci}^{*\lambda} + \theta_j \bar{R}_{cj}^{*\lambda} - \theta_i \left(\bar{R}_{ci}^* + \beta\right)^{\lambda} - \theta_j \left(\bar{R}_{cj}^* - \beta\right)^{\lambda}$. Since $\frac{d\Delta J}{d\beta} > 0$, $\forall \beta \in (0, T]$, and also $\Delta J(0) = 0$, one can deduce that $\Delta J(\beta) > 0$, $\forall \beta \in (0, T]$. This contradicts the initial assumption of the lemma about the optimality; thus, $\bar{R}_{ci}^* = R_{c,\max}$, and this completes the proof. ■

Definition 2.3. *Define $\acute{R}_{ci}^*$, $i \in \mathbf{n}$, as the new communication radii obtained by solving the UDOP after setting the sensing radius of sensor 1 to $\bar{R}_{s1}^*$ (which corresponds to the solution of the CDOP). Let q and l be the indices of the smallest and largest θ_i, $i \in \mathbf{n}$, i.e. $\theta_q = \min_{i\in\mathbf{n}} \theta_i$ and $\theta_l = \max_{i\in\mathbf{n}} \theta_i$. This notation will be used in the sequel.*

Remark 2.4. *It can be concluded from Lemma 2.3 that $\bar{R}_{cq}^* \geq \bar{R}_{ci}^*$ for all $i \in \mathbf{n}$. Also, from Remark 2.2, $\acute{R}_{cq}^* \geq \acute{R}_{ci}^*$ for all $i \in \mathbf{n}$.*

Lemma 2.4. *Consider the UDOP and the parameters introduced in Definition 2.3. If $\acute{R}_{cq}^* > R_{c,\max}$, then $\bar{R}_{cq}^* = R_{c,\max}$.*

Proof. It is known that $\sum_{i=1}^n \acute{R}_{ci}^* = \sum_{i=1}^n \bar{R}_{ci}^* = x - \bar{R}_{s1}^*$, and that $\bar{R}_{cq}^* \leq R_{c,\max} < \acute{R}_{cq}^*$. This implies that $\exists z \in \mathbf{n}$ such that $\bar{R}_{cz}^* > \acute{R}_{cz}^*$. Now, it follows from Remark 2.2 that $\theta_q \bar{R}_{cq}^{*\lambda-1} < \theta_q \acute{R}_{cq}^{*\lambda-1} = \theta_z \acute{R}_{cz}^{*\lambda-1} < \theta_z \bar{R}_{cz}^{*\lambda-1}$, and hence part (iv) of Lemma 2.3 yields $\bar{R}_{cq}^* = R_{c,\max}$. ■

Remark 2.5. *It is straightforward to show that if $\acute{R}_{cq}^* \leq R_{c,\max}$, then $\acute{R}_{ci}^* = \bar{R}_{ci}^*$, $\forall i \in \mathbf{n}$. Therefore, for the case when $\acute{R}_{cq}^* = R_{c,\max}$, one can conclude that $\bar{R}_{cq}^* = R_{c,\max}$.*

Lemma 2.5. *Consider the UDOP and the parameters introduced in Definition 2.3. If $\bar{R}_{s1}^* > R_{s1}^*$, then $\acute{R}_{cq}^* \geq R_{c,\max}$.*

Proof. Assume that $\bar{R}^*_{s1} > R^*_{s1}$, but $\acute{R}^*_{cq} < R_{c,\max}$. It follows from Lemma 2.1 that $\acute{R}^*_{ci} < R^*_{ci}$, for all $i \in \mathbf{n}$. It follows from Remark 2.5 that $\acute{R}^*_{ci} = \bar{R}^*_{ci}$, $\forall i \in \mathbf{n}$. Now, it can be concluded from Lemma 2.1 that:

$$\bar{J}^* = \frac{\left(x - \bar{R}^*_{s1}\right)^\lambda}{\sigma_n^{\lambda-1}} + \alpha\theta_{s1}\bar{R}^{*\gamma}_{s1} \tag{2.12}$$

Define $T = \frac{1}{2}\min\left\{ \bar{R}^*_{s1} - R^*_{s1} \ , \ \theta_q^{\frac{1}{\lambda-1}}\sigma_n\left(R_{c,\max} - \bar{R}^*_{cq}\right) \right\}$ (note that T is a positive value). By replacing $\bar{R}^*_{s1}$ with $\bar{R}^*_{s1} - \beta$ for an arbitrary $\beta \in (0, T)$, one arrives at:

$$\bar{J}^*_\beta = \frac{\left(x - \bar{R}^*_{s1} + \beta\right)^\lambda}{\sigma_n^{\lambda-1}} + \alpha\theta_{s1}\left(\bar{R}^*_{s1} - \beta\right)^\gamma \tag{2.13}$$

Define now $\Delta J(\beta) = \bar{J}^* - \bar{J}^*_\beta$. By taking the derivative of $\Delta J(\beta)$ and on noting that $f(R_{s1})$ in (2.7) is a strictly increasing function of R_{s1} in the closed interval $[0, x]$, and that $\bar{R}^*_{s1} - \beta > R^*_{s1}$, $f(R^*_{s1}) = 0$, one can conclude that $\frac{d\Delta J}{d\beta} > 0$. Since $\Delta J(0) = 0$ and the above derivative is strictly positive for all $\beta \in (0, T)$, it results that $\bar{J}^*_\beta < \bar{J}^*$ which is a contradiction. This means that $\acute{R}^*_{cq} \geq R_{c,\max}$. ■

Theorem 2.2. *Consider the UDOP and the index q in Definition 2.3. If $R^*_{cq} \geq R_{c,\max}$, then $\bar{R}^*_{cq} = R_{c,\max}$.*

Proof. If $\bar{R}^*_{s1} > R^*_{s1}$, then according to Lemma 2.5 $\acute{R}^*_{cq} \geq R_{c,\max}$. Thus, it results from Lemma 2.4 and Remark 2.5 that $\bar{R}^*_{cq} = R_{c,\max}$. If, on the other hand, $\bar{R}^*_{s1} \leq R^*_{s1}$, then it follows from the proof of Lemma 2.1 that $\acute{R}^*_{cq} \geq R^*_{cq} \geq R_{c,\max}$. One can therefore conclude that $\bar{R}^*_{cq} = R_{c,\max}$. ■

Lemma 2.6. *Consider the UDOP and set the communication radii of all but one sensor, say sensor n, to R^*_{ci}, $i \in \{1, \ldots, n-1\}$. Let the communication radius of sensor n be chosen as $\hat{R}_{cn} < R^*_{cn}$, and solve the new unconstrained optimization problem for $n-1$ remaining sensors with $x_{\text{new}} = x - \hat{R}_{cn}$. Then the new optimal radii $\hat{R}^*_{s1}$ and $\hat{R}^*_{ci}$ have the following properties:*

i) $\hat{R}^*_{s1} > R^*_{s1}$

ii) $\hat{R}^*_{ci} > R^*_{ci}$, $i = 1, \ldots, n-1$

Proof.

i) From Lemma 2.2:

$$(\sigma_{n-1})^{\lambda-1}\,\alpha\gamma\theta_{s1}\hat{R}_{s1}^{*\gamma-1} = \lambda\left(x-\hat{R}_{cn}-\hat{R}_{s1}^{*}\right)^{\lambda-1} \tag{2.14}$$

Assume now that $\hat{R}_{s1}^{*} \leq R_{s1}^{*}$. Then:

$$(\sigma_{n-1})^{\lambda-1}\,\alpha\gamma\theta_{s1}\hat{R}_{s1}^{*\gamma-1} \leq (\sigma_{n-1})^{\lambda-1}\,\alpha\gamma\theta_{s1}R_{s1}^{*\gamma-1} \tag{2.15}$$

The above assumption along with the inequality $\hat{R}_{cn} < R_{cn}^{*}$ yields:

$$\lambda\,(x-R_{cn}^{*}-R_{s1}^{*})^{\lambda-1} < \lambda\left(x-\hat{R}_{cn}-\hat{R}_{s1}^{*}\right)^{\lambda-1} \tag{2.16}$$

Now, from (2.10), (2.14) and (2.16) it can be concluded that:

$$(\sigma_{n-1})^{\lambda-1}\,\alpha\gamma\theta_{s1}R_{s1}^{*\gamma-1} < (\sigma_{n-1})^{\lambda-1}\,\alpha\gamma\theta_{s1}\hat{R}_{s1}^{*\gamma-1}$$

which contradicts (2.15). This means that $\hat{R}_{s1}^{*} > R_{s1}^{*}$.

ii) From Theorem 2.1 and equation (2.9):

$$R_{ci}^{*\lambda-1} = \frac{(x-R_{s1}^{*})^{\lambda-1}}{\theta_i\sigma_n^{\lambda-1}} = \frac{\alpha\gamma\theta_{s1}}{\lambda\theta_i}R_{s1}^{*\gamma-1}, \quad i=1,\ldots,n-1 \tag{2.17}$$

Furthermore, from Theorem 2.1 and equation (2.14):

$$\hat{R}_{ci}^{*\lambda-1} = \frac{\left(x-\hat{R}_{cn}-\hat{R}_{s1}^{*}\right)^{\lambda-1}}{\theta_i\,(\sigma_{n-1})^{\lambda-1}} = \frac{\alpha\gamma\theta_{s1}}{\lambda\theta_i}\hat{R}_{s1}^{*\gamma-1}, \quad i=1,\ldots,n-1 \tag{2.18}$$

Now, from part (i) of the lemma and by using (2.17) and (2.18), one arrives at the following inequality:

$$\hat{R}_{ci}^{*} > R_{ci}^{*}$$

for all $i \in \{1,\ldots,n-1\}$, and this completes the proof. ■

The following lemma is the key to prove one of the important features of the UDOP.

Lemma 2.7. *Consider the UDOP and assume that the optimal communication radii of k sensors (say, sensors $n-k+1,\ldots,n$) are greater than or equal to $R_{c,\max}$. Set $R_{ci} = R_{c,\max}$ for all $i \in \{n-k+1,\ldots,n\}$, and solve the new unconstrained optimization problem for $n-k$ sensors and $x_{\text{new}} = x - kR_{c,\max}$. Then $\check{R}^*_{s1} \geq R^*_{s1}$, where $\check{R}^*_{s1}$ is the optimal sensing radius of sensor 1 in the new unconstrained optimization problem.*

Proof. The proof follows directly from Lemma 2.6, by using parts (i) and (ii) k times. In fact, it can be shown (by using Lemma 2.2 and Theorem 2.1) that $\check{R}^*_{s1}$ is the unique positive real root of the following equation over the interval $[0, x-kR_{c,\max}]$:

$$\check{f}(R) = (\sigma_{n-k})^{\lambda-1}\,\alpha\gamma\theta_{s1}R^{\gamma-1} - \lambda\,(x - kR_{c,\max} - R)^{\lambda-1} \tag{2.19}$$

where $\check{f}$ is the dual of the function f (introduced in Lemma 2.2) for the new unconstrained optimization problem. ■

Theorem 2.3. *Consider the UDOP and assume that $R^*_{s1} > R_{s,\max}$; then $\bar{R}^*_{s1} = R_{s,\max}$.*

Proof. Assume that $\bar{R}^*_{s1} < R_{s,\max}$. Let $\bar{R}^*_{ci} = R_{c,\max}$, $i = n-k+1,\ldots,n$ and $\bar{R}^*_{ci} < R_{c,\max}$, $i = 1,\ldots,n-k$. Let also g be the index of the smallest θ_i, $i \in \{1,\ldots,n-k\}$. It results from parts (i) and (ii) of Lemma 2.3 that $\bar{R}^*_{cg} = \max_{i\in\{1,\ldots,n-k\}}\left\{\bar{R}^*_{ci}\right\}$. Since $\bar{R}^*_{ci} < R_{c,\max}$, $\forall i \in \{1,\ldots,n-k\}$, one can conclude from Theorem 2.2 (for the new unconstrained optimization problem with $x_{\text{new}} = x - kR_{c,\max}$, where $n-k$ sensors' optimal communication radii are to be determined) that $R^*_{cg} < R_{c,\max}$. Therefore, it can be concluded from Remark 2.2 that $R^*_{ci} < R_{c,\max}$, $\forall i \in \{1,\ldots,n-k\}$. Hence, in the new problem setting the constrained and unconstrained optimizations both lead to the same result for the above-mentioned $n-k$ sensors. Thus, from Lemma 2.1:

$$\bar{J}^* = \frac{\left(x - kR_{c,\max} - \bar{R}^*_{s1}\right)^{\lambda}}{(\sigma_{n-k})^{\lambda-1}} + \alpha\theta_{s1}\bar{R}^{*\gamma}_{s1} + R^{\lambda}_{c,\max}\sum_{i=n-k+1}^{n}\theta_i \tag{2.20}$$

Define:

$$T := \frac{1}{2}\min\left\{R_{s,\max} - \bar{R}^*_{s1}, \theta_g^{\frac{1}{\lambda-1}}\sigma_{n-k}\left(R_{c,\max} - \bar{R}^*_{cg}\right)\right\}$$

(note that T is a positive value). Consider now the following values for the communication and sensing radii for an arbitrary $\beta \in (0, T)$:

$$\tilde{R}^*_{s1} = \bar{R}^*_{s1} + \beta$$

$$\tilde{R}^*_{ci} = R_{c,\max}, \quad i = n-k+1, \ldots, n$$
$$\tilde{R}^*_{ci} = \frac{\left(x - kR_{c,\max} - \bar{R}^*_{s1} - \beta\right)}{\theta_i^{\frac{1}{\lambda-1}}\sigma_{n-k}}, \quad i = 1, \ldots, n-k$$

As a result:

$$\tilde{J}^* = \frac{\left(x - kR_{c,\max} - \bar{R}^*_{s1} - \beta\right)^\lambda}{\left(\sigma_{n-k}\right)^{\lambda-1}} + \alpha\theta_{s1}\left(\bar{R}^*_{s1} + \beta\right)^\gamma + R^\lambda_{c,\max}\sum_{i=n-k+1}^{n}\theta_i \qquad (2.21)$$

Define $\Delta J(\beta) = \bar{J}^* - \tilde{J}^*$; since $\beta \in (0, T)$, thus $\bar{R}^*_{s1} + \beta < R_{s,\max} \leq R^*_{s1}$. Furthermore, according to Lemma 2.7, $R^*_{s1} \leq \check{R}^*_{s1}$. As a result, $\bar{R}^*_{s1} + \beta < \check{R}^*_{s1}$. On the other hand, it is known from the proof of Lemma 2.2 that $\check{f}(R)$ in (2.19) is strictly increasing with respect to R over $[0, x_{\text{new}}]$, which implies that $\check{f}(\bar{R}^*_{s1} + \beta) < \check{f}(\check{R}^*_{s1}) = 0$. By taking the derivative of $\Delta J(\beta)$ with respect to β and using the inequality $\check{f}(\bar{R}^*_{s1} + \beta) < 0$, one obtains $\frac{d\Delta J}{d\beta} > 0$. Hence, it can be deduced from $\Delta J(0) = 0$ that $\bar{J}^* > \tilde{J}^*$, which contradicts the minimality of $\bar{J}^*$ in (2.20). This means that $\bar{R}^*_{s1} = R_{s,\max}$. ■

Consider now the problem of minimizing the sum of the power consumed by all sensors, which is a special case of the underlying optimization problem. In this case, all θ_i's ($i \in \mathbf{n}$) are equal to 1. Note that the smaller the total consumed power at every instant of a given interval is, the smaller the total consumed energy in that interval is.

Theorem 2.4. *Consider the CPOP and denote the corresponding minimum cost by* $\bar{J}^*_P$. *Let* R_p *be the real positive root of* $f(R_{s1}) = n^{\lambda-1}\alpha\gamma R^{\gamma-1}_{s1} - \lambda\left(x - R_{s1}\right)^{\lambda-1}$ *over* $[0, x]$ *(see Lemma 2.2). Then* $\bar{J}^*_P$ *is equal to:*

i) $\bar{J}_P^* = \frac{(x-R_p)^\lambda}{n^{\lambda-1}} + \alpha R_p^\gamma$

if $R_p \leq R_{s,\max}$ *and* $\frac{1}{n}(x - R_p) \leq R_{c,\max}$

ii) $\bar{J}_P^* = \frac{(x-R_{s,\max})^\lambda}{n^{\lambda-1}} + \alpha R_{s,\max}^\gamma$

if $R_p > R_{s,\max}$ *and* $\frac{1}{n}(x - R_p) \leq R_{c,\max}$

iii) $\bar{J}_P^* = nR_{c,\max}^\lambda + \alpha\,(x - nR_{c,\max})^\gamma$

if $R_p \leq R_{s,\max}$ *and* $\frac{1}{n}(x - R_p) > R_{c,\max}$

Proof.

i) The proof of this part follows immediately from Theorem 2.1, on noting that θ_i's are all equal to 1, and that the solutions of the constrained and unconstrained optimization problems are equal.

ii) According to Theorem 2.3, $\bar{R}_{s1}^* = R_{s,\max}$. On the other hand, since all θ_i's are equal, one can conclude from part (i) of Lemma 2.3 that all communication radii are equal. This completes the proof of this part.

iii) Again, since all θ_i's are equal, one can conclude from part (i) of Lemma 2.3 that all communication radii are also equal. Furthermore, it can be concluded from Theorem 2.2 that this value is equal to $R_{c,\max}$, and this completes the proof. ■

Remark 2.6. *Note in Theorem 2.4 that the inequalities* $R_p > R_{s,\max}$ *and* $\frac{1}{n}(x - R_p) > R_{c,\max}$ *cannot both be satisfied at the same time, according to Assumption 2.3.*

2.2.1 A procedure to solve the CDOP

The following algorithm can be used to solve the CDOP systematically, in order to find the optimal communication and sensing radii.

Algorithm 1.

1. Choose $\zeta = n$

2. Sort θ_i $(i = 1, ..., n)$ in descending order, and let them be represented as $\theta_{i_1} \geq \theta_{i_2} \geq \ldots \geq \theta_{i_n}$

3. Find the real positive root of the following equation (with respect to R) over $[0, x]$, and denote it by R_p:

$$\left[\sum_{l=1}^{\zeta} \left(\frac{1}{\theta_{i_l}}\right)^{\frac{1}{\lambda-1}}\right]^{\lambda-1} \alpha\gamma\theta_{s1}R^{\gamma-1} - \lambda\left(x - R\right)^{\lambda-1} = 0$$

4. Set $\omega = 0$

5. Set $\bar{R}^*_{s1}$ =min$\{R_p, R_{s,\max}\}$

6. Let $\acute{R}^*_{ci_j} = \frac{x - \bar{R}^*_{s1}}{\theta_{i_j}^{\frac{1}{\lambda-1}} \left[\sum_{l=1}^{\zeta} \left(\frac{1}{\theta_{i_l}}\right)^{\frac{1}{\lambda-1}}\right]}$, $j = 1, \ldots, \zeta$

   if $\acute{R}^*_{ci_j} > R_{c,\max}$, then $\bar{R}^*_{ci_j} = R_{c,\max}$ and $\omega = \omega + 1$

   else, $\bar{R}^*_{ci_j} = \acute{R}^*_{ci_j}$

7. Set $x = x - \omega R_{c,\max}$ and $\zeta = \zeta - \omega$

8. If $\zeta \neq 0$ and $\omega \neq 0$, then go to step 3

9. If $\zeta = 0$, then $\bar{R}^*_{s1} = x - nR_{c,\max}$

10. End

Remark 2.7. *To run Algorithm 1 in a given time interval, it is required first to specify the monitoring sensor. The selection criteria can include, for example, the distance between the sensors and the target, and the residual energy of the sensors. Note that a sensor that is very close to the target and has a high level of residual energy would be more desirable, to increase the reliability and durability of target monitoring.*

In the sequel, the real-time implementation of Algorithm 1 is investigated thoroughly, and some important practical issues are addressed. Let the algorithm be executed at the time instants t_0, $t_1 := t_0 + \Delta T$, $t_2 := t_0 + 2\Delta T$, ..., where ΔT is the time interval within which the corresponding computations have to be completed and the sensors should be relocated accordingly.

For real-time implementation of Algorithm 1, the cooperating sensors need to share certain information. All sensors will need to know the position of the target as well as the weight functions $\theta_i(t)$ at $t = t_0, t_1, t_2, \ldots$ in order to minimize the cost function (2.4) in the CDOP. Three execution cycles are considered in $[t_j, t_{j+1}]$ ($j = 0, 1, 2, \ldots$) as discussed below.

i) $[t_j, t_j + \delta t_1]$: In this time interval, due to the connectivity preserving property of Algorithm 1 (from sensor 1 to the destination point), a unidirectional multi-hop communication link is always available in the network. In addition, it is assumed that the destination point is equipped with a transmitter capable of sharing the received data with other sensors (this is a realistic assumption in most sensor network applications). Then, all the required information at $t = t_j$ is shared between all sensors in this cycle, and the positions of the sensors along with their sensing and communication radii are computed.

ii) $[t_j + \delta t_1, t_j + \delta t_2]$: In this cycle, the sensors are placed in the field according to the values obtained in the first cycle (note that the sensors move in this cycle only).

iii) $[t_j + \delta t_2, t_{j+1}]$: In this cycle, it is desired to maintain connectivity and transmit the information from the target to the destination point. To this end, the target must be in the sensing range of the monitoring sensor.

In the sequel, a sufficient condition is provided which ensures the connectivity requirement of the last execution cycle given above. Suppose the monitoring sensor

detects the target at time instant t_j; let the position of the target at the time instants t_j and t_{j+1} be represented by $x_0(t_j)$ and $x_0(t_{j+1})$, respectively. Furthermore, denote the position of the monitoring sensor in the first and third execution cycles by $x_1(t_j)$ and $x_1(t_{j+1})$, respectively. Let also $D_m(t_j) = \max_{t_j \leq t \leq t_{j+1}} \|x_0(t) - x_0(t_j)\|$. Now, to ensure the target is within the sensing range in the last cycle, the maximum sensing radius $R_{s,\max}$ in Algorithm 1 should be chosen smaller than the actual maximum sensing radius $\bar{R}_{s,\max}$. Moreover, the following inequality must be satisfied:

$$\|x_0(\acute{t}) - x_1(t_{j+1})\| \leq \bar{R}_{s,\max} \tag{2.22}$$

for all $\acute{t} \in [t_j + \delta t_2, t_{j+1}]$. It can be observed from Figure 2.1 that:

$$\|x_0(\acute{t}) - x_1(t_{j+1}) \leq \|x_0(t_j) - x_1(t_{j+1})\| + \|x_0(\acute{t}) - x_0(t_j)\| \tag{2.23}$$

On the other hand:

$$\|x_0(t_j) - x_1(t_{j+1})\| \leq R_{s,\max} \tag{2.24}$$

$$\|x_0(\acute{t}) - x_0(t_j)\| \leq D_m(t_j) \tag{2.25}$$

Thus, (2.22) holds if:

$$D_m(t_j) \leq \bar{R}_{s,\max} - R_{s,\max} \tag{2.26}$$

(note that $R_{s,\max}$ is a design parameter). Denote the average speed of the target in the time interval $[t_j, t_{j+1}]$ by $\bar{v}(t_j)$. It is straightforward to show that (2.26) holds if:

$$\Delta T \bar{v}(t_j) \leq \bar{R}_{s,\max} - R_{s,\max} \tag{2.27}$$

This implies that in order to preserve connectivity, the target should not move too fast. It is worth mentioning that by choosing a small $R_{s,\max}$, the condition given in (2.27) will be satisfied for faster target. However, this would be achieved at the expense of inefficient use of the sensing capabilities of the monitoring sensor.

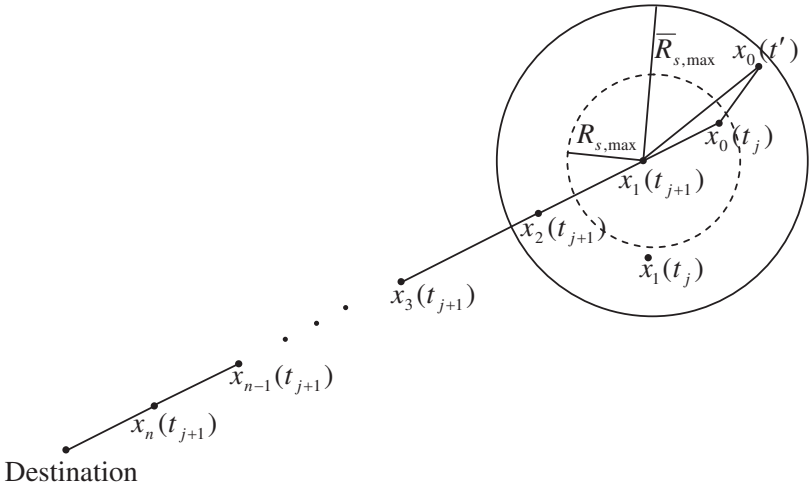


Figure 2.1: An illustrative figure for the real-time implementation of Algorithm 1.

It is also to be noted that if the target is fixed in the interval $[t_j, t_{j+1}]$, the sensor locations in the third execution cycle are the optimal sensor positions for transferring information from the target to the destination point. Otherwise, if the target moves, since the sensor positions are obtained from Algorithm 1 using $x_0(t_j)$, the sensor locations are near-optimal.

Remark 2.8. *For solving CPOP, one can set θ_{s1} and all θ_i's ($i \in \mathbf{n}$) to 1 and use Algorithm 1.*

2.3 Simulation Results

Example 1. Consider an MSN consisting of 6 sensors, and let the corresponding parameters be given by $R_{c,\max} = 20$m, $\bar{R}_{s,\max} = 7$m, $\lambda = 3.2$, $\gamma = 5$, and $\alpha = 3$. Let $\Delta T = 2$sec, and assume that the average speed of the target in any execution time interval is less than 0.5m/sec. One can verify that (2.27) is satisfied by choosing $R_{s,\max} = 6$m. Let the initial residual energy of each sensor be a uniformly distributed random number between 480J and 2400J. It is assumed that the sensor with the highest initial energy is selected as the monitoring sensor, and that only this sensor will be monitoring the target throughout the mission (the latter assumption is mainly

for simplicity of analysis, as stated earlier). The simulation results presented here are obtained by MATLAB for two different scenarios.

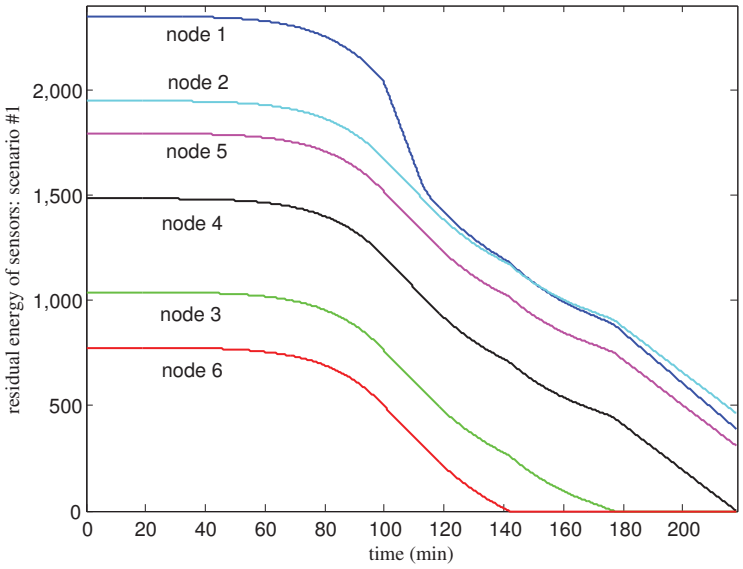


Figure 2.2: Residual energy of the sensors in Example 1, under the total power consumption minimization strategy (first scenario).

Scenario 1: In the first scenario, it is desired to minimize the sum of the power consumption of all sensors. In this case, one can use Algorithm 1 and Theorem 2.4 to find the optimal solution for the CPOP. Figure 2.2 depicts the residual energy of the sensors versus time, using the optimal strategy in this case. At $t = 142$min, sensor 6 (which has the smallest residual energy) runs out of energy and network has to operate using the five remaining sensors. Furthermore, at $t = 177$min sensor 3 also runs out of energy, and the network has to continue its operation by only four sensors. As can be perceived from Figure 2.2, the rate of energy consumption increases in the network after losing each one of these two sensors. In addition, the energy of sensor 4 is depleted at $t = 218$min, and since the distance between the target and destination point is 76m (which is more than $3R_{c,\max} + R_{s,\max} = 66$m), the network cannot operate with the remaining three sensors.

Remark 2.9. *To apply Algorithm 1 in this scenario, only the value of x needs to*

be transmitted through the unidirectional link from sensor 1 to the sensor with the smallest distance to the destination point. The optimization procedure will then be performed by each sensor separately.

Scenario 2: The objective here is that all sensors work cooperatively for a long period of time. For this purpose, the residual energy of every sensor needs to be monitored at all times, and the power consumption of each sensor is to be adjusted accordingly, so that the life-span of every sensor in the network becomes more or less the same as shown in Figure 2.3. To this end, the sensing and communication radii are chosen in such a way that if the residual energy of one sensor, say sensor i, at one instant is k times greater than that of another sensor, say sensor j, then the rate of energy consumption of sensor i must be k times greater than that of sensor j. It is desired now to choose the values of θ_i's at every time instant such that the above objective is achieved by minimizing $\bar{J}_D^*$. As noted from (2.2) and (2.8), if at time t the residual energy of sensor i is k times greater than that of sensor j, then θ_j should be set $k^{\frac{\lambda-1}{\lambda}}$ times greater than θ_i at that time instant, for all $i, j \in \mathbf{n}$. This condition is satisfied by choosing $\theta_i = (\text{residual energy of sensor } i)^{\frac{1-\lambda}{\lambda}}$, for any $i \in \mathbf{n}$. Algorithm 1 can now be used to solve CDOP.

Remark 2.10. *It is to be noted that the power consumption of sensor 1 is not due solely to communication, and part of it is due to sensing. Furthermore, the relation (2.8) is not necessarily valid in constrained optimization. However, since the coefficients $\theta_1, \ldots, \theta_n$ are tuned online, the strategy described above is still effective in increasing the life-span of the network in both constrained and unconstrained optimization problems.*

In Figure 2.3, the residual energy of sensors is plotted versus time. This figure shows that all sensors run out of energy simultaneously at $t = 336$min. Furthermore, one can observe that there is a significant drop in the residual energy of sensor 1 from

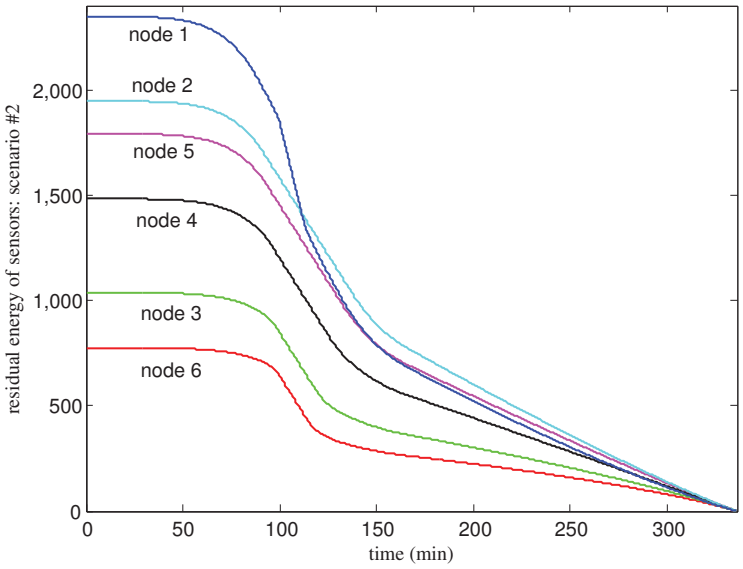
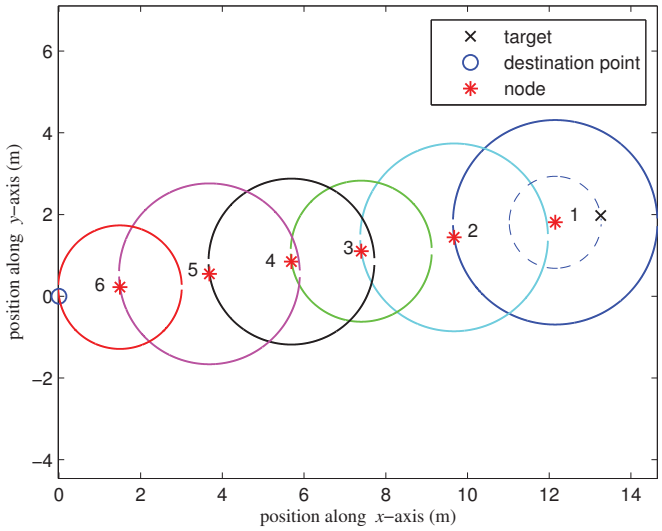


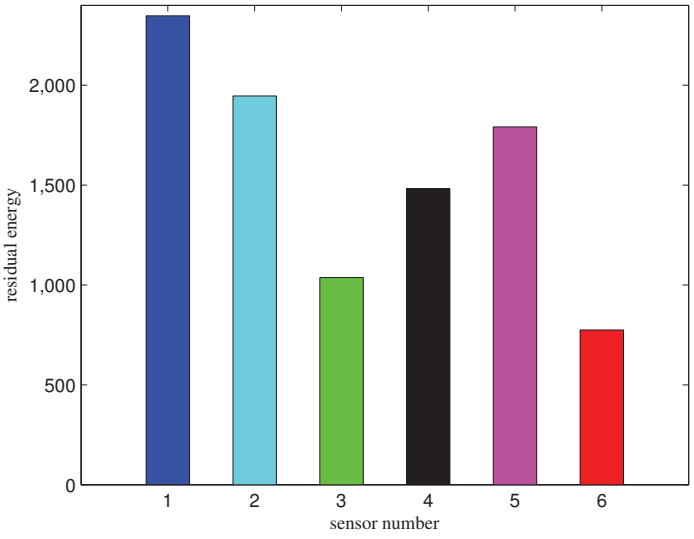
Figure 2.3: Residual energy of sensors in Example 1, under the life-span maximization strategy (second scenario).

$t = 100$min to $t = 112$min in the figure. This is due to the fact that the distance between the target and destination point becomes close to $nR_{c,\max} + R_{s,\max} = 126$m in this time interval (see also Figure 2.5(a)). This in turn pushes the communication radius of each sensor closer to its maximum allowable value, and more importantly, the sensing radius of sensor 1 closer to its maximum allowable value.

Figures 2.4(a), 2.5(a) and 2.6(a) depict the location of target and sensors under the second scenario in the $x-y$ plane in three different time instants $t = 15$, 105 and 200min, respectively. As it can be observed from Figures 2.4(a), 2.5(a) and 2.6(a), the proposed strategy aligns all sensors on a straight line. Their exact location on the line as well as their communication and sensing radii are computed online, based on the residual energy of every sensor in the network. The boundary of the region where the signal transmitted by each sensor can be received is marked by a colored solid circle in these figures. Similarly, the boundary of the sensing region corresponding to sensor 1 is marked by a blue dashed circle. The residual energy of each sensor in the above time instants is plotted in Figures 2.4(b), 2.5(b) and 2.6(b),

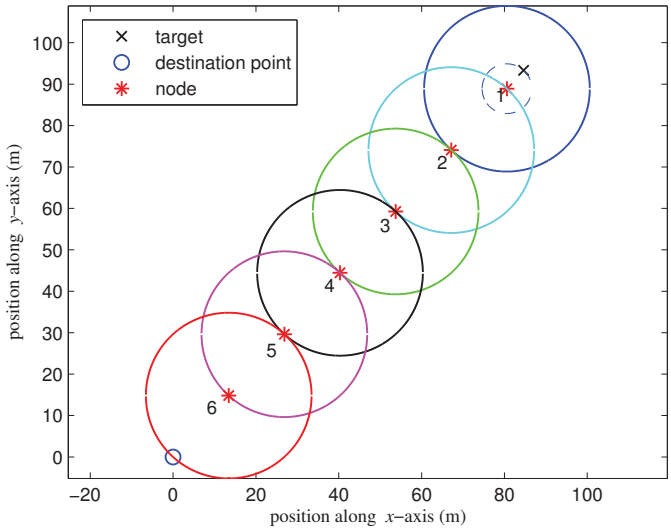


(a)

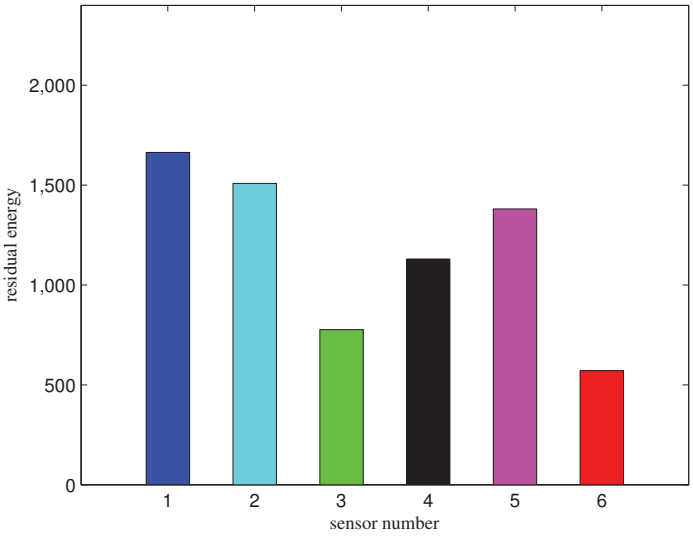


(b)

Figure 2.4: (a) The location of the target, destination point, and sensors at $t =$ 15min. (b) The residual energy of each sensor at $t = 15$min.



(a)



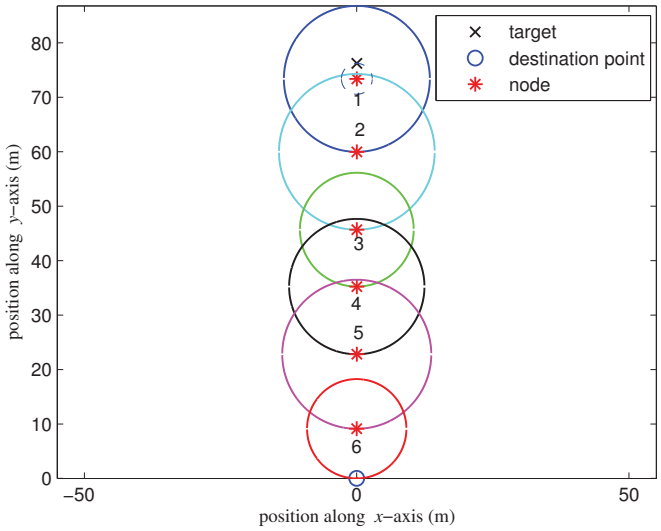
(b)

Figure 2.5: (a) The location of the target, destination point, and sensors at $t =$ 105min. (b) The residual energy of each sensor at $t = 105$min.

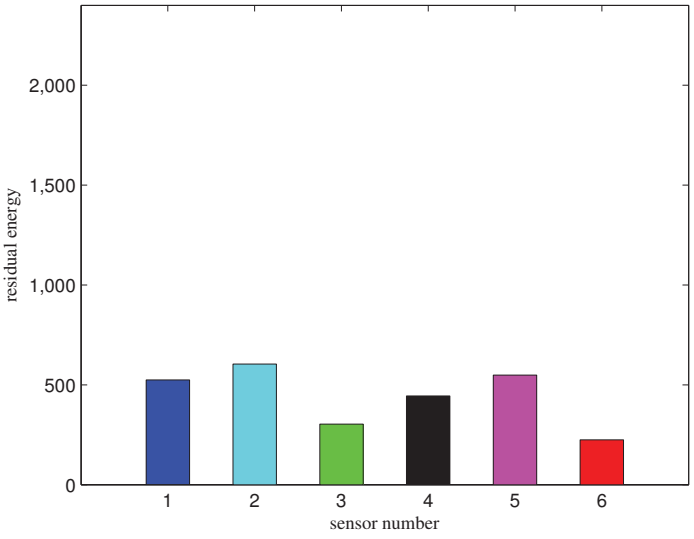
accordingly. Note that the communication radii of the sensors are proportional to their residual energy at $t = 15$min, but they are all equal at $t = 105$min (although different sensors have different residual energies at this time instant). This is due to the fact that the distance between the target and destination point is nearly 126m at this instant, and this is the farthest distance the sensors can cover. Hence, the maximum radii for communication and sensing need to be adopted by the sensors, regardless of their residual energies. Moreover, since the distance between the target and destination point is about 76m at $t = 200$min, sensors do not need to use their maximum communication and sensing radii. Hence, the communication radius of each sensor is again proportional to its residual energy.

Remark 2.11. *To implement scenario 2, two computational schemes can be considered here: centralized and distributed. In a centralized scheme, the information about θ_{s1}, x and θ_i's, $\forall i \in \mathbf{n}$, is transmitted to the destination point, and the optimal values R_{s1} and R_{ci}'s are computed accordingly, along with the location of each sensor. The optimal parameters are then transmitted back to all sensors in the MSN. In a distributed scheme, on the other hand, any required information is shared between all sensors through the destination point, and each sensor in the MSN uses Algorithm 1 separately to find the optimal parameters.*

By comparing Figures 2.2 and 2.3, it can be observed that the cooperation of the sensors in the second scenario lasts 54% longer than that of the same sensors in the first scenario. Another comparison of the two strategies is provided in Figure 2.7. This figure shows the sum of residual energies of all sensors versus time in the first scenario (dotted curves) and the second scenario (solid curves). As long as all 6 sensors are operating in the network (before $t = 142$min), the total residual energy of the sensors in the first scenario is more than that in the second scenario as expected. Once a sensor runs out of energy, the rate of total residual energy consumption in the first scenario becomes greater than that in the second scenario, such that after



(a)



(b)

Figure 2.6: (a) The location of the target, destination point, and sensors at $t =$ 200min. (b) The residual energy of each sensor at $t = 200$min.

$t = 161$min the sum of residual energies in scenario 1 becomes less than that in the second scenario. This is due to the fact that once a sensor stops operating, the remaining sensors are forced to adopt larger communication and sensing radii to preserve network connectivity, which leads to an increase in the rate of total energy consumption in scenario 1 (this would be more significant for larger values of λ and γ). It can be observed from Figure 6 that right before the first sensor runs out of energy in scenario 1, the sum of residual energies of the sensors in the first scenario is at most 3% larger than that in the second scenario. Thus, the relocation strategy proposed in scenario 2 which increases the durability of the mobile sensor network will, at the same time, reduce the total energy consumption significantly.

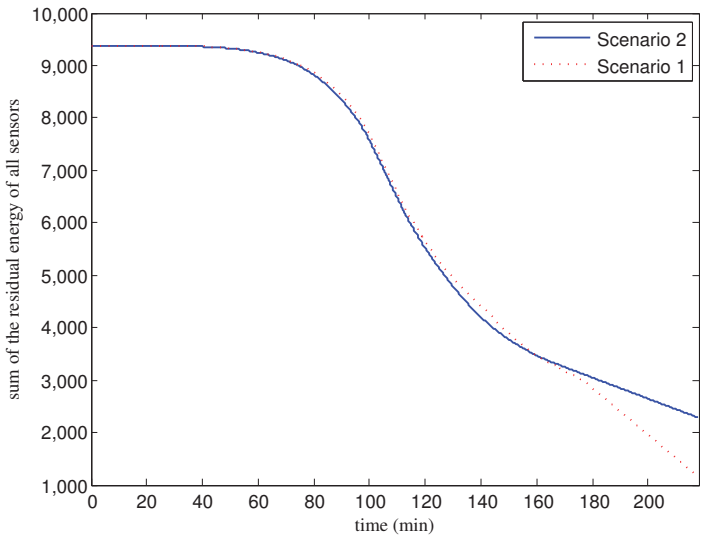


Figure 2.7: The total residual energy of all sensors in both scenarios.

Remark 2.12. *One can use a combination of the two strategies proposed in this chapter, to address the trade off between the total power consumption and the durability of the network. This is carried out in two phases. First, the minimum sum of power consumption strategy is adopted until the total energy of the network reaches a predefined level, at which time the system switches to the maximum durability strategy. Depending on the network parameters, this new strategy may provide a good compromise between the energy consumption and life-span of the network.*

Chapter 3

Cost-Efficient Routing with Controlled Node Mobility in Sensor Networks

In this chapter, an energy-efficient strategy is proposed for tracking a moving target in a mobile sensor network. The energy expenditure of the sensors in the network is assumed to be due to communication, sensing and movement. First, the target area is divided into a grid of sufficiently small rectangular cells in order to search for near optimal locations for the sensors in different time instants. The grid is then converted to a graph with properly weighted edges. A shortest-path algorithm is subsequently used to route information from target to destination by a subset of sensors.

The organization of this chapter is as follows. In Section 3.1 the problem is introduced and important assumptions and definitions are provided. Section 3.2 presents the proposed routing technique as the main contribution of this chapter. In Section 3.3, the complexity and performance of the algorithm are discussed. Simulations are presented in Section 3.4 to support the theoretical findings.

3.1 Problem Statement

Consider a group of n mobile sensors $S_1, \ldots, S_n$ aimed to track a moving target. The sensors are distributed in a field where the target moves, and their mission is to preserve connectivity between target and destination (a fixed location where the network information is collected). Furthermore, in selecting those sensors which create a route from the target to destination, cost-effectiveness must be taken into account. This cost is concerned with the energy consumed to establish connectivity, and depends mainly on movement, sensing and communication.

Assumption 3.1. *It is assumed that one sensor is properly selected to sense the target, and all other sensors can potentially be used to create an information route to the destination at any time instant. This sensor is referred to as the tracking sensor, and is not necessarily fixed. The tracking sensor at any point in time is selected based on the target position and the energy-efficient deployment strategy discussed later.*

Let the tracking sensor be denoted by S_T, with the maximum sensing radius R_S (note that $S_T \in \{S_1, S_2, ..., S_n\}$ at any time instant). This means that if the target is within a circle of radius R_S centered at S_T, then it can be detected by this sensor.

Assumption 3.2. *The target is assumed to be within a reachable distance from the destination at all times, i.e. $x(t) \leq nR_C + R_S, \forall t$, where x is the distance between the target and destination, R_C is the maximum communication radius of each sensor, and n is the number of sensors.*

To minimize the energy consumption, the sensors must operate in a collaborative fashion in order to determine the best locations for sensors, and the best routing path to communicate the information. The energy loss due to movement is assumed to be proportional to the distance. The energy loss due to communication and sensing between two nodes P and Q, on the other hand, is proportional to

$d(P,Q)^{\lambda}$ and $d(P,Q)^{\gamma}$, respectively, where $d(P,Q)$ is the distance between P and Q. Moreover, λ and γ are positive real values which depend on the characteristics of the environment (typically $\gamma > \lambda$).

To develop the energy-efficient routing technique, the field is first divided into a grid, and it is assumed that the sensors are located on the nodes of the grid at any time instant. Three different graphs are then constructed, whose vertices are the grid nodes, and whose edges are weighted properly, in accordance with the available models for the three sources of energy consumption.

The three weighted graphs (which are, in fact, directed) are subsequently combined to obtain the overall energy consumption graph, which will be referred to as the *combined energy digraph*. The following notation and definitions will prove convenient in the development of the main results.

Notation 3.1. *Throughout this chapter, the j-th nearest sensor to node P will be denoted by S_P^j, for any $j \in \mathbf{n} := \{1, 2, ..., n\}$. For example, S_P^1 represents the nearest sensor to node P. Furthermore, d_P^j denotes the distance between S_P^j and P.*

Definition 3.1. *In this chapter, the term path nodes refers to all the nodes on a given path connecting the target to destination, excluding the target and destination themselves.*

3.2 Main Results

In this section, a strategy is presented to properly place the sensors in the field at any time instant in such a way that the total energy consumption due to the sensing, communication and movement of the sensors is sufficiently close to its minimum value.

Consider n sensors which can move on the surface of a field. Let the field be divided into a grid of a given size. Partition also the field into a Voronoi diagram

with n regions (each region associated with one sensor). Let the j-th region of the Voronoi diagram be denoted by Λ_j, for any $j \in \mathbf{n}$. Three different graphs are constructed in the sequel.

1) Communication energy digraph: Construct a directed graph (digraph) in which the edges are properly weighted to model the communication cost between the sensors. In this digraph, there is an edge from the node P_i to P_j if P_j is in the communication range of P_i, i.e., if the distance between them is less than or equal to R_C. It is to be noted that all edges in this digraph are bidirectional.

Definition 3.2. *The region containing the target and the node on which the target resides will hereafter be denoted by Λ_T and P_T, respectively. In addition, the destination will be denoted by P_D.*

2) Sensing energy digraph: From the properties of the Voronoi diagram, it is known that the closest sensor to any point in a Voronoi region is the sensor associated with that region. The target is assumed to be tracked by the closest sensor to it. This implies that the target and the sensor which tracks it at any point in time, are in the same Voronoi region. Every node of the grid whose distance from the target is less than R_S and is in Λ_T is connected to the target by a directed edge (from the target to the node) with a weight proportional to the corresponding sensing energy. Moreover, any node on the grid which is in Λ_T and is within a distance of R_S from P_T will be referred to as a *sensing node*. Figure 3.1 shows a sample position of the target and the sensor energy digraph edges. R_S is assumed to be 2 in this figure, and the target is connected to any point in the grid in the region Λ_T and enclosed in the circle with radius of R_S centered at the target.

3) Movement energy digraph: Construct a directed graph in which the edges are appropriately weighted to model the energy required for the sensors to move to

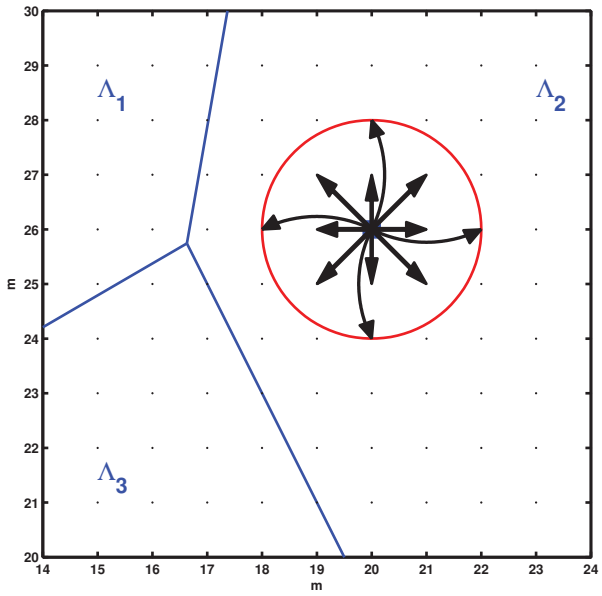


Figure 3.1: An illustrative figure of the sensing energy digraph.

a proper location in order to transmit information from the target to destination. This directed graph is called movement energy digraph. The weight of the directed edge from P_i ($P_i \neq P_T$) to P_j is denoted by $mov(i,j)$, which depends on the locations of the two nodes. The following procedure is used to find this weight.

- Consider the case where P_i and P_j are in different Voronoi regions, OR P_j is the destination node.

  i) If the target and P_i are in the same region AND P_i is not a sensing node, then:

  $$mov(i,j) = \beta.d_{P_i}^2$$

  where β is a constant coefficient.

  ii) If the target and P_i are in different regions OR P_i is a sensing node, then:

  $$mov(i,j) = \beta.d_{P_i}^1$$

- Consider now the case where P_i and P_j are in the same region, AND P_j is not the destination node.

  i) If the target and P_i are in the same region AND P_i is not a sensing node, then:

$$mov(i,j) = \beta.d^2_{P_i}$$

  ii) If the target and P_i are in different regions, then:

$$mov(i,j) = \beta.[min(d^1_{P_i} + d^2_{P_j}, d^1_{P_j} + d^2_{P_i}) - d^1_{P_j}]$$

  iii) If P_i is a sensing node, then:

$$mov(i,j) = \beta.d^1_{P_i}$$

It is important to notice that, our algorithm tries to allocate weights to the movement energy digraph such that in any arbitrary path from target to destination, the sum of the allocated weights to the path edges by the algorithm is a lower limit which is as close as possible to the minimum movement energy required for a subset of sensors to move to path nodes and start routing the information. As simulation results will show, our proposed method is successful in a high percent of cases. Figure 3.2 gives an illustration of the above cases for edges which each are assumed to belong to an arbitrary path. For example, in this figure, nodes B and C satisfy the conditions of part (i) of the second case and the edge BC is assigned a weight of $\beta.d^1_B = \beta.d(S_4, B)$ where $d(S_4, B)$ is the distance between S_4 and B . Edges AD, EF and GH satisfy part (ii) of the first case. In the edge AD, node $P_j = D$ is the destination. For EF, $P_i = E$ and the target are in different regions and in GH, $P_i = G$ is a sensing node, therefore they will be assigned the weights $\beta.d^1_A = \beta.d(S_3, A)$, $\beta.d^1_E = \beta.d(S_1, E)$ and $\beta.d^1_G = \beta.d(S_2, G)$ respectively. The edge KL is an example of part (i) of the second case in which $P_i = K$ and the target are in different regions and K is not a sensing node. The

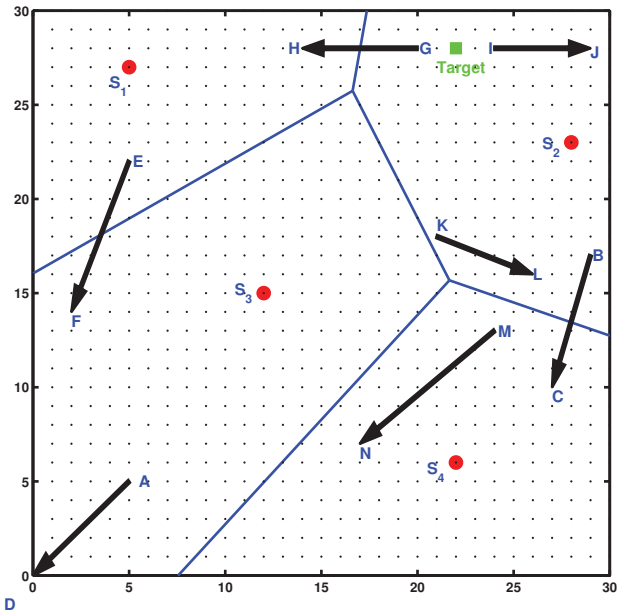


Figure 3.2: An illustrative figure about the edges of the movement energy digraph.

algorithm assigns the weight $\beta.d_K^2 = \beta.d(S_3, K)$ to it. The edge MN satisfies part (iii) of the second case. In this case, the nearest sensor to M and N is the same and the algorithm assigns the weight $\beta.[min(d_M^1 + d_N^2, d_M^2 + d_N^1) - d_N^1] = \beta.[min(d(S_4, M) + d(S_3, N), d(S_4, N) + d(S_2, M)) - d(S_4, N)]$ to this edge. Finally, the edge IJ is an example of part (iii) of the second case. In this case, since $P_i = I$ is a sensing node, the weight of this edge is $\beta.d_I^1 = \beta.d(S_2, I)$.

Once the above three digraphs are constructed, derive a new digraph called *combined energy digraph*, in which the node P_i is connected to P_j if there is a directed edge from P_i to P_j in at least one of the three digraphs. The weight assigned to this edge is the sum of the weights of the corresponding existing edges in the three digraphs. Notice that, if the distance between any two grid points P_i and P_j is greater than R_C, the corresponding edge in the communication energy digraph has been assigned a weight of infinity, therefore making the corresponding edge in the combined energy digraph of infinite weight. It is desired in this new graph to find the

shortest weighted path connecting the target to destination, subject to the constraint that the number of nodes in the path is less than or equal to the number of sensors. It will be shown that this path provides a cost-effective route, which can, under some conditions, be optimal.

Remark 3.1. *One can use a proper fast and efficient routing algorithm (such as Dijkstra) to find the shortest path. If in the end the number of nodes in the shortest path was greater than n, then one can switch to a constrained shortest path algorithm, which is normally slower than unconstrained algorithms.*

Definition 3.3. *The sum of the weights of the directed edges of a path Π is referred to as the path weight, and is denoted by $W(\Pi)$. Note that the path weight is, in fact, the sum of the weights of directed edges of the sensing energy digraph, communication energy digraph and movement energy digraph, which will hereafter be called the sensing path weight, communication path weight, and movement path weight, respectively.*

Definition 3.4. *Given a path $\Pi = (P_T, P_1, P_2, ..., P_m, P_D)$ connecting the target to destination, the minimum energy required for any group of m sensors to be located at $P_1, P_2, ..., P_m$ and transmit the information from the target to destination is called the path cost, and is denoted by $C(\Pi)$. Note that the path cost is, in fact, the sum of the minimum energy required for the selected sensors to move to their designated locations on the path, sense the target, and communicate with each other on the path, which will hereafter be referred to as the movement path cost, sensing path cost, and communication path cost, respectively. However, to find this value, it suffices to consider the movement energy only, and add it to the fixed sensing and communication energy required to establish the underlying information link. This is due to the fact that all sensors are assumed to be identical in terms of sensing and also communication capabilities.*

Definition 3.5. *The optimal path in a combined energy digraph is a path consisting of at most n nodes, such that there exist a group of sensors which the cost of moving them to these nodes and establishing an information link from the target to destination is minimum, among all possible choices of paths and sensors. This path will be denoted by* Π^*.

Theorem 3.1. *Consider a path* Π *which connects the target to destination such that:*

i) Π *has at most two nodes in each Voronoi region.*

ii) If a region Λ_k *contains exactly two nodes of the path, say* P_i *and* P_j*, then the path* Π *does not pass through any other region containing the second nearest sensor to* P_i *or* P_j.

Then, the path cost and path weight of Π *are equal.*

Proof. Since the communication and sensing path costs for any fixed path are equal to the communication and sensing path weights, respectively, it suffices to show that the movement path cost and movement path weight are equal. To this end, consider the following three cases:

Case 1: Region Λ_k *contains only one node.* To minimize the movement energy in this case, one can assign the nearest sensor of this node to it. From the weight assignment rule in the movement energy digraph, it follows that the movement path cost and movement path weight are equal.

Case 2: Region Λ_k *contains the two nodes* P_i *and* P_j*, but not the node* P_T *(target).* Similar to the previous case, it results from the weight assignment rule in the movement energy digraph that the sum of the weights of the edge from P_i to P_j and the edge coming out of P_j is given by:

$$\begin{aligned} X_k &= \beta.[min(d^1_{P_i} + d^2_{P_j}, d^1_{P_j} + d^2_{P_i}) - d^1_{P_j}] + \beta.d^1_{P_j} \\ &= \beta.[min(d^1_{P_i} + d^2_{P_j}, d^1_{P_j} + d^2_{P_i})] \end{aligned} \tag{3.1}$$

Since Π does not pass through the Voronoi regions containing the second nearest sensors to P_i and P_j, the above value is the minimum energy required to place the sensors in these two nodes.

Case 3: Region Λ_k contains the two nodes P_i and P_j, as well as the node P_T. In this case, the sum of the weights of the corresponding edges is:

$$\beta.d^1_{P_i} + \beta.d^2_{P_j} \tag{3.2}$$

Since Π does not pass through the region containing the second nearest sensor to P_j, thus (3.2) gives the minimum energy to place two sensors in P_i and P_j.

Since the discussions given above are valid for all Voronoi regions, one can conclude that the path cost and path weight of Π are equal. ■

Corollary 3.1. *Consider a path Π connecting the target to destination in a given combined energy digraph. If Π has exactly one node in any region it passes through, then the path cost and path weight of Π are equal.*

Proof. The proof follows immediately from Theorem 3.1, as a special case. ■

Theorem 3.2. *For any path Π connecting the target to destination in a combined energy digraph, the relation $W(\Pi) \leq C(\Pi)$ holds.*

Proof. Since for any fixed path the communication and sensing path costs are equal to communication and sensing path weights, respectively, it suffices to show that the movement path cost is greater than or equal to the movement path weight. To this end, assume that the path Π passes through the regions $\Lambda_1, \Lambda_2, ..., \Lambda_k$, and that the path has n_i nodes in region $\Lambda_i,\ i \in \{1, 2, ..., k\}$. Partition Π into k sub-paths as follows:

$\Pi^1 = (P_T, P^1_1, P^1_2, ..., P^1_{n_1}, P^2_1)$
$\Pi^2 = (P^2_1, P^2_2, ..., P^2_{n_2}, P^3_1)$

$\vdots$

$\Pi^k = (P_1^k, P_2^k, ..., P_{n_k}^k, P_D)$

Now, it suffices to show that the movement path weight of the sub-path Π^i is less than or equal to the corresponding movement path cost, for any $i \in \{1, 2, ..., k\}$. If Λ_i contains exactly one node, then the sub-path Π^i contains only the edge (P_1^i, P_1^{i+1}) (note that P_D is, in fact, P_1^{k+1}). The assigned weight to this edge in the movement energy digraph is $\beta.d^1_{P_1^i}$, which is the energy required to move to the node P_1^i, the nearest sensor to it. It is obvious that the minimum required energy for a sensor to move to P_1^i is equal to $\beta.d^1_{P_1^i}$ as well (note that sometimes the sensor assigned to a node is not necessarily its nearest sensor, because that may be the nearest sensor to multiple nodes in the path). Therefore, in this case, the movement path weight of sub-path Π^i is less than or equal to the movement path cost.

If Λ_i contains more than one node, there will be two possibilities as follows:

Case 1: $i \neq 1$. In this case, the weight assigned to the sub-path Π^i in the movement energy digraph is:

$$X = \beta.[\sum_{k=1}^{n_i-1} min(d^1_{P_k^i} + d^2_{P_{k+1}^i}, d^1_{P_{k+1}^i} + d^2_{P_k^i}) - d^1_{P_{k+1}^i}] + \beta.d^1_{P_{n_i}^i}$$

From the properties of the Voronoi diagram, the nearest sensor to all nodes of the sub-path Π^i is the same. However, this sensor can move to only one node; therefore, the cost of moving n_i sensors to the n_i nodes of the path which lie in the region Λ_i is greater than or equal to:

$$Y = \beta.[d^1_{P_j^i} + \sum_{k=1, k\neq j}^{n_i} d^2_{P_k^i}]$$

for any $j \in \{1, 2, ..., n_i\}$. Now, consider the following relations:

$$\begin{aligned} X_1 &= \beta.[\sum_{k=1}^{j-1}(d^1_{P^i_{k+1}} + d^2_{P^i_k}) - d^1_{P^i_{k+1}}] \\ &\geq \beta.[\sum_{k=1}^{j-1} min(d^1_{P^i_k} + d^2_{P^i_{k+1}}, d^1_{P^i_{k+1}} + d^2_{P^i_k}) - d^1_{P^i_{k+1}}] \end{aligned} \tag{3.3}$$

$$\begin{aligned} X_2 &= \beta.[\sum_{k=j}^{n_i-1}(d^1_{P^i_k} + d^2_{P^i_{k+1}}) - d^1_{P^i_{k+1}}] + \beta.d^1_{P^i_{n_i}} \\ &\geq \beta.[\sum_{k=j}^{n_i-1} min(d^1_{P^i_k} + d^2_{P^i_{k+1}}, d^1_{P^i_{k+1}} + d^2_{P^i_k}) - d^1_{P^i_{k+1}}] + \beta.d^1_{P^i_{n_i}} \end{aligned} \tag{3.4}$$

By expanding and simplifying the last two inequalities, one can conclude that:

$$Y = X_1 + X_2$$

Hence:

$$\begin{aligned} Y &= X_1 + X_2 \\ &\geq \beta.[\sum_{k=1}^{n_i-1} min(d^1_{P^i_k} + d^2_{P^i_{k+1}}, d^1_{P^i_{k+1}} + d^2_{P^i_k}) - d^1_{P^i_{k+1}}] + \beta.d^1_{P^i_{n_i}} = X \end{aligned} \tag{3.5}$$

Since Y is less than or equal to the movement path cost of the sub-path Π^i, it results from the above relation that the movement path weight of this sub-path is less than or equal to its movement path cost.

Case 2: $i = 1$ (the region contains the target). In this case, the nearest sensor to the nodes of this region is clearly assigned to detect the target, and hence cannot be assigned to another node simultaneously. As a result, the cost of moving n_1 sensors to n_1 nodes of the sub-path Π^1 is greater than or equal to:

$$Y = \beta.[d^1_{P^1_1} + \sum_{k=2}^{n_1} d^2_{P^1_k}]$$

On the other hand, the weight assigned to the sub-path Π^1 in the movement energy digraph is:

$$X = \beta.[d^1_{P^1_1} + \sum_{k=2}^{n_1} d^2_{P^1_k}]$$

This means that the movement path weight of Π^1 is less than or equal to its movement path cost.

On the other hand, the movement path weight and movement path cost of Π are the sum of the movement path weights and movement path costs of its sub-paths. It can be concluded from this fact and the results of the above two cases that the movement path weight of the path is less than or equal to its movement path cost. This completes the proof. ■

Theorem 3.3. *Assume the shortest path $\bar{\Pi}$ connecting the target to destination in a given combined energy digraph has the following properties:*

i) $\bar{\Pi}$ has at most two nodes in each Voronoi region it passes through.

ii) If Λ_k contains the nodes P_i and P_j, then $\bar{\Pi}$ does not pass through the regions containing the second nearest sensor to P_i or P_j.

Then, $\bar{\Pi}$ is the optimal path.

Proof. Suppose the shortest path $\bar{\Pi}$ and the optimal path Π^* are not the same. Then:

$$C(\Pi^*) < C(\bar{\Pi}) \tag{3.6}$$

From Theorem 3.1:

$$W(\bar{\Pi}) = C(\bar{\Pi}) \tag{3.7}$$

Also, from Theorem 3.2:

$$W(\Pi^*) \leq C(\Pi^*) \tag{3.8}$$

Combining the three relations given above, one arrives at the following inequality:

$$W(\Pi^*) < W(\bar{\Pi})$$

which contradicts the fact that $\bar{\Pi}$ is the shortest path. Thus, $\bar{\Pi}$ is the same as Π^*. ■

Corollary 3.2. *If the shortest path* $\bar{\Pi}$ *connecting the target to destination in the combined energy digraph has exactly one node in each Voronoi region it passes through, then* $\bar{\Pi}$ *is, in fact, the optimal path.*

Proof. The proof is straightforward, on noting that this is a special case of Theorem 3.3. ■

Remark 3.2. *To the best of the authors' knowledge, the problem of target tracking using a wireless sensor network with a sufficiently accurate energy-consumption model is not studied in the general form in a continuous-time setup. However, using the strategy proposed here, one can divide the field to a grid in order to transfer the problem to the discrete domain, where efficient techniques are available to solve it. One can use a larger grid to obtain smaller cells, which in turn leads to a more accurate solution to the underlying problem at the expense of higher computational complexity. Furthermore, the proposed strategy can also be quite effective in constrained trajectory tracking problems (e.g., obstacle avoidance).*

3.3 Discussion on Algorithm Performance and Efficiency

In this chapter, we proposed a centralized algorithm which is used by the network to find a cost-efficient route for the information from target to destination. In practice, sensors have limited power and processing capabilities, and this makes the motivation to avoid heavy computations by the sensors during network action. However,

since the present algorithm is centralized, significant parts of the computations can be done off-line easily. In fact, the communication energy digraphs can be fully constructed off-line. In addition, Although sensing and movement energy digraphs depend on target and sensors positions and they have to be updated in each time interval, parts of this update process still can be done off-line. One of the most important parts of the algorithm is the shortest path subroutine, which finds the shortest path connecting the target to the destination in the combined energy digraph. There are several algorithms to find this path in graphs. One of the most efficient algorithms among all, especially when the graph is not sparse, is Dijkstra algorithm, which we have used in our simulations. Results show that the algorithm finds the path in a reasonable time for the network to react. Simulation results, also, show that in more than 95 percent of the steps, the resulting shortest path satisfies the conditions of Theorem 3.3, thus, it is the optimal path. Furthermore, the algorithm is flexible in case more precision is needed for sensor locations, and this is achievable easily by increasing the number of grid points. In fact, one can make a trade off between the computational load and the precision of the algorithm by changing the number of the grid points.

3.4 Simulation Results

Consider a rectangular 30m $\times$ 20m field, and divide it into a 30×20 grid. Assume that there are 6 sensors in the field which are to follow the target and route the data from it to the destination in an energy-efficient manner. Assume also that all sensors have communication and sensing ranges of 10m and 1.5m, respectively. Let the respective movement, communication, and sensing energy consumption be:

$$W_m = \beta.d_{i,j}, \quad W_c = \alpha.d_{i,j}^{\lambda}, \quad W_s = \theta.d_{i,j}^{\gamma}$$

where α, β, and θ are given constants, and $d_{i,j}$ is the distance between nodes i and j (whose edge is to be weighted using the proposed procedure). For the simulations, assume $\alpha = 1$, $\beta = 50$, $\theta = 10$, $\lambda = 2$, and $\gamma = 4$.

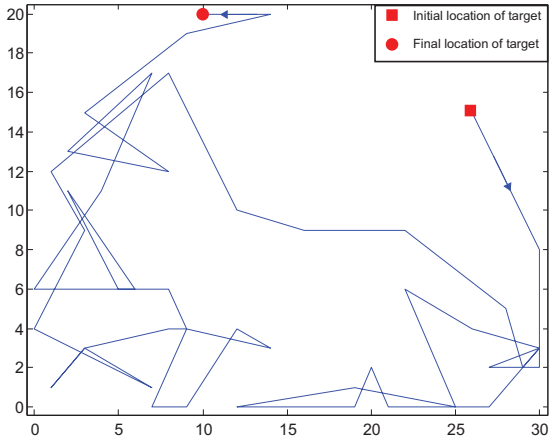


Figure 3.3: Random movement of the target (50 steps).

Let the movement of the target be random integer steps in the interval $[-7, 7]$ for both horizontal and vertical axes. The network processes the data in discrete time instants. This means that the proposed technique can be used at any time instant to determine the route and the new locations of the sensors to move to. The time interval between the consecutive time instants is chosen based on the target's speed.

Let the initial locations of the sensors be chosen randomly, with a uniform distribution on both horizontal and vertical axes. Let also the destination be at the origin. Simulations are performed for 50 steps of the target. Figure 3.3 shows the random movement of the target, and Figure 3.4 illustrates the tracking process in three snapshots: steps 1, 33 and 50. In each snapshot, the locations of the target and sensors, along with the shortest path and the Voronoi regions are depicted. The present locations of the sensors are shown by small circles, while their previous locations are depicted by asterisks. Moreover, the location of the target is shown by a solid square in each snapshot, and the shortest path obtained is drawn in dotted

line. Figure 3.5 depicts the results for three consecutive steps in the tracking process. It can be observed from this figure that the sensors are relocated properly in order to track the target continuously.

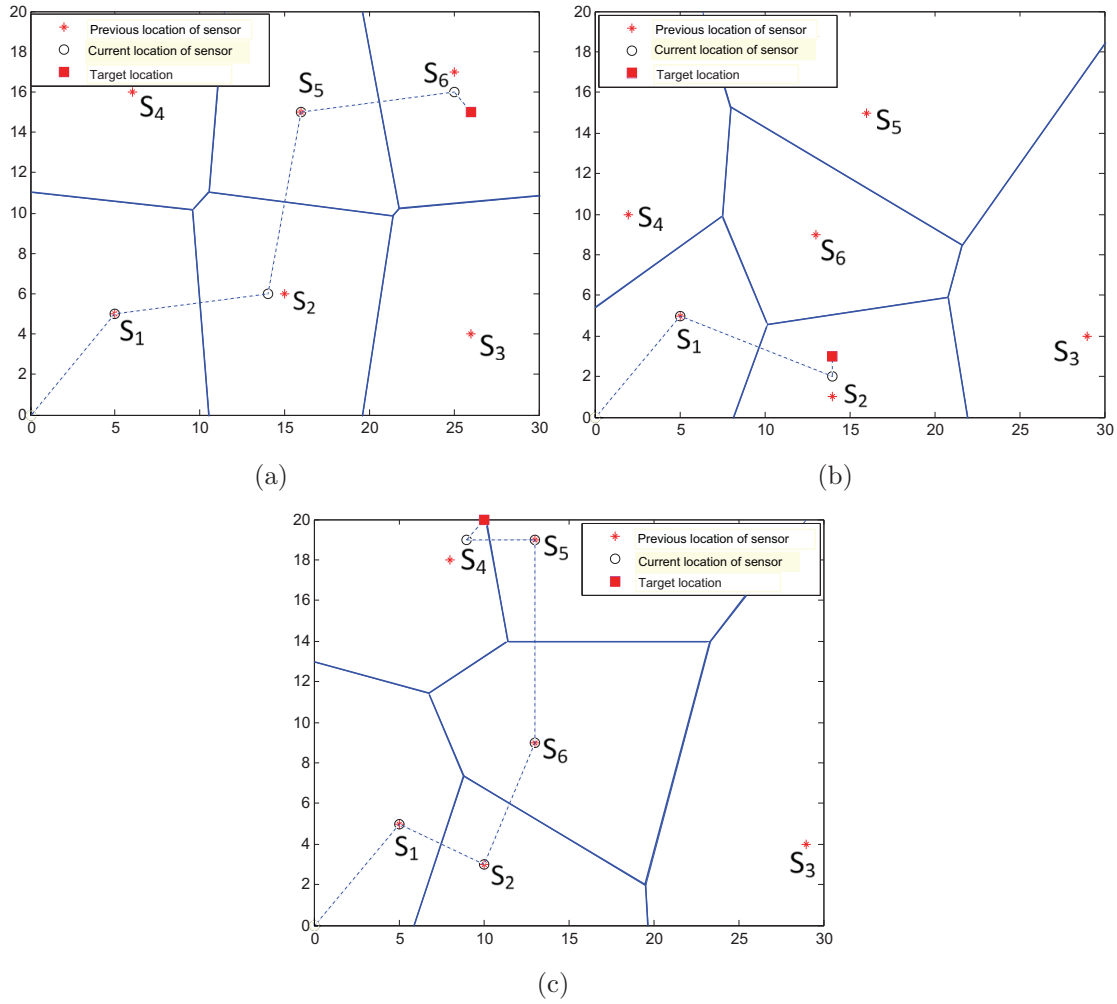


Figure 3.4: Snapshots of the network configuration obtained by the proposed technique for 6 sensors in three different steps: (a) 1^{st} step; (b) 33^{rd} step, and (c) 50^{th} step.

It is desired now to show the tracking performance for a larger number of sensors, with the same sensing and communication ranges as in the previous case. It can be observed from Figure 3.6 that for the case of 24 sensors, the only sensor that is required to move under the proposed technique is the one assigned to detect the target. The result is not surprising, as in this case the connection between the

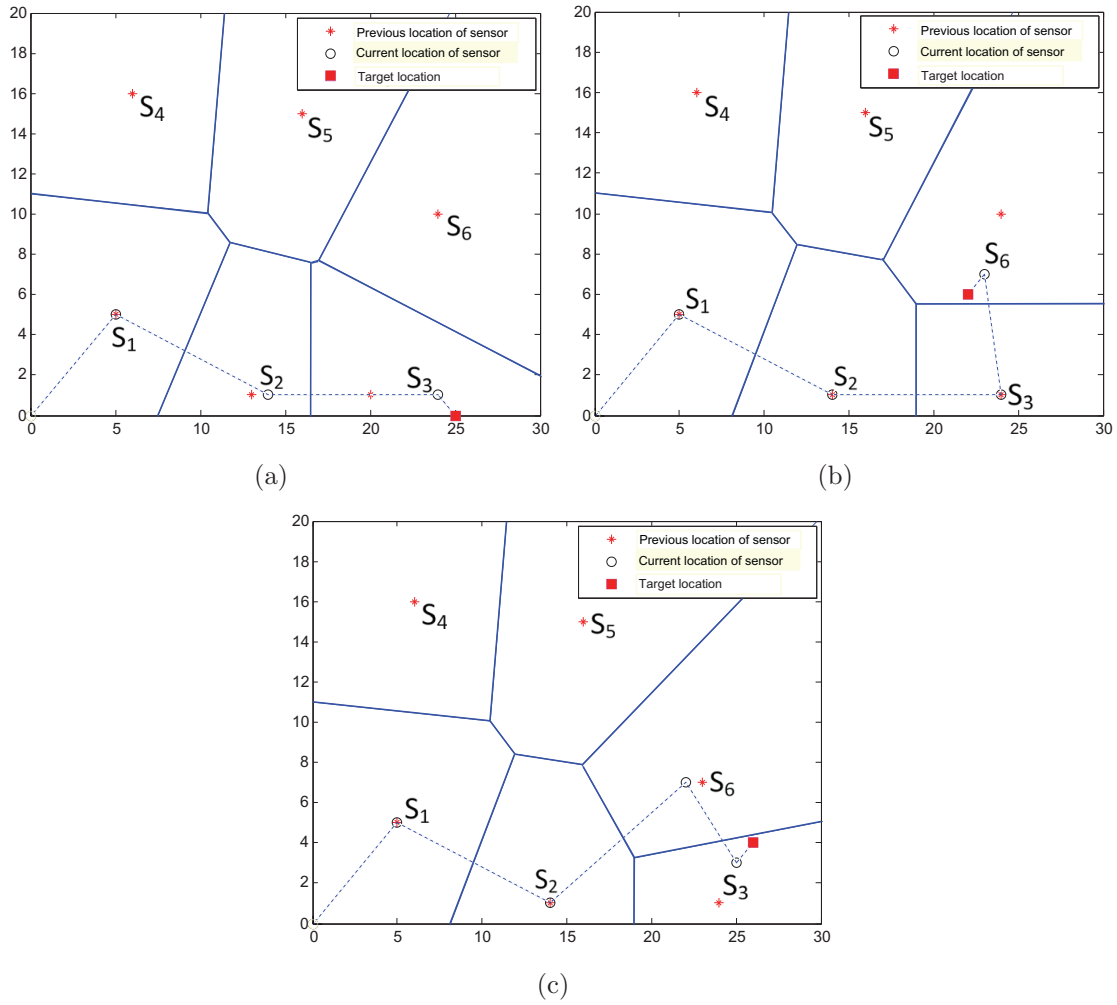


Figure 3.5: Snapshots of the network configuration obtained by the proposed technique for 6 sensors in three consecutive steps: (a) 13^{th} step; (b) 14^{th} step, and (c) 15^{th} step.

target and destination can be established through different routes, and hence there is no need to move the sensors for this purpose (note that the movement energy is typically greater than sensing and communication energies).

Remark 3.3. *As simulation results show, although conditions of Theorem 3.3 seem to be strong, our method finds the optimal path in more than 95 percent of the cases.*

Remark 3.4. *It can be verified that in all of the snapshots provided in the simulations, the shortest path is the same as the optimal path. This is not a coincidence,*

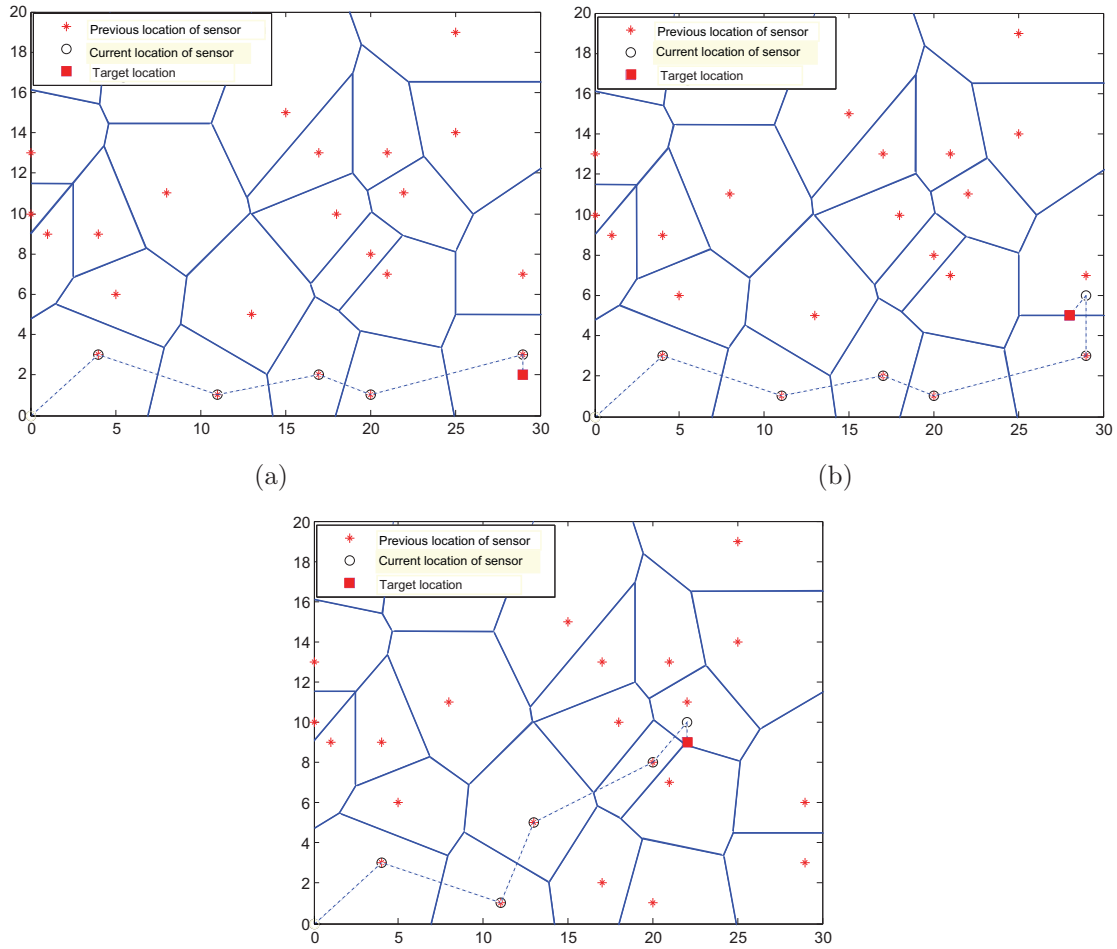


Figure 3.6: Snapshots of the network configuration obtained by the proposed technique for 24 sensors in three consecutive steps: (a) 17^{th} step; (b) 18^{th} step, and (c) 19^{th} step.

and is true for typical network configurations (where, for example, one Voronoi regions is not significantly larger than another one).

Chapter 4

Distributed Deployment Algorithms for Improved Coverage in a Network of Wireless Mobile Sensors

In this chapter, efficient sensor deployment strategies are developed to increase coverage in wireless mobile sensor networks. The sensors find coverage holes within their Voronoi polygons, and then move in an appropriate direction to minimize them. Novel edge-based and vertex-based strategies are introduced, and their performances are compared with existing techniques. The proposed movement strategies are based on the distances of each sensor and the points inside its Voronoi polygon from the edges or vertices of the polygon. It is shown that the methods introduced in this work outperform existing strategies. Simulations confirm the effectiveness of the proposed deployment algorithms and their superiority to the techniques reported in the literature.

The plan of the chapter is as follows. In Section 4.1, preliminary material

concerning the Voronoi diagram is provided and its properties are briefly discussed. Section 4.2 presents the proposed algorithms for efficient coverage, as the main contribution of the chapter. Finally, in Section 4.3, simulation results are given to show the effectiveness of the proposed approaches.

4.1 Preliminaries

Consider a flat polygon-shaped surface and a set of networked sensors denoted by $\mathbf{S} := \{S_1, S_2, ..., S_n\}$. Let the network be represented by a graph, where each node denotes a sensor. Partition the plane into n convex polygons such that each polygon contains only one node and any point inside each polygon is closer to its generating node than to any other node in the plane. The resultant diagram is called a Voronoi diagram, and each individual cell in it is referred to as a Voronoi polygon (or region). An example of a Voronoi diagram for a network of 15 sensors is depicted in Fig. 4.1.

The Voronoi region Π_i generated by S_i can be mathematically formulated as (see [123], [124]):

$$\Pi_i = \left\{ X \in \mathbb{R}^2 \mid d(X, Y_i) \ \leq \ d(X, Y_j), \quad j \in \mathbf{n} := \{1, \cdots, n\}, \ i \neq j \right\} \tag{4.1}$$

where Y_i is the coordinate of S_i, and $d(X, Y_i)$ denotes the Euclidean distance between the points X and Y_i in the 2D plane. To construct the Voronoi diagram, the bisectors of each node and its neighbors need to be drawn first. Among all polygons generated by these bisectors, the smallest one which contains the node is the Voronoi polygon of that node. It follows from (4.1) that any point in a Voronoi polygon which is not detected by the sensor associated with that polygon, cannot be detected by any other sensor either. Thus, in order to find the so called "coverage holes", i.e. the points that are not detected by any sensor in the network, each sensor would only need to check its own Voronoi polygon. The Voronoi diagram is used for the analysis and synthesis of sensor deployment algorithms in this chapter.

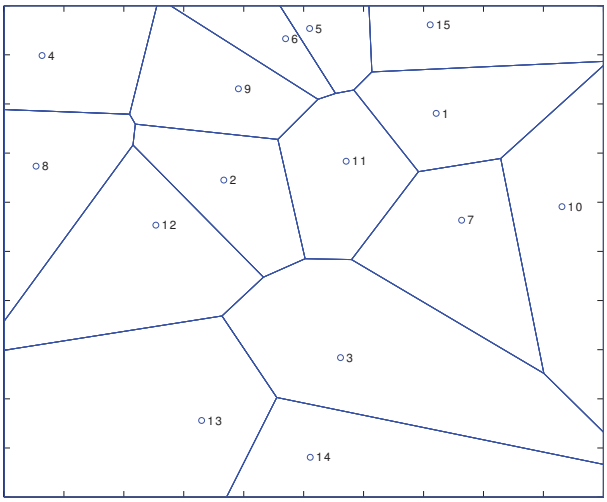


Figure 4.1: An example of a Voronoi diagram

Definition 4.1. *A pair of nodes whose Voronoi polygons share an edge are referred to as* neighbors.

Definition 4.2. *Consider a sensor S_i with the sensing radius r and the corresponding Voronoi polygon Π_i, $i \in \mathbf{n}$, and let Q be an arbitrary point inside Π_i. The intersection of the polygon Π_i and a circle of radius r centered at Q is referred to as the* i-th coverage area w.r.t. Q*, and is denoted by $\beta^Q_{\Pi_i}$. The i-th coverage area w.r.t. the location of the sensor S_i is called the local coverage area of that sensor.*

Definition 4.3. *Consider an arbitrary point Q inside the Voronoi polygon Π_i, $i \in \mathbf{n}$. The area inside the Voronoi polygon Π_i which lies outside the i-th coverage area w.r.t. Q is referred to as the* i-th coverage hole w.r.t. Q*, and is denoted by $\theta^Q_{\Pi_i}$. The i-th coverage hole w.r.t. the location of the sensor S_i is called the local coverage hole of that sensor. Also, the union of all local coverage holes in the sensing field is referred to as the* total coverage hole*, and is denoted by θ, i.e. $\theta = \sum_{i=1}^{n} \theta^{P_i}_{\Pi_i}$, where P_i denotes the location of the sensor S_i.*

Assumption 4.1. *In this chapter, it is assumed that there is no obstacle in the field. This means that every sensor can move to any desired location using existing techniques, e.g. [64], [65], [125], [97].*

Assumption 4.2. *All sensors are assumed to be capable of locating themselves in the field (using, for instance, the methods proposed in [126], [127]). Moreover, the localization error of every sensor is assumed to be negligible [64], [97].*

Assumption 4.3. *It is assumed that the graph representing sensors' communication topology is connected [128]. Hence, each sensor can obtain the information about the locations of the other sensors through proper communication routes, and consequently calculate its Voronoi polygon accurately (using the position information of its neighbors). Note that this is a realistic assumption as the number of sensors in a mobile sensor network is typically large (or more precisely, there is a sufficient number of sensors per area unit) [129], [130].*

Problem Statement: In this work, it is desired that each sensor finds a candidate location for itself using the available local information, and moves to this new position such that the total coverage of the network increases (or, equivalently, the total coverage hole decreases).

4.2 Main Results

Four efficient sensor relocation algorithms are introduced in this section to increase sensing coverage in a mobile sensor network. The main characteristic of these algorithms is that the sensor movement is performed iteratively until the termination condition is satisfied. Each round in the proposed algorithms consists of four phases. In the first phase, every sensor S_i, $i \in \mathbf{n}$, broadcasts its location information to other sensors, and then constructs its Voronoi polygon based on the similar information it receives from other sensors. Then in the second phase, each sensor checks its polygon for possible coverage holes. If any coverage hole exists, the sensor finds a target location $\acute{P}_i$ for itself (but does not move there) using an appropriate scheme, such that by moving there the coverage hole would be eliminated, or at least its size

would be reduced by a certain threshold. Once the new target location is calculated, the coverage area w.r.t. this location, i.e. $\beta_{\Pi_i}^{\acute{P}_i}$, is obtained in the third phase. If this coverage area is greater than the current local coverage area, i.e. $\beta_{\Pi_i}^{\acute{P}_i} > \beta_{\Pi_i}^{P_i}$, the sensor moves to the new destination; otherwise, it remains at its current location. Finally, in the termination phase, if none of the sensors' local coverage area in its Voronoi polygon would be increased by a certain amount, the iterations stop. This termination condition guarantees that the proposed algorithms stop in finite time.

As noted above, one of the important characteristics of the sensor deployment strategies proposed in this chapter is that each sensor moves to a new location only if its coverage area w.r.t. the new location in the old Voronoi polygon increases. The following theorem shows that the total coverage is increased under this type of deployment scheme.

Theorem 4.1. *Consider the set* $\mathbf{S}$ *of* n *sensors described in the previous section, and let the position of the* i*-th sensor be denoted by* P_i*, with the corresponding Voronoi polygon* Π_i*. Assume the* i*-th sensor moves to a new position* $\acute{P}_i$*, for any* $i \in \mathbf{n}$*, with the corresponding Voronoi polygon* $\acute{\Pi}_i$ *such that* $\acute{P}_i \neq P_i$ *if and only if* $i \in \mathbf{k}$*, where* $\mathbf{k}$ *is a non-empty subset of* $\mathbf{n}$*. If the* i*-th coverage area w.r.t.* $\acute{P}_i$ *in the previously constructed Voronoi polygon* Π_i *is greater than the* i*-th local coverage area in* Π_i *(i.e.,* $\beta_{\Pi_i}^{\acute{P}_i} > \beta_{\Pi_i}^{P_i}$*) for all* $i \in \mathbf{k}$*, then the total coverage in the network increases.*

Proof. Let the total uncovered area of the sensing field when the sensors are located at the positions $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ and $\acute{\mathbf{P}} = \{\acute{P}_1, \acute{P}_2, \ldots, \acute{P}_n\}$ be denoted by θ and $\acute{\theta}$, respectively. From the characterization of the Voronoi diagram, one can write:

$$\theta = \sum_{i=1}^{n} \theta_{\Pi_i}^{P_i} \tag{4.2}$$

It is straightforward to show that for any $i \in \mathbf{k}$, if the coverage area in Π_i increases, then the corresponding coverage hole will become smaller. Since it is assumed that

the i-th coverage area w.r.t. $\acute{P}_i$ is greater than the i-th local coverage area for any $i \in \mathbf{k}$, one can conclude that:

$$\theta_{\Pi_i}^{\acute{P}_i} < \theta_{\Pi_i}^{P_i}, \quad \forall i \in \mathbf{k} \tag{4.3}$$

In addition, note that if $\acute{P}_i = P_i$, then:

$$\theta_{\Pi_i}^{\acute{P}_i} = \theta_{\Pi_i}^{P_i}, \quad \forall i \in \mathbf{n} \backslash \mathbf{k} \tag{4.4}$$

On the other hand, it is possible that part of the area in $\theta_{\Pi_i}^{\acute{P}_i}$ is also covered by some other sensors in the set $\in \mathbf{n} \backslash \{i\}$. Hence:

$$\acute{\theta} \leq \sum_{i=1}^{n} \theta_{\Pi_i}^{\acute{P}_i} \tag{4.5}$$

Furthermore, from (4.3), (4.4) and (4.5), one arrives at the following inequality:

$$\acute{\theta} < \sum_{i=1}^{n} \theta_{\Pi_i}^{P_i} \tag{4.6}$$

Now, it is concluded from (4.2) and (4.6) that:

$$\acute{\theta} < \theta \tag{4.7}$$

which means that the total coverage area increases using the proposed deployment scheme. ■

4.2.1 The Maxmin-Vertex Strategy

The rationale behind the Maxmin-vertex strategy is that when the sensors are evenly distributed, none of them should be too close to any of its Voronoi vertices. In this strategy, a point inside the Voronoi polygon whose distance from the nearest Voronoi vertex is maximized is selected as the candidate destination point. This point will be referred to as the *Maxmin-vertex centroid*, and will be denoted by $\bar{O}$. Let the distance between this point and the nearest vertex to it on the polygon be represented by $\bar{r}$. Let also $C(O, r)$ denote a circle of radius r centered at the point O. The Maxmin-vertex circle is defined next.

Definition 4.4. *The Maxmin-vertex circle of a polygon is defined as the largest circle centered inside the polygon such that all of the vertices of the polygon are either outside the circle, or on it. This circle is, in fact,* $C(\bar{O}, \bar{r})$.

Lemma 4.1. *The Maxmin-vertex circle passes through at least two Voronoi vertices.*

Proof. Let $\bar{V}$ be the nearest vertex of the i-th polygon to its Maxmin-vertex centroid $\bar{O}$, and define:

$$\hat{u} := \min_{V \in \mathbf{V}_i - \{\bar{V}\}} \left\{ d(\bar{O}, V) \right\}, \quad i \in \mathbf{n} \tag{4.8}$$

where $\mathbf{V}_i$ is the set of all vertices of polygon i in the Voronoi diagram. Suppose that the Maxmin-vertex circle does not pass through any vertex other than $\bar{V}$, and hence $\delta^* = (\hat{u} - \bar{r})/2$ is positive. There are two possibilities, as discussed below.

Case 1: $\bar{O}$ *is inside the polygon.* Let $\hat{O}$ be a point on the line $\bar{V}\bar{O}$, but closer to $\bar{O}$, such that the distance between $\bar{O}$ and $\hat{O}$ is equal to δ, where δ is an arbitrary value in $(0, \delta^*]$ (see Fig. 4.2(a)).

Case 2: $\bar{O}$ *is on the polygon.* Suppose $\bar{O}$ is on the edge ϵ. Let $\hat{O}$ be a point on ϵ such that $d(\hat{O}, \bar{V}) > d(\bar{O}, \bar{V})$ and the distance between $\bar{O}$ and $\hat{O}$ is equal to δ, where δ is an arbitrary value in the interval $(0, \delta^*]$ (see Fig. 4.2(b)).

In both cases, according to the triangle inequality:

$$d(\hat{O}, V) \geq d(\bar{O}, V) - \delta \geq \hat{u} - \delta, \quad \forall V \in \mathbf{V}_i - \{\bar{V}\},\ i \in \mathbf{n} \tag{4.9}$$

From the above relation and on nothing that $\hat{u} - \delta \geq \bar{r} + \delta > \bar{r}$, it can be concluded that

$$\min_{V \in \mathbf{V}_i} \left\{ d(\hat{O}, V) \right\} > \bar{r} \tag{4.10}$$

which contradicts the fact that $\bar{O}$ is the Maxmin-vertex centroid. Thus, there is at least one more vertex on the Maxmin-vertex circle, and this completes the proof. ■

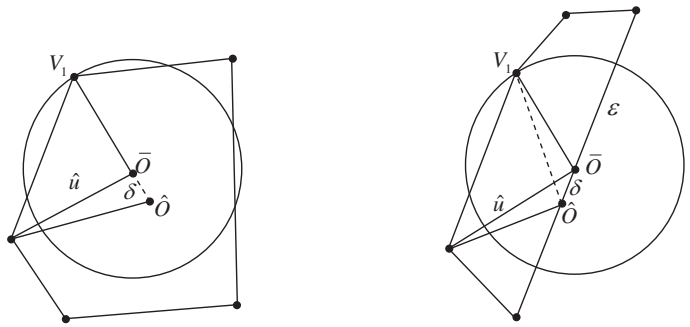


Figure 4.2: An illustrative example of a Voronoi polygon and the corresponding Maxmin-vertex circle when the Maxmin-vertex centroid is: (a) inside the polygon, and (b) on the polygon.

Lemma 4.2. *If the Maxmin-vertex circle passes through exactly two Voronoi vertices, say $\bar{V}_1$ and $\bar{V}_2$, then $\bar{O}$ is the intersection of the perpendicular bisector of $\bar{V}_1\bar{V}_2$ and an edge of the polygon.*

Proof. Suppose $\bar{O}$ is not the intersection of the perpendicular bisector of $\bar{V}_1\bar{V}_2$ and an edge of the polygon, i.e., $\bar{O}$ is inside the polygon. Define:

$$\tilde{u} := \min_{V \in \mathbf{V}_i - \{\bar{V}_1, \bar{V}_2\}} \left\{ d(\bar{O}, V) \right\}, \ i \in \mathbf{n} \tag{4.11}$$

Since $C(\bar{O}, \bar{r})$ passes through exactly two vertices, thus $\delta^* = (\tilde{u} - \bar{r})/2$ is positive. Let $\tilde{O}$ be a point on the perpendicular bisector of $\bar{V}_1\bar{V}_2$ and outside the triangle $\bar{V}_1\bar{V}_2\bar{O}$, but closer to $\bar{O}$, such that the distance between the points $\bar{O}$ and $\tilde{O}$ is equal to δ, where δ is an arbitrary value in the interval $(0, \delta^*]$ (see Fig. 4.3). Using the triangle inequality, one can write:

$$d(\tilde{O}, V) \geq d(\bar{O}, V) - \delta \geq \tilde{u} - \delta \tag{4.12}$$

The above result along with the relations $\tilde{u} - \delta \geq \tilde{u} - \delta^* = \bar{r} + \delta^* > \bar{r}$ and $d(\tilde{O}, \bar{V}_1) = d(\tilde{O}, \bar{V}_2) > \bar{r}$ yields:

$$\min_{V \in \mathbf{V}_i} \left\{ d(\tilde{O}, V) \right\} > \bar{r}, \ i \in \mathbf{n} \tag{4.13}$$

which contradicts the fact that $\bar{O}$ is the Maxmin-vertex centroid. ■

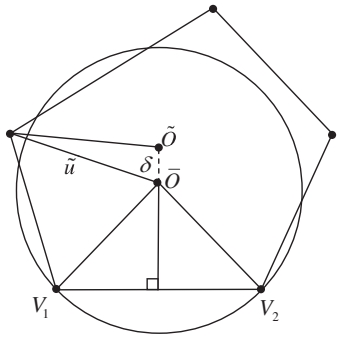


Figure 4.3: An illustrative figure used in the proof of Lemma 4.2.

Definition 4.5. *For convenience of notation, the circle passing through two vertices V_p and V_q of polygon i, centered at the intersection of the perpendicular bisector of V_pV_q and the edge V_kV_l is denoted by $\Omega^{k,l}_{p,q}$, $k,l,p,q \in \mathbf{m}_i := \{1,...,m_i\}$, where m_i is the number of vertices of the i-th polygon, for any $i \in \mathbf{n}$. Also, the circle passing through three vertices V_p, V_q and V_r of polygon i is denoted by $\Omega_{p,q,r}$, for $p,q,r \in \mathbf{m}_i$.*

Theorem 4.2. *For any $k,l,p,q \in \mathbf{m}_i$, let $\breve{\mathbf{C}}_i$ be the set of all circles $\Omega^{k,l}_{p,q}$ whose centers are on polygon i, and do not enclose any of the vertices of the polygon, and $\grave{\mathbf{C}}_i$ be the set of all circumcircles of any three vertices, centered inside or on the polygon, which do not enclose any of the vertices of the polygon. Define $\mathbf{C}_i := \breve{\mathbf{C}}_i \cup \grave{\mathbf{C}}_i$. Then $C(\bar{O},\bar{r}) \in \mathbf{C}_i$, and also for all $C(O,r) \in \mathbf{C}_i$, $r \leq \bar{r}$.*

Proof. According to Lemma 4.1, the Maxmin-vertex circle passes through at least two Voronoi vertices. If it passes through exactly two Voronoi vertices, say V_1, V_2, then according to Lemma 4.2 there exist $k,l \in \mathbf{m}_i$ such that $C(\bar{O},\bar{r}) = \Omega^{k,l}_{1,2}$. Hence, in this case $C(\bar{O},\bar{r}) \in \mathbf{C}_i$, and from Definition 4.4, $\bar{r} = \max_{C(O,r)\in\mathbf{C}_i}\{r\}$. If, on the other hand, the Maxmin-vertex circle passes through three or more Voronoi vertices, then it is the circumcircle of those vertices. Therefore, $C(\bar{O},\bar{r}) \in \mathbf{C}_i$, and again it is deduced from Definition 4.4 that $\bar{r} = \max_{C(O,r)\in\mathbf{C}_i}\{r\}$. ■

Using the result of Theorem 4.2, one can develop an algorithm of complexity $O(m_i^4)$ to calculate the Maxmin-vertex centroid in Voronoi polygon i. Since typically

a Voronoi polygon does not have too many vertices, the computational complexity of such an algorithm is not expected to be high, typically. Detailed steps are presented in Algorithm 1.

Algorithm 1: Finding the Maxmin-vertex centroid of the i-th Voronoi polygon

begin

1) **for** $p = 1, 2, \ldots, m_i - 2$
   for $q = p + 1, p + 2, \ldots, m_i - 1$
   for $r = q + 1, q + 2, \ldots, m_i$
   calculate $\Omega_{p,q,r}$
   if $\Omega_{p,q,r}$ is centered inside or on the polygon and does not enclose any of the vertices of the polygon, **then**
   record it.
   end
   end
   end
   end
2) **for** $p = 1, 2, \ldots, m_i - 1$
   for $q = p + 1, p + 2, \ldots, m_i$
   calculate $\Omega_{p,q}^{k,l}$
   if $\Omega_{p,q}^{k,l}$ is centered on the polygon and does not enclose any of the vertices of the polygon, **then**
   record it.
   end
   end
   end
3) The center of the largest circle is the Maxmin-vertex centroid of the polygon.

The sensor deployment technique discussed above as well as the two algorithms given in [64] are all vertex-based, in the sense that they are concerned with the distances of the nodes from the vertices of the Voronoi diagram. While algorithms of this type prove effective in many cases, they may not be as effective for certain node configurations. For instance, consider the polygon in Fig. 7.2, and let the sensor be placed at point S. It is easy to verify that in order to increase the coverage area,

the sensor must move to the left. However, both VOR and Minimax algorithms proposed in [64] consider the candidate points A and B, respectively, which are in the right side of S. To remedy this shortcoming of the vertex-based algorithms, two edge-based techniques will be presented in the next subsection.

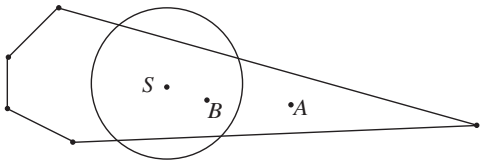


Figure 4.4: An example of a configuration for which the vertex-based strategies are not as effective.

4.2.2 Minmax-Edge Strategy

The rationale behind the Minmax-edge technique is that when the sensors are evenly distributed, none of them should be too far from any of its Voronoi edges. The Minmax-edge strategy chooses the target location of the sensor S_i as a point inside Voronoi polygon i whose distance from the farthest Voronoi edge is minimized. This point will be referred to as the *Minmax-edge centroid*, and will be denoted by $\acute{O}$. Furthermore, the distance between this point and the farthest edge on Voronoi polygon i will be represented by $\acute{r}$. In the remainder of this subsection, intersecting or tangent to or touching an edge means intersecting or tangent to or touching that edge or its extension. The Minmax-edge circle is defined next.

Definition 4.6. *The Minmax-edge circle is the smallest circle centered inside or on a polygon, intersecting or touching all of its edges. This circle is in fact* $C(\acute{O}, \acute{r})$*, and is not necessarily unique (this issue will be addressed later).*

Lemma 4.3. *Consider two points A, B and a line* Δ*. Let the distance between A and* Δ *be denoted by* σ*, and that between B and* Δ *by* ρ*. Let also the length of the segment AB be denoted by* ξ*. Then:*

$$\sigma - \xi \leq \rho \leq \sigma + \xi \tag{4.14}$$

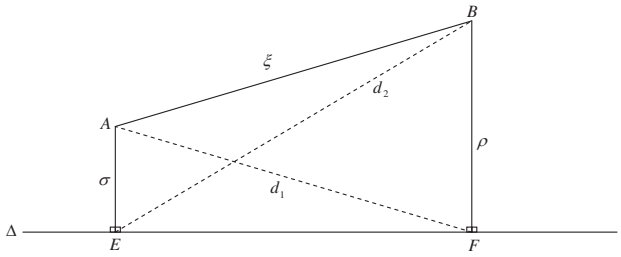


Figure 4.5: An illustrative figure used in the proof of Lemma 4.3.

Proof. Let E and F be two points on Δ, such that $AE \perp \Delta$ and $BF \perp \Delta$. Let $d(A, F) = d_1$ and $d(B, E) = d_2$ (see Fig. 4.5). According to the triangle inequality:

$$\sigma \leq d_1 \leq \rho + \xi \tag{4.15}$$

$$\rho \leq d_2 \leq \sigma + \xi \tag{4.16}$$

Relation (4.14) follows directly from (4.15) and (4.16). ■

Lemma 4.4. *The Minmax-edge circle is tangent to at least two of the edges of its Voronoi polygon.*

Proof. Let $\acute{\epsilon}$ be the farthest edge from the Minmax-edge centroid of a given Voronoi polygon. It is obvious that $\acute{r}$ is equal to the distance between $\acute{O}$ and $\acute{\epsilon}$, denoted by $d(\acute{O}, \acute{\epsilon})$. Thus, $C(\acute{O}, \acute{r})$ is tangent to $\acute{\epsilon}$. Define:

$$\hat{v} := \max_{\epsilon \in \mathbf{E}_i - \{\acute{\epsilon}\}} \left\{ d(\acute{O}, \epsilon) \right\}, \ i \in \mathbf{n} \tag{4.17}$$

where $\mathbf{E}_i$ represents the set of all edges of polygon i, and suppose that the Minmax-edge circle is not tangent to any other edge, implying that $\delta = (\acute{r} - \hat{v})/2$ is positive. Let M be a point on $\acute{\epsilon}$ or its extension, such that $M\acute{O} \perp \acute{\epsilon}$. Let also $\hat{O}$ be a point on $M\acute{O}$ such that $\acute{O}\hat{O} = \delta$ (for example, see Fig. 4.6). According to Lemma 4.3:

$$d(\hat{O}, \epsilon) \leq d(\acute{O}, \epsilon) + \delta \leq \hat{v} + \delta, \ \forall \epsilon \in \mathbf{E}_i - \{\acute{\epsilon}\} \tag{4.18}$$

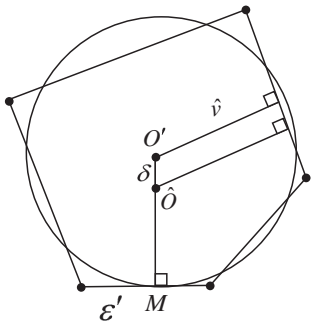


Figure 4.6: An illustrative figure used in the proof of Lemma 4.4.

From (4.18) and the relation $\hat{v} + \delta = \acute{r} - \delta < \acute{r}$, one can conclude that:

$$\max_{\epsilon \in \mathbf{E}_i} \left\{ d(\hat{O}, \epsilon) \right\} < \acute{r}, \ i \in \mathbf{n} \tag{4.19}$$

which contradicts the fact that $\acute{O}$ is the Minmax-edge centroid. This completes the proof. ■

Lemma 4.5. *Given a Voronoi diagram, assume that the i-th Voronoi polygon has at least three edges. Then, the Minmax-edge circle of this polygon is tangent to at least two edges. Furthermore, if the Minmax-edge circle is tangent to exactly two edges, say ϵ_1 and ϵ_2, then at least one of the following conditions holds:*
i) the two edges ϵ_1 and ϵ_2 are parallel, or
ii) the centroid $\acute{O}_i$ is the intersection of the bisector of the angle between ϵ_1, ϵ_2, and one of the edges of the polygon.

Proof. Suppose the Minmax-edge circle is tangent to exactly two non-parallel Voronoi edges ϵ_1 and ϵ_2, but $\acute{O}$ is not the intersection of the bisector of the angle between ϵ_1 and ϵ_2 and an edge of the polygon; i.e., $\acute{O}$ is inside the polygon. Define:

$$\tilde{v} := \max_{\epsilon \in \mathbf{E}_i - \{\epsilon_1, \epsilon_2\}} \left\{ d(\acute{O}, \epsilon) \right\}, \ i \in \mathbf{n} \tag{4.20}$$

Since $C(\acute{O}, \acute{r})$ is tangent to exactly two edges, thus $\delta^* = (\acute{r} - \tilde{v})/2$ is positive. Let the point F be the intersection of ϵ_1 and ϵ_2 (or their extensions). Let also $\tilde{O}$ be a point on $F\acute{O}$ such that $\acute{O}\tilde{O} = \delta$, where δ is an arbitrary value in the interval $(0, \delta^*]$ (as an example, see Fig. 4.7). According to Lemma 4.3:

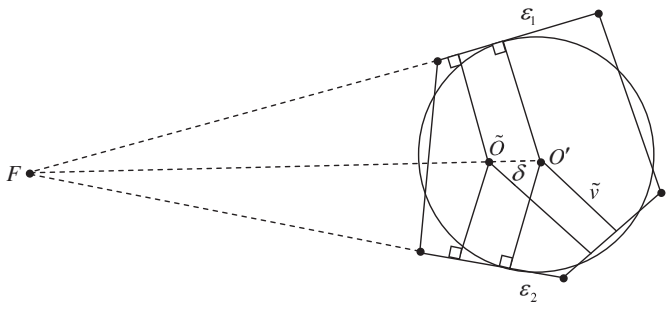


Figure 4.7: An illustrative figure used in the proof of Lemma 4.5.

$$d(\tilde{O},\epsilon) \leq d(\acute{O},\epsilon) + \delta \leq \tilde{v} + \delta,\ \forall \epsilon \in \mathbf{E}_i - \{\epsilon_1, \epsilon_2\},\ i \in \mathbf{n} \tag{4.21}$$

It results from (4.21) and the relations $\tilde{v} + \delta \leq \acute{r} - \delta < \acute{r}$ and $d(\tilde{O},\epsilon_1) = d(\tilde{O},\epsilon_2) < \acute{r}$ that:

$$\max_{\epsilon \in \mathbf{E}_i} \left\{ d(\tilde{O},\epsilon) \right\} < \acute{r},\ i \in \mathbf{n} \tag{4.22}$$

which contradicts the fact that $\acute{O}$ is the Minmax-edge centroid. On the other hand, if the Minmax-edge circle is not touching exactly two Voronoi edges, then according to Lemma 4.4 it is tangent to at least three Voronoi edges. This completes the proof. ■

Lemma 4.6. *If a Minmax-edge circle is tangent to two parallel edges, then there will generically be other Minmax-edge circles, all of which are also tangent to these parallel edges.*

Proof. Suppose one Minmax-edge circle, say C_1, is tangent to two parallel edges, say ϵ_1 and ϵ_2, but there exists another Minmax-edge circle, say C_2, that is not tangent to these two edges. Let the distance between ϵ_1 and ϵ_2 be denoted by $d(\epsilon_1, \epsilon_2)$. It is obvious that the radius of the circle C_1 is equal to $\frac{d(\epsilon_1,\epsilon_2)}{2}$. This implies that the radius of the circle C_2 must be grater than $\frac{d(\epsilon_1,\epsilon_2)}{2}$, which contradicts the initial assumption that C_2 is a Minmax-edge circle. ■

Remark 4.1. *In the case when all Minmax-edge circles are tangent to two parallel edges, some of these circles are tangent to three or more edges. In this case, one of such circles is arbitrarily chosen as the Minmax-edge circle.*

Definition 4.7. *For convenience of notation, the circle touching two edges ϵ_g and ϵ_h of polygon i, centered at the intersection of the edge ϵ_k and the bisector of the angle between ϵ_g and ϵ_h is denoted by $\Omega^k_{g,h}$, for any $k, g, h \in \mathbf{e}_i := \{1, \ldots, e_i\}$, where e_i is the number of edges of polygon i in the Voronoi diagram. Also, the circle touching three edges ϵ_f, ϵ_g and ϵ_h of polygon i is denoted by $\Omega^{f,g,h}$, for $f, g, h \in \mathbf{e}_i$.*

Theorem 4.3. *Let $\breve{\mathbf{D}}_i$ be the set of all circles $\Omega^k_{g,h}, \forall k, g, h \in \mathbf{e}_i$, such that: (i) their centers lie inside or on the i-th polygon, and (ii) they intersect or are tangent to all edges of the polygon. Let also $\grave{\mathbf{D}}_i$ be the set of all circles such that: (i) they are tangent to at least three edges of a Voronoi polygon; (ii) their centers lie inside or on the i-th polygon, and (iii) they intersect or are tangent to all edges of the polygon. Define $\mathbf{D}_i := \breve{\mathbf{D}}_i \cup \grave{\mathbf{D}}_i$; then the Minmax-edge circle belongs to $\mathbf{D}_i$, and is the smallest circle in this set.*

Proof. The proof follows directly from Lemmas 4.5 and 4.6, and Remark 4.1. ■

Using the result of Theorem 4.3, the following algorithm is developed to find the Minmax-edge centroid in the i-th Voronoi polygon. The computational complexity of this algorithm is $O(e_i^4)$, which is typically not too high.

4.2.3 Maxmin-Edge Strategy

Similar to the two methods introduced so far, the idea behind this strategy is that when the sensors are evenly distributed, none of them should be too close to any of its Voronoi edges. The target location of a sensor under the Maxmin-edge strategy is a point inside the corresponding Voronoi polygon whose distance from the nearest Voronoi edge is maximized. This point will be referred to as the *Maxmin-edge centroid*, and will be denoted by $\breve{O}$. Furthermore, the distance between this point and the nearest edge to it will be represented by $\breve{r}$. The Maxmin-edge circle is defined next.

Algorithm 2: Finding the Minmax-edge centroid of the i-th Voronoi polygon

begin

1) **for** $f = 1, 2, \ldots, e_i - 2$
   for $g = f + 1, f + 2, \ldots, e_i - 1$
   for $h = g + 1, g + 2, \ldots, e_i$
   calculate $\Omega^{f,g,h}$
   if $\Omega^{f,g,h}$ is centered inside or on the polygon and it intersects or is tangent to all edges of the polygon, **then**
   record it.
   end
   end
   end
   end
2) **for** $g = 1, 2, \ldots, e_i - 1$
   for $h = g + 1, g + 2, \ldots, e_i$
   calculate $\Omega^{k}_{g,h}$
   if $\Omega^{k,l}_{p,q}$ is centered inside or on the polygon and it intersects or is tangent to all edges of the polygon, **then**
   record it.
   end
   end
   end
3) The center of the smallest circle is the Minmax-edge centroid of the polygon.

Definition 4.8. *The Maxmin-edge circle of a polygon is the largest circle inside the polygon. This circle is, in fact,* $C(\breve{O}, \breve{r})$.

Lemma 4.7. *The Maxmin-edge circle is tangent to at least two of the Voronoi edges.*

Proof. Consider a Voronoi polygon, and let $\breve{\epsilon}$ be the nearest edge to the Maxmin-edge centroid of the polygon. The radius $\breve{r}$ is equal to the distance between $\breve{O}$ and $\breve{\epsilon}$, i.e. $d(\breve{O}, \breve{\epsilon})$. Thus, $C(\breve{O}, \breve{r})$ is tangent to $\breve{\epsilon}$. Define:

$$\hat{w} = \min_{\epsilon \in \mathbf{E}_i - \{\breve{\epsilon}\}} \left\{ d(\breve{O}, \epsilon) \right\}, \ i \in \mathbf{n} \tag{4.23}$$

and suppose that the Maxmin-edge circle is not tangent to any other edge, implying that $\delta^* = (\breve{w} - \breve{r})/2$ is positive. Let M be a point on $\breve{\epsilon}$, such that $M\breve{O} \perp \breve{\epsilon}$. Let also $\hat{O}$ be a point on $M\breve{O}$ such that $\breve{O}\hat{O} = \delta$, where δ is an arbitrary value in the interval $(0, \delta^*]$ (as an example, see Fig. 4.8). According to Lemma 4.3:

$$d(\hat{O}, \epsilon) \geq d(\breve{O}, \epsilon) - \delta \geq \hat{w} - \delta, \ \forall \epsilon \in \mathbf{E}_i - \{\breve{\epsilon}\} \tag{4.24}$$

From (4.24) and the relation $\hat{w} - \delta \geq \breve{r} + \delta > \breve{r}$, one can conclude that:

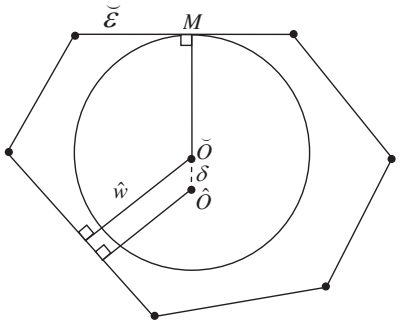


Figure 4.8: An illustrative figure used in the proof of Lemma 4.7.

$$\min_{\epsilon \in \mathbf{E}_i} \left\{ d(\hat{O}, \epsilon) \right\} > \breve{r} \tag{4.25}$$

which contradicts the fact that $\breve{O}$ is the Maxmin-edge centroid. This completes the proof. ■

Lemma 4.8. *If the Maxmin-edge circle is tangent to exactly two edges, then these two edges are parallel. Furthermore, in such a case there will generically be other Maxmin-edge circles, all of which are also tangent to these parallel edges.*

Proof. Suppose a Maxmin-edge circle is tangent to exactly two Voronoi edges, say ϵ_1 and ϵ_2, but these two edges are not parallel. Let the point T be the intersection of ϵ_1 and ϵ_2 (or their extensions). Define:

$$\tilde{w} := \min_{\epsilon \in \mathbf{E}_i - \{\epsilon_1, \epsilon_2\}} \left\{ d(\breve{O}, \epsilon) \right\}, \ i \in \mathbf{n} \tag{4.26}$$

Since $C(\breve{O}, \breve{r})$ is tangent to exactly two edges, the term $\delta^* = (\tilde{w} - \breve{r})/2$ is positive. Let also $\tilde{O}$ be a point on the extension of $T\breve{O}$ (closer to $\breve{O}$) such that $\breve{O}\tilde{O} = \delta$, where δ is an arbitrary value in the interval $(0, \delta^*]$ (as an example, see Fig. 4.9). According to Lemma 4.3:

$$\tilde{w} - \delta \leq d(\breve{O}, \epsilon) - \delta \leq d(\tilde{O}, \epsilon), \ \forall \epsilon \in \mathbf{E}_i - \{\epsilon_1, \epsilon_2\}, \ i \in \mathbf{n} \tag{4.27}$$

It results from (4.27) and the relations $\breve{r} < \breve{r} + \delta \leq \tilde{w} - \delta$ and $d(\tilde{O}, \epsilon_1) = d(\tilde{O}, \epsilon_2) > \breve{r}$ that:

$$\min_{\epsilon \in \mathbf{E}_i} \left\{ d(\tilde{O}, \epsilon) \right\} > \breve{r}, \ i \in \mathbf{n} \tag{4.28}$$

which contradicts the fact that $\breve{O}$ is the Maxmin-edge centroid.

Now, suppose that one Maxmin-edge circle, say C_1, is tangent to two parallel edges, say ϵ_1 and ϵ_2, but there exists another Maxmin-edge circle, say C_2, that it is not tangent to these two edges. Note that the radius of the circle C_1 is equal to $\frac{d(\epsilon_1, \epsilon_2)}{2}$. This implies that the radius of the circle C_2 must be less than $\frac{d(\epsilon_1, \epsilon_2)}{2}$, which contradicts the initial assumption that C_2 is a Maxmin-edge circle. This completes the proof. ■

Remark 4.2. *Similar to the Minmax-edge circle, in the case when all Maxmin-edge circles are tangent to two parallel edges, some of these circles are tangent to three or more edges. In this case, one of such circles is arbitrarily chosen as the Maxmin-edge circle.*

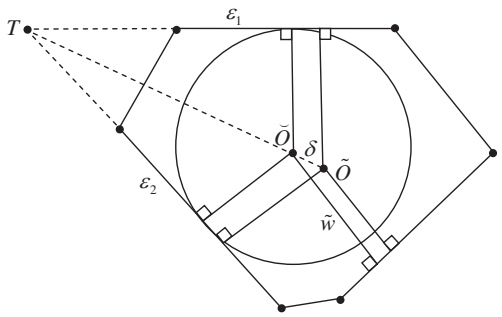


Figure 4.9: An illustrative figure used in the proof of Lemma 4.8.

Theorem 4.4. *Let* $\mathbf{Z}$ *be the set of all circles which: (i) are tangent to at least three edges of a Voronoi polygon, and (ii) are inside the polygon. The Maxmin-edge circle belongs to* $\mathbf{Z}$*, and is the largest circle in this set.*

Proof. According to Lemma 4.8 (and Remark 4.2), the Maxmin-edge circle is tangent to three or more Voronoi edges, and hence it is the incircle or excircle of the triangles created by these edges (possibly extended edges). It is known that $C(\breve{O}, \breve{r}) \in \mathbf{Z}$; thus, it results from Definition 4.8 that $\breve{r} = \max_{C(O,r)\in\mathbf{Z}} \{r\}$. ■

According to Theorem 4.4, the Maxmin-edge centroid is the center of the incircle or excircle of one of the triangles created by three (extended) edges of the polygon. Hence, one can develop an algorithm of complexity $O(e_i^4)$ (which is typically not too high, as noted earlier) to find the Maxmin-edge centroid of a Voronoi polygon.

4.2.4 VEDGE Strategy

As noted earlier, sometimes the vertex-based algorithms are not suitable for coverage improvement, as illustrated in Fig. 7.2. On the other hand, in certain cases the vertex-based algorithms can outperform the edge-based ones in terms of coverage. For example, in Fig. 4.10 the candidate locations for sensor S using the VOR [64], Minimax [64], and Maxmin-edge strategies are the points A, B and D, respectively.

Algorithm 3: Finding the Maxmin-edge centroid of the i-th Voronoi polygon

begin

1) **for** $f = 1, 2, \ldots, e_i - 2$
   for $g = f + 1, f + 2, \ldots, e_i - 1$
   for $h = g + 1, g + 2, \ldots, e_i$
   calculate $\Omega^{f,g,h}$
   if $\Omega^{f,g,h}$ is inside the polygon, **then**
   record it.
   end
   end
   end
   end
2) The center of the largest circle is the Maxmin-edge centroid of the polygon.

It is clear in this case that the VOR and Minimax algorithms increase the coverage area, but the Maxmin-edge algorithm does not. This motivates the development of a new algorithm called VEDGE, as a combination of Minimax (as a vertex-based algorithm) and Maxmin-edge (as an edge-based algorithm). In each round of this algorithm, every sensor selects two points as its candidate locations: one point according to the Minimax strategy and the other one according to the Maxmin-edge strategy. Any of the two points that provides better coverage is selected as the new location of the sensor.

It is worth noting that one can also use a numerical approach such as linear programing or other existing techniques in order to find the centroid point of each region in the second phase of the proposed algorithms [131].

Remark 4.3. *The problem investigated in this chapter is a non-convex optimization problem and all the proposed algorithms are distributed. Thus, if every sensor moves to its optimal location in each iteration, it will not necessarily result in the optimal sensor configuration.*

Remark 4.4. *An important property of the Voronoi diagram is that it partitions the*

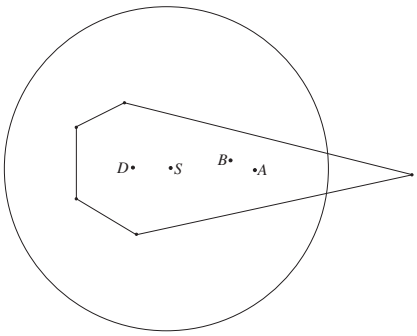
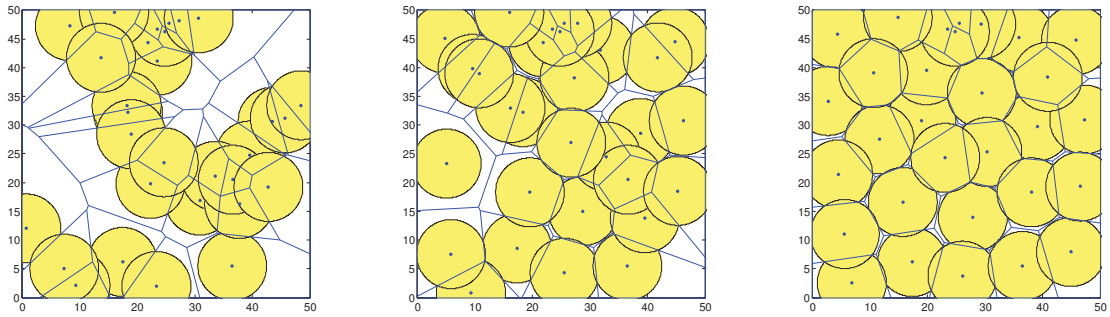


Figure 4.10: An example of a configuration for which the edge-based strategies are not as effective.



(a) (b) (c)

Figure 4.11: Snapshots of the movement of sensors as well as the Voronoi polygons and sensing circles under the VEDGE strategy in Example 1. (a) Initial configuration of sensors; (b) configuration of sensors after the first round, and (c) final configuration.

field in such a way that there is exactly one sensor in each Voronoi polygon. Since under the proposed algorithms the new candidate location of each sensor is inside its current Voronoi polygon, thus the sensor moves within its own Voronoi polygon only to reach the new location. This implies that the sensors will not collide. Assume now that there exists a sensor that cannot communicate with some of its neighbors, and consequently some of the edges of the resultant polygon may be different from the exact Voronoi polygon. As a result, the polygons constructed in this case do not necessarily partition the field in the sense that some of them may overlap with each other. This can have a negative impact on the detection of coverage holes. Furthermore, the overlap of the polygons can lead to sensor collisions.

Remark 4.5. *In order to prevent oscillatory movement of the sensors, a control mechanism similar to the one in [64] is implemented. Under this mechanism, each sensor compares the newly computed direction with the previous one; it will not move in the current round if the new direction is backwards w.r.t. that in the preceding round.*

4.3 Simulation Results

Example 1: In this example, 30 sensors with the sensing range of 6m and the communication range of 20m are randomly deployed in a 50m by 50m flat surface. Fig. 4.11 depicts an operational example of the VEDGE strategy for the above setup. The algorithm is set to terminate when no sensor's coverage in its Voronoi polygon increases by more than 1% in its next move. Three snapshots are provided, and in each one both sensing circles of the sensors (filled circles) and the Voronoi diagram are depicted. After the first round of the algorithm, the coverage increases from the initial value of 60.7% to 81.7%. The algorithm terminates after 13 rounds, and the final coverage is 95.1%. It can be observed from this figure that in the final round the sensors are distributed more evenly than the initial configuration, resulting in significant increase in network coverage.

Remark 4.6. *It is important to note that an analytical solution to the sensor deployment problem for optimal coverage is mathematically too complex to compute. This issue has also been pointed out in the literature, and the performance of any sensor deployment technique is typically evaluated by running a number of simulations with random initial positions for sensors [132], [64], [65], [128], [133], [134]. This approach will be adopted in the next example in order to evaluate the effectiveness of the proposed techniques.*

Example 2: In this example, the proposed algorithms are applied to the same flat

surface and the same type of sensor as the previous example. The results are then compared with the results of the algorithms given in [64]. In these simulations, the algorithms stop when none of the sensors' coverage in its Voronoi polygon would be improved by more than 1% in the next move. It is to be noted that all of the results presented in this example are the average values obtained by performing 100 simulations with random initial positions for sensors. Furthermore, while the horizontal axes of Figs. 4.13-4.16 represent a discrete quantity (number of sensors), the corresponding curves are depicted as continuous graphs for the sake of clarity.

Fig. 4.12 gives the coverage factor (the ratio of the covered area to the total area) for 30 sensors, calculated after each round of different algorithms. It can be observed that all algorithms reach a satisfactory coverage level in the first few rounds. The resultant curves also show that the VEDGE algorithm has the best coverage performance.

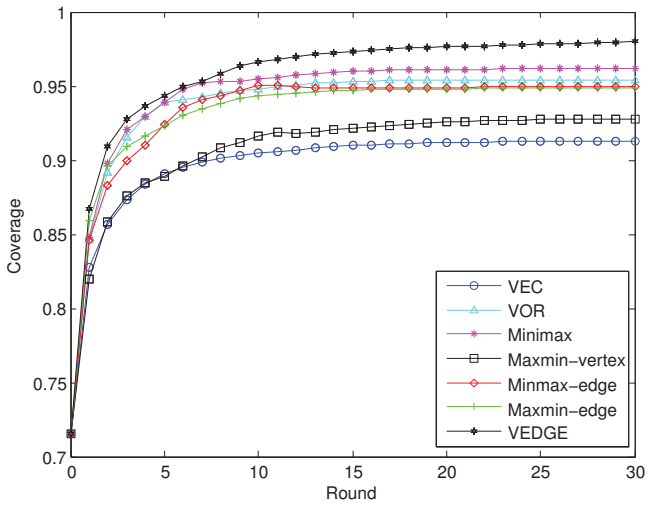


Figure 4.12: The coverage factor for 30 sensors using different strategies.

The time it takes for the network to reach the desired coverage level is another important criterion for measuring the efficiency of the algorithms. Since the deployment time of the sensors in each round is almost the same in all algorithms, the number of rounds required to reach a certain coverage level is used to evaluate

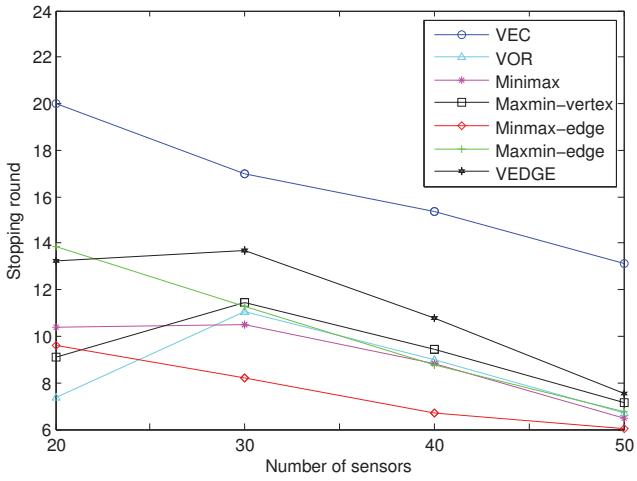


Figure 4.13: The number of rounds required to reach the termination condition for different number of sensors using different strategies.

time efficiency. Fig. 4.13 shows the stopping round of the algorithms for different number of sensors. The simulation is carried out for $n = 20,\ 30,\ 40,\ 50$. It can be seen from this figure that for $n > 30$, the number of rounds decreases as the number of sensors increases. This is due to the fact that when the number of sensors is large, the probability that each sensor covers its Voronoi polygon becomes higher. As a result, the termination condition is satisfied in a shorter period of time in such cases. It can also be observed from Fig. 4.13 that the stopping round for the case of 20 sensors in the VOR strategy is less than that in the other strategies, but for 30 or more sensors the Minmax-edge algorithm converges faster.

Energy-efficiency is another important measure of performance in mobile sensor networks. Energy consumption due to movement is known to be directly related to the moving distance of the sensors, as well as the number of times they stop (note that each time a sensor stops, it will need to overcome the static friction in the next movement). Thus, it is important to also compare the algorithms in terms of the overall moving distance of the sensors, and the number of times they stop. Fig. 4.14 depicts the average moving distance for different number of sensors using different algorithms. These graphs show that the average moving distance is smaller for a

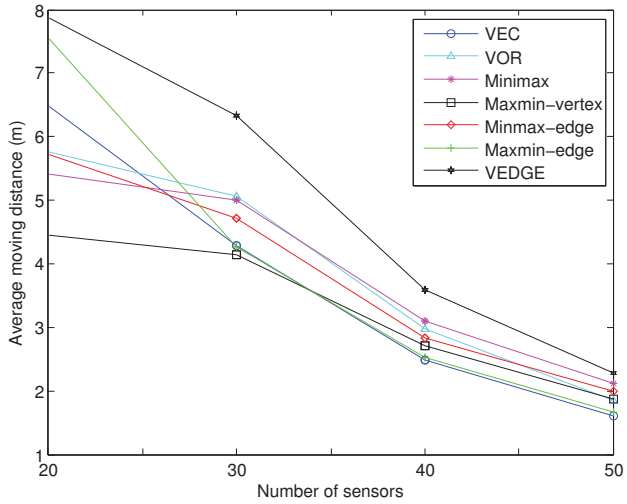


Figure 4.14: The average distance each sensor travels for different number of sensors using different strategies.

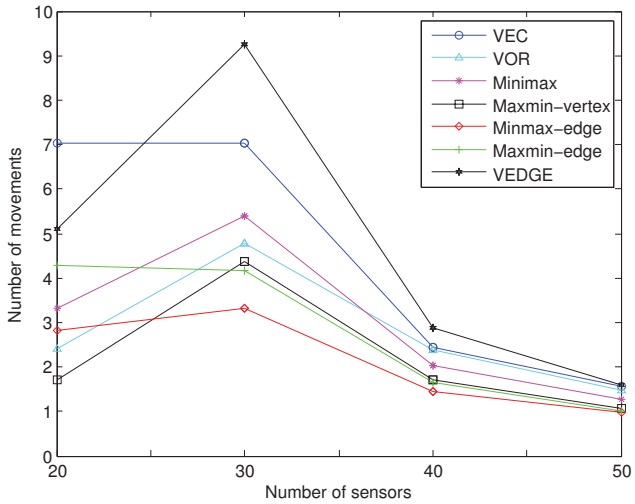


Figure 4.15: The number of movements for different number of sensors using different strategies.

larger number of sensors (for the same reason given earlier). Simulations show that for small number of sensors, the Maxmin-vertex algorithm has the smallest average moving distance. The number of movements versus the number of sensors is given in Fig. 4.15. This figure shows that in most algorithms when the number of sensors increases from 20 to 30, the number of sensor movements also increases. The reason is that when there is a small number of sensors in the network, the Voronoi polygons are relatively large compared to the sensing circles. Thus, it is likely that each Voronoi polygon completely contains the sensing circle of the sensor associated with it. This implies that the sensor's local coverage is maximum (i.e., it is equal to the area of the sensing circle), and hence it will likely not increase if the sensor moves in any direction. However, when the number of sensors increases beyond 30, then the number of movements decreases considerably. In fact, when the number of sensors increases beyond a certain value, it is more likely that each sensor covers its Voronoi polygon. Hence, the termination condition will be satisfied in a shorter period of time, resulting in a decrease in the number of movements. Fig. 4.15 confirms this expectation, and shows that as the number of sensors increases beyond 30, the number of required movements decreases.

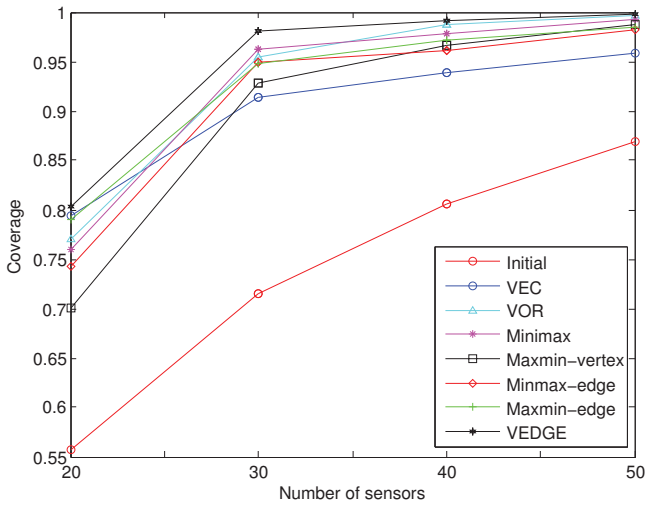


Figure 4.16: The coverage factor for different number of sensors using different strategies.

Table 4.1: The energy consumption in Joule for different number of sensors using different algorithms.

	$n = 20$	$n = 30$	$n = 40$	$n = 50$
VEC	111.5994 J	93.6881 J	40.7438 J	26.2542 J
VOR	67.4912 J	81.3886 J	44.3346 J	27.59789 J
Minimax	72.1187 J	86.0919 J	42.3478 J	27.9395 J
Maxmin-vertx	58.9354 J	77.4206 J	39.5842 J	26.3091 J
Minmax-edge	70.5660 J	66.4283 J	35.2764 J	24.5355 J
Maxmin-edge	97.7894 J	69.7719 J	34.5885 J	22.1933 J
VEDGE	107.3883 J	128.8364 J	53.4931 J	31.8599 J

Assume that the energy required to move a sensor a 1m distance (without stopping in between) be 8.268J [130], [135]. Let the energy required to stop a sensor and then overcome the static friction (in order to move it) be also equal to the above value [65]. Table 4.1 summarizes the results, where it can be observed that when the number of sensors in the network is not large, the Maxmin-vertex strategy outperforms the other techniques in terms of energy consumption. For a large number of sensors, on the other hand, the Maxmin-edge strategy is more energy-efficient compared to the other methods.

In Fig. 4.16, the final coverage of each strategy is depicted for different number of sensors. It can be observed that the VEDGE algorithm has the largest final coverage in all scenarios. It is also interesting to note that although the VEC algorithm does not have a good performance for large number of sensors, it performs relatively well for small number of sensors.

It follows from the above discussion that the choice of an appropriate deployment algorithm involves a trade-off between three main factors: network coverage, deployment time, and energy-efficiency. The discussion is summarized below:

1. The VEDGE algorithm outperforms the other algorithms as far as network coverage is concerned.

2. The Minmax-edge algorithm is more desirable when the deployment time is the main concern AND the number of sensors in the field is not small.

3. The Maxmin-vertex algorithm is more preferable when the energy consumption is the main concern AND the number of sensors in the field is not large.

4. The Maxmin-edge algorithm is more energy-efficient than the other algorithms when there is a large number of sensors in the network.

Chapter 5

Distributed Deployment Algorithms for Efficient Coverage in a Network of Mobile Sensors with Nonidentical Sensing Capabilities

In this chapter, efficient deployment algorithms are proposed for a mobile sensor network to improve the coverage area. The proposed algorithms find the target position of each sensor iteratively, based on the existing coverage holes in the network. The multiplicatively weighted Voronoi (MW-Voronoi) diagram is used to discover the coverage holes corresponding to different sensors with different sensing ranges. Three sensor deployment algorithms are provided: Under the proposed procedures, the sensors move in such a way that the coverage holes in the target field are reduced. Simulations confirm the effectiveness of the deployment algorithms proposed in this chapter.

The plan of the remainder of the chapter is as follows. Some background material along with useful notions and definitions is provided in Section 5.1. Section 5.2 presents the main results of the chapter, where new deployment algorithms are introduced. Finally, in Section 5.3, simulation results are given to demonstrate the effectiveness of the proposed strategies.

5.1 Background

Let each sensor in the network be represented as a node and find the Voronoi polygon for every sensor. These polygons cover the whole field, and the generating node of a polygon is the closest node to any point inside that polygon. Assume that the sensing radii of sensors are all the same. Then a point inside a polygon which is not covered by the sensor in that polygon cannot be covered by other sensors in the network either. This means that in order to identify the coverage holes, it suffices that each sensor checks its own Voronoi polygon to discover the points it cannot cover. However, this fundamental statement is not necessarily true for the case when the sensors have different sensing ranges. When the sensors have different sensing radii, it can be shown that a point inside a polygon may be covered by a sensor in a neighboring polygon, even if it is not covered by the sensor which lies in the same polygon. Hence, in this case the conventional Voronoi diagram is not as useful for effective sensor deployment in the network. The MW-Voronoi diagram described in the next subsection is used to address this issue.

5.1.1 MW-Voronoi Diagram

Consider a set $\mathbf{S}$ containing n distinct weighted nodes $(S_1, w_1), (S_2, w_2), \ldots, (S_n, w_n)$ in a 2D field, where $w_i > 0$ is the weighting factor of the node S_i, $i \in \mathbf{n}$. Define the

weighted distance of a point q from the node (S_i, w_i) as:

$$d_w(q, S_i) = \frac{d(q, S_i)}{w_i}$$

where $d(q, S_i)$ denotes the Euclidean distance between the point q and the node S_i. Divide the field into n regions, where each region contains only one node, which is the closest node, in terms of weighted distance, to any point within that region. The diagram obtained by this partitioning is referred to as the *multiplicatively weighted Voronoi* (MW-Voronoi) diagram [91]. Each region Π_i in this diagram can be characterized as:

$$\Pi_i = \left\{ q \in \mathbb{R}^2 \mid d_w(q, S_i) \leq d_w(q, S_j), \forall j \in \mathbf{n} - \{i\} \right\} \tag{5.1}$$

for any $i \in \mathbf{n}$. It follows from (5.1) that any point Q in Π_i has the following property:

$$\frac{d(q, S_i)}{d(q, S_j)} \leq \frac{w_i}{w_j}, \quad \forall i \in \mathbf{n}, \ \forall j \in \mathbf{n} - \{i\} \tag{5.2}$$

Definition 5.1. *Given two points A, B and a constant k, the* Apollonian circle *$\Omega_{AB,k}$ is the locus of all points C such that $\frac{AC}{BC} = k$ [136].*

To construct the i-th MW-Voronoi region Π_i, the Apollonian circles $\Omega_{S_i S_j, \frac{w_i}{w_j}}$ are first obtained for all $S_j \in \mathbf{S} \backslash \{S_i\}$. Among all regions created by these circles, the smallest one containing S_i is, in fact, Π_i. This process is demonstrated in Fig. 5.1, and an example of an MW-Voronoi diagram with 16 nodes is sketched in Fig. 5.2.

The MW-Voronoi diagram is used to develop sensor deployment strategies in this chapter. Each sensor has a sensing area which is a circle whose size can be different for distinct sensors. Let each sensor in the field be represented by a weighted node in the network whose weight is equal to the sensing radius of that sensor. Draw the MW-Voronoi diagram for the sensors. It is concluded from the mathematical characteristics of the MW-Voronoi diagram given in (5.1) that any point in the i-th MW-Voronoi region which is not covered by sensor S_i cannot be

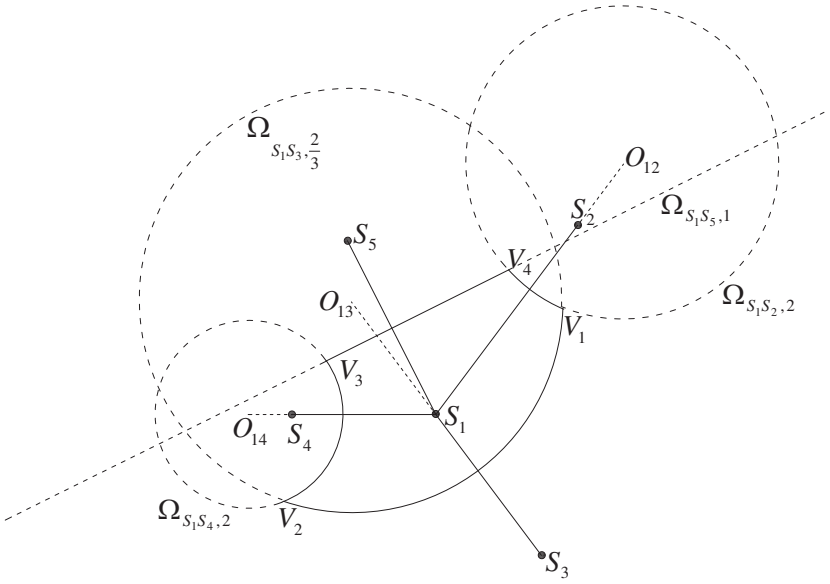


Figure 5.1: The MW-Voronoi region for a node S_1 with four neighboring nodes $S_2, \ldots, S_5$.

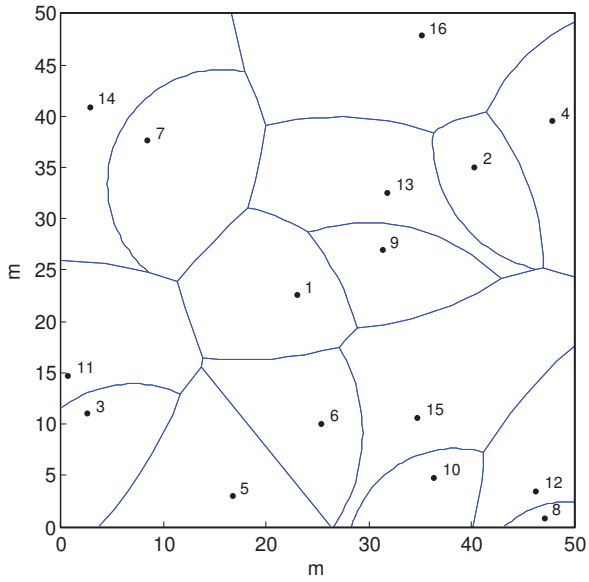


Figure 5.2: An example of an MW-Voronoi diagram for a group of 16 weighted nodes in a 2D plane.

covered by any other sensor in the network. This means that in order to find coverage holes, it suffices to find the points in the MW-Voronoi region of each node, which lie outside its local coverage area.

In the remainder of this chapter, Definitions 4.1, 4.2, 4.3 from Chapter 4 will be used, and it will be assumed that the conditions of Assumptions 4.1, 4.2, 4.3 hold.

5.2 Deployment Protocols

Three distributed deployment protocols are provided in this section for a mobile sensor network. In each step of the procedure every sensor transmits its information (sensing radius and location) to other sensors in the network. The information gathered by each sensor is used to construct its MW-Voronoi region. Each sensor checks its region subsequently to detect the possible coverage holes. If any coverage hole exists, the sensor calculates its target location using a proper deployment strategy to reduce (or eliminate) the holes. Once the new target location $\acute{P}_i$ for sensor i is calculated, the coverage area w.r.t. this location in the current MW-Voronoi region before the sensor movement, i.e., $\beta_{\Pi_i}^{\acute{P}_i}$ is obtained. If this coverage area is greater than the local coverage area before the sensor movement, i.e. $\beta_{\Pi_i}^{\acute{P}_i} > \beta_{\Pi_i}^{P_i}$, the sensor moves to the new location; otherwise, it remains in its current position. In order to terminate the algorithm in finite time, a proper coverage improvement threshold ε is defined such that the algorithm will continue only if there is a sensor in the network whose coverage increases at least by ε in the next iteration. Finally, when none of the sensors' coverage area in its corresponding MW-Voronoi region would be increased by a certain threshold level, there is no need to continue the iterations.

As noted above, one of the important characteristics of the sensor deployment strategies proposed in this chapter is that each sensor moves to its new destination

point only if its coverage area w.r.t. the new location in the old MW-Voronoi region increases. The following result is similar to Theorem 4.1, and shows that the total coverage increases under the proposed algorithms.

Theorem 5.1. *Consider the set* $\mathbf{S}$ *of* n *sensors in the plane, and let their positions and sensing radii be denoted by* $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ *and* $\mathbf{r} = \{r_1, r_2, \ldots, r_n\}$*, respectively, with the corresponding MW-Voronoi regions* $\Pi_1, \Pi_2, \ldots, \Pi_n$*. Assume the sensors move to new positions* $\acute{\mathbf{P}} = \{\acute{P}_1, \acute{P}_2, \ldots, \acute{P}_n\}$ *with the corresponding MW-Voronoi regions* $\acute{\Pi}_1, \acute{\Pi}_2, \ldots, \acute{\Pi}_n$ *such that* $\acute{P}_i \neq P_i$ *for all* $i \in \mathbf{K}$*, where* $\mathbf{K}$ *is a non-empty subset of* $\mathbf{n}$*. If the* i*-th coverage area w.r.t.* $\acute{P}_i$ *in the previously constructed MW-Voronoi region* Π_i *is greater than the* i*-th local coverage area in* Π_i *(i.e.,* $\beta_{\Pi_i}^{\acute{P}_i} > \beta_{\Pi_i}^{P_i}$*) for all* $i \in \mathbf{K}$*, then the total coverage in the network increases.*

Proof. The proof is similar to that of Theorem 4.1, and is omitted here. ■

Remark 5.1. *It is important to note that in Theorem 5.1 the coverage of each sensor w.r.t. its position after moving is obtained in the MW-Voronoi region just before the move, and is then compared to the local coverage of that sensor before the move. In other words, according to Theorem 5.1 in order to verify whether sensor movements would increase total network coverage, it suffices to compare the coverage of every sensor w.r.t. its positions before and after the move, in the old MW-Voronoi region. It is to be noted that an increase in the coverage of a sensor in the old MW-Voronoi region does not necessarily imply that the achieved local coverage in the updated MW-Voronoi region is more than the local coverage of that sensor in the old MW-Voronoi region. Hence, the statement of the theorem is not trivial. This is a very important result and it is guaranteed that under any algorithm which follows the above scheme (including the three algorithms introduced later in this chapter), the total coverage can never decrease.*

Notation 5.1. *Given an MW-Voronoi diagram with* n *regions (each one corresponding to a node), the number of boundary curves and vertices (corners of the boundary,*

associated with the intersections of the boundary curves) of the i-th region ($i \in \mathbf{n}$) are denoted by e_i, and m_i, respectively. It is easy to verify that $m_i = e_i$, for the case when the corresponding region has at least two vertices.

The procedure described in the beginning of this section will be used in the next three subsections to develop the weighted vector based, Minmax-curve, and Maxmin-curve algorithms. It is to be noted that the candidate location for each sensor is determined using the techniques presented in Subsections 5.2.1, 5.2.2 and 5.2.3.

5.2.1 Weighted Vector Based (WVB) Strategy

This method tends to move the sensors out of densely packed areas. Denote by d_{ij} the distance between the sensors S_i and S_j, for any $i, j \in \mathbf{n}$. Define a new (virtual) sensor network with $\bar{n} := \lfloor \sum_{i=1}^{n} w_i^2 \rfloor$ sensors of unit sensing radius, evenly distributed in the sensing area. Let $\bar{d}$ be the distance between a sensor and its nearest neighboring sensor in this new network (this distance can be calculated off-line). In the WVB strategy, if d_{ij} (the distance between the two sensors S_i and S_j in the original network) is less than $\frac{w_i+w_j}{2}\bar{d}$ and none of the two sensors covers its MW-Voronoi region completely, then a virtual force between the two sensors will tend to push them away, as if it wants to move the sensor S_i by $\frac{w_i}{w_i+w_j}D_{ij}$ and the sensor S_j by $\frac{w_j}{w_i+w_j}D_{ij}$, where $D_{ij} = \frac{w_i+w_j}{2}\bar{d} - d_{ij}$. If, however, one of the two sensors, say S_i, covers its region completely, then it will not move, but will push the other sensor S_j by a virtual force as if it wants to move S_j by D_{ij}. In the case when both sensors cover their regions completely, then they will not apply any virtual force to one another. In other words, for every pair of sensors, if there is a coverage hole in any of the corresponding two regions, then a virtual force tends to push the sensors away from each other by $\frac{w_i+w_j}{2}\bar{d}$. On the other hand, virtual forces are also applied in a similar manner from each boundary to any sensor which is closer than a certain

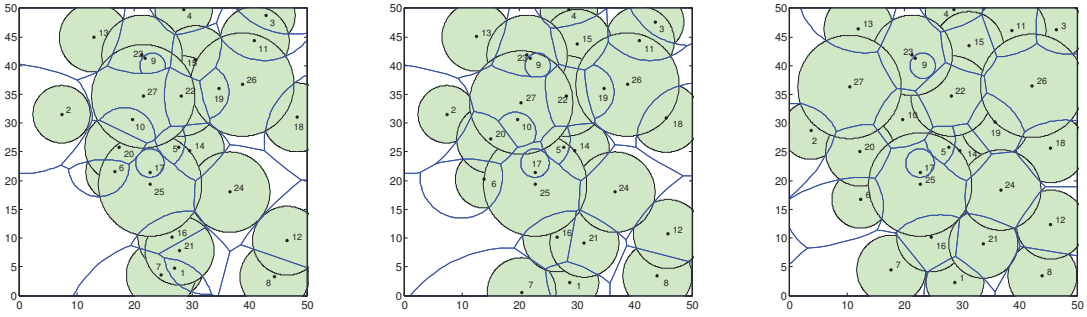
distance to that boundary. More precisely, if the distance d_{bi} between S_i and a certain boundary is less than $\frac{w_i}{2}\bar{d}$, then a virtual force tends to push the sensor away from that boundary by $\frac{w_i}{2}\bar{d} - d_{bi}$. Eventually, each sensor is moved by the vector sum of all virtual forces applied to it from the boundaries and from other sensors. From the above discussion, one can develop an algorithm to find the new candidate location of a sensor in the WVB strategy. For example, see the procedure given in Algorithm 5.4.

Algorithm 4: An algorithm for finding the new candidate location of the i-th sensor in the WVB strategy

begin

$\bar{d}$, d_{ij}, w_i, w_j, D_{ij}: defined before
$\vec{v_i}$: moving vector of S_i
$\vec{f_{ij}}$: the virtual force from S_j to push S_i by $\frac{w_i}{w_i+w_j}D_{ij}$
$\vec{f'_{ij}}$: the virtual force from S_j to push S_i by D_{ij}
$B = \{b_1, \ldots, b_k\}$: the set of the boundary curves of the sensing field
$d_{b_j i}$: the distance between S_i and b_j
$\vec{f_{b_j i}}$: the virtual force from b_j to push S_j by $\frac{w_i}{2}\bar{d} - d_{b_j i}$
C_i: whether the i-th MW-Voronoi region is completely covered
$\vec{v_i} = 0$
1) **for** $j = 1, 2, \ldots, n$
  if $d_{ij} < \frac{w_i+w_j}{2}\bar{d}$ and $C_i \neq true$ and $C_j \neq true$, **then**
    $\vec{v_i} = \vec{v_i} + \vec{f_{ij}}$
  end
  if $d_{ij} < \frac{w_i+w_j}{2}\bar{d}$ and $C_i \neq true$ and $C_j = true$, **then**
    $\vec{v_i} = \vec{v_i} + \vec{f'_{ij}}$
  end
end
2) **for** $j = 1, 2, \ldots, k$
  if $d_{b_j i} < \frac{w_i}{2}\bar{d}$, **then**
    $\vec{v_i} = \vec{v_i} + \vec{f_{b_j i}}$
  end
end
3) S_i moves by $\vec{v_i}$

Fig. 5.3 shows an operational example of the WVB strategy. In this example, 27 sensors are randomly deployed in a 50m $\times$ 50m flat surface: 15 with a sensing



(a) (b) (c)

Figure 5.3: Snapshots of the execution of the movement of the sensors under the WVB strategy; (a) initial coverage; (b) field coverage after the first round, and (c) final coverage.

radius of 6m, 6 with a sensing radius of 5m, 3 with a sensing radius of 7m, and 3 with a sensing radius of 9m. Moreover, the communication range of each sensor is assumed to be 10/3 times its sensing range. In this figure, three snapshots are provided, and in each one the sensing circles of the sensors (filled circles) as well as the MW-Voronoi regions is depicted. The initial coverage is 66.7%, but after the first round it increases to 71.9%, and the final coverage is 85.1%.

Although the vertex-based algorithms prove effective in many cases [101], [64], [99], they may not be as effective for some sensor configurations. For example, consider the MW-Voronoi region in Fig. 5.4, and let the sensor be located at S. It can be easily shown that for increasing the coverage area, the sensor must move up-left. However, the Minmax-vertex, Maxmin-vertex and FPB algorithms [101], [99] tend to move the sensor in the opposite direction (more precisely, to the points A, B and C, respectively). To remedy this shortcoming of the vertex-based algorithms, the Minmax-curve and Maxmin-curve strategies are presented in the sequel.

5.2.2 Minmax-Curve Strategy

The idea behind the Minmax-curve technique is that normally for optimal coverage, any sensor in the network should not be too far from any of its Voronoi curves.

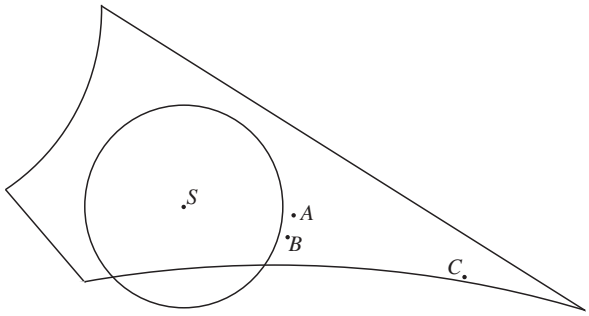


Figure 5.4: An example in which vertex-based algorithms are not as effective.

The Minmax-curve strategy selects the target location for each sensor as a point inside the corresponding MW-Voronoi region which has the smallest distance from the farthest curve. This point will be referred to as the *Minmax-curve centroid*, and will be denoted by $\acute{O}_i$ for the i-th region, $i \in \mathbf{n}$. Furthermore, the distance between this point and the farthest curve from it will be represented by $\acute{r}_i$.

Notation 5.2. *Throughout this chapter, a circle of radius r, centered at O, will be represented by $\Omega(O, r)$.*

Definition 5.2. *The Minmax-curve circle of an MW-Voronoi region is the smallest circle centered inside or on the boundary of that region, intersecting or touching the region's all curves (or their extensions). This circle is, in fact, $\Omega(\acute{O}_i, \acute{r}_i)$, for the i-th region, and is generically unique. In some special configurations, however, there can be infinitely many Minmax-curve circles.*

Some preliminary results will be presented in the sequel, which will be used in the Minmax-curve and Maxmin-curve strategies.

Fact 5.1. *Consider two points A and B in a 2D plane, and let the distance between them be d. It is well-known that:*

a) The locus of any point E such that $d(E, A) - d(E, B) = k$ is:

i) The perpendicular bisector of segment AB, for $k = 0$.

ii) A branch of a hyperbola, for $0 < k < d$.

iii) The extension of the segment AB from B, for $k = d$.

iv) The empty set, for $k > d$.

b) The locus of any point E such that $d(E, A) + d(E, B) = k$ *is:*

i) An empty set, for $k < d$.

ii) The segment AB, for $k = d$.

iii) An ellipse, for $k > d$.

Notation 5.3. *The set of all boundary curves* ϵ_{i1}, ϵ_{i2}, ..., ϵ_{ie_i} *of the i-th MW-Voronoi region will hereafter be denoted by the boldfaced symbol* $\boldsymbol{\epsilon}_i$. *In the present subsection, intersecting/touching/tangent to a boundary curve* ϵ_{ij} *means intersecting/touching/tangent to* ϵ_{ij} *or its extension (*$i \in \mathbf{n}$, $j \in \{1, \ldots, e_i\}$*). Note that the extension of the boundary curve* ϵ_{ij} *belongs to the same Apollonian circle as* ϵ_{ij}.

Definition 5.3. *The bisector of two curves* ϵ_{i1} *and* ϵ_{i2} *is defined as the locus of all points whose distance from* ϵ_{i1} *is equal to that from* ϵ_{i2}. *The bisector of the curves* ϵ_{i1} *and* ϵ_{i2} *is denoted by* $\Gamma_{\epsilon_{i1},\epsilon_{i2}}$.

Lemma 5.1. *Consider two circles* $\Omega_1(O_1, r_1)$ *and* $\Omega_2(O_2, r_2)$. *The bisector of* Ω_1 *and* Ω_2 *is:*

i) A branch of a hyperbola or the perpendicular bisector of O_1O_2, *if* Ω_2 *is outside* Ω_1.

ii) An ellipse, if Ω_2 *is inside* Ω_1.

iii) The union of a branch of a hyperbola or the perpendicular bisector of O_1O_2 *and an ellipse, if* Ω_1 *intersects* Ω_2.

Proof.

i) Consider two circles $\Omega_1(O_1, r_1)$ and $\Omega_2(O_2, r_2)$, where Ω_2 is outside Ω_1. Let E be a point on the plane, such that $d(E, \Omega_1) = d(E, \Omega_2) = \delta$, where δ is a given

strictly positive constant (see Fig. 5.5). Since $EO_1 = \delta + r_1$ and $EO_2 = \delta + r_2$, it is concluded that:

$$EO_1 - EO_2 = r_1 - r_2 \tag{5.3}$$

The proof of this part follows now from Fact 5.1 (note that the right side of (5.3) is constant).

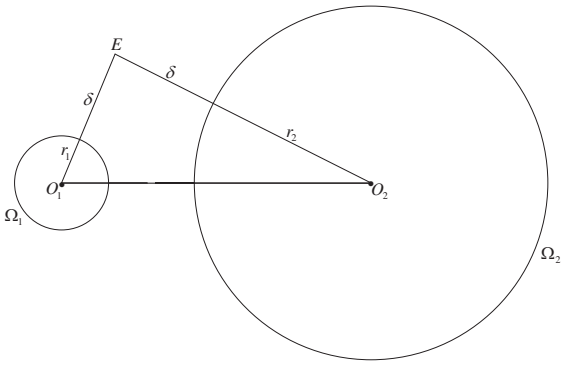


Figure 5.5: Figure for the proof of Lemma 5.1, part (i).

ii) This part can be proved analogously to part (i), on noting that if $\Omega_2(O_2, r_2)$ is inside $\Omega_1(O_1, r_1)$, then for any point E satisfying the relation $d(E, \Omega_1) = d(E, \Omega_2)$ (see Fig. 5.6):

$$EO_1 + EO_2 = r_1 + r_2 \tag{5.4}$$

(note that the right side of the above equation is constant).

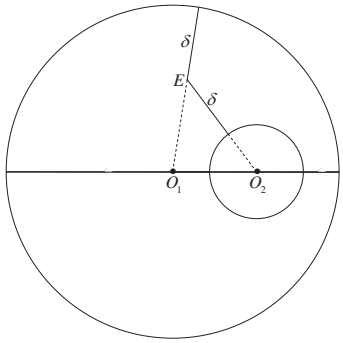


Figure 5.6: Figure for the proof of Lemma 5.1, part (ii).

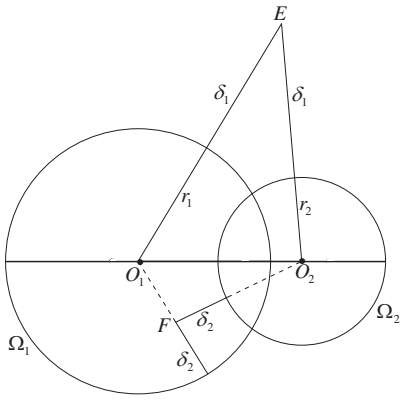


Figure 5.7: Figure for the proof of Lemma 5.1, part (iii).

iii) Consider two intersecting circles $\Omega_1(O_1, r_1)$ and $\Omega_2(O_2, r_2)$, and let E be a point either inside, or outside both circles. Let also F be a point inside Ω_1 and outside Ω_2 such that $d(E, \Omega_1) = d(E, \Omega_2) = \delta_1$ and $d(F, \Omega_1) = d(F, \Omega_2) = \delta_2$, where δ_1 and δ_2 are strictly positive constants (see Fig. 5.7). Since $EO_1 = \delta_1 + r_1$, $EO_2 = \delta_1 + r_2$, $FO_1 = r_1 - \delta_2$ and $FO_2 = r_2 + \delta_2$, hence:

$$EO_1 - EO_2 = r_1 - r_2 \tag{5.5}$$

$$FO_1 + FO_2 = r_1 + r_2 \tag{5.6}$$

The proof of this part follows now from Fact 5.1. ■

Lemma 5.2. *Consider a circle $\Omega(O, r)$ and a line Δ. The bisector of Ω and Δ is:*
i) A parabola, if Δ does not intersect Ω.
ii) The union of two parabolas, if Δ intersects Ω.

Proof. The proof follows immediately from basic algebraic manipulations and Fact 5.1 (see Figs. 5.8 and 5.9). ■

Lemma 5.3. *Consider two points A, B, and a circle $\Omega(O, r)$ (which in the particular case can be a straight line). Let the distance between A and $\Omega(O, r)$ be denoted by σ, and that between B and this circle by ρ. Let also the distance between A and B*

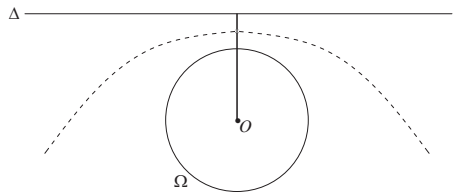


Figure 5.8: Figure for the proof of Lemma 5.2, part (i).

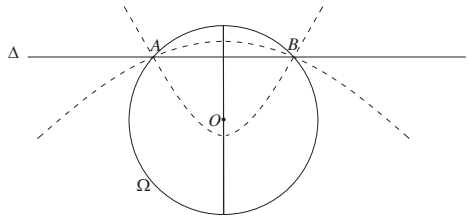


Figure 5.9: Figure for the proof of Lemma 5.2, part (ii).

be denoted by ξ*. Then:*

$$\sigma - \xi \leq \rho \leq \sigma + \xi \tag{5.7}$$

Proof. Let E and F be two points on $\Omega(O, r)$, such that $AE \perp \Omega$ and $BF \perp \Omega$. Let $d(A, F) = \delta_1$ and $d(B, E) = \delta_2$, where δ_1, δ_2 are strictly positive constants (see Fig. 5.10). Then, according to the triangle inequality, $OA + AB \geq OB$ and $OB + AB \geq OA$. Now, since $OA = r + \sigma$ and $OB = r + \rho$, one can conclude that:

$$\sigma + \xi \geq \rho \tag{5.8}$$

$$\rho + \xi \geq \sigma \tag{5.9}$$

The relation (5.7) follows directly from (5.8), (5.9), and this completes the proof. ■

Lemma 5.4. *If an MW-Voronoi region has more than one boundary curve, then the corresponding Minmax-curve circle is tangent to at least two of the boundary curves.*

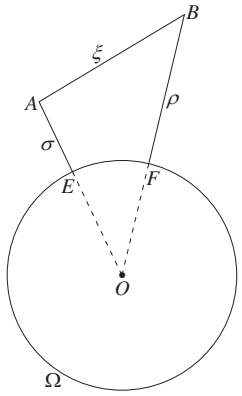


Figure 5.10: Figure for the proof of Lemma 5.3.

Proof. Suppose the i-th region of an MW-Voronoi diagram has more than one boundary curve. Let $\acute{\epsilon}_{i1}$ be the farthest boundary curve from the Minmax-curve centroid of this region. By definition, $\acute{r}_i$ is equal to $d(\acute{O}_i, \acute{\epsilon}_{i1})$, i.e., the distance between $\acute{O}_i$ and $\acute{\epsilon}_{i1}$. Thus, $\Omega(\acute{O}_i, \acute{r}_i)$ is tangent to $\acute{\epsilon}_{i1}$ (or its extension). Define:

$$\hat{v} := \max_{\epsilon_{ij} \in \epsilon_i \backslash \{\acute{\epsilon}_{i1}\}} \left\{ d(\acute{O}_i, \epsilon_{ij}) \right\} \tag{5.10}$$

Suppose that the Minmax-curve circle is not tangent to any other edge, and hence $\delta^* = (\acute{r}_i - \hat{v})/2$ is strictly positive. Let M be a point on $\acute{\epsilon}_{i1}$ or its extension, such that $M\acute{O}_i \perp \acute{\epsilon}_{i1}$. Let also $\hat{O}$ be a point on $M\acute{O}$, such that the distance between $\acute{O}_i$ and $\hat{O}$ is equal to an arbitrary value $\delta \in (0, \delta^*]$ (see, e.g. Fig. 5.11). According to Lemma 5.3:

$$d(\hat{O}, \epsilon_{ij}) \leq d(\acute{O}_i, \epsilon_{ij}) + \delta \leq \hat{v} + \delta, \ \forall \epsilon_{ij} \in \epsilon_i \backslash \{\acute{\epsilon}_{i1}\} \tag{5.11}$$

From the relation (5.11) as well as the inequalities $\hat{v} + \delta < \acute{r}_i - \delta < \acute{r}_i$ and $d(\hat{O}, \acute{\epsilon}_{i1}) < d(\acute{O}_i, \acute{\epsilon}_{i1})$, one arrives at the following inequality:

$$\max_{\epsilon_{ij} \in \epsilon_i} \left\{ d(\hat{O}, \epsilon_{ij}) \right\} < \acute{r}_i \tag{5.12}$$

which contradicts the initial assumption that $\acute{O}_i$ is the Minmax-curve centroid. ■

Remark 5.2. *If an MW-Voronoi region has exactly one boundary curve, then this curve is a circle and it is, in fact, the Minmax-curve circle. If it has exactly two*

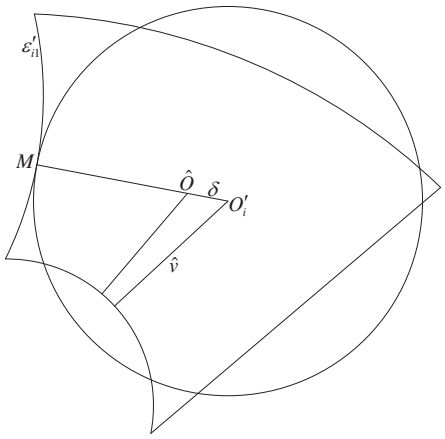


Figure 5.11: Figure for the proof of Lemma 5.4.

boundary curves, then according to Lemma 5.4 the Minmax-curve circle is tangent to both curves.

Definition 5.4. *For any two curves ϵ_{i1} and ϵ_{i2}, the sets $\Psi^{max}_{\epsilon_{i1},\epsilon_{i2}}$ and $\Psi^{min}_{\epsilon_{i1},\epsilon_{i2}}$ are defined as follows:*

$$\Psi^{min}_{\epsilon_{i1},\epsilon_{i2}} = \{X \in \Gamma_{\epsilon_{i1},\epsilon_{i2}} | \exists \delta > 0 : \forall Y \in \Gamma_{\epsilon_{i1},\epsilon_{i2}}, |Y - X| \leq \delta \Rightarrow d(X, \epsilon_{i1}) \leq d(Y, \epsilon_{i1})\} \tag{5.13}$$

$$\Psi^{max}_{\epsilon_{i1},\epsilon_{i2}} = \{X \in \Gamma_{\epsilon_{i1},\epsilon_{i2}} | \exists \delta > 0 : \forall Y \in \Gamma_{\epsilon_{i1},\epsilon_{i2}}, |Y - X| \leq \delta \Rightarrow d(X, \epsilon_{i1}) \geq d(Y, \epsilon_{i1})\} \tag{5.14}$$

Definition 5.5. *Let ϵ_{i1} and ϵ_{i2} be two arbitrary circular arcs of circles Ω_1 and Ω_2 respectively. The curves ϵ_{i1} and ϵ_{i2} are called parallel if the circles Ω_1 and Ω_2 are concentric.*

Lemma 5.5. *Consider an MW-Voronoi diagram, and assume that the i-th region has at least three boundary curves (and hence, according to Lemma 5.4 the Minmax-curve circle of this region is tangent to two or more boundary curves). If the Minmax-curve circle is tangent to exactly two boundary curves, say $\acute{\epsilon}_{i1}$ and $\acute{\epsilon}_{i2}$, then at least*

one of the following conditions holds:

i) $\acute{\epsilon}_{i1}$ *and* $\acute{\epsilon}_{i2}$ *are parallel;*

ii) $\acute{O}_i \in \Psi^{min}_{\acute{\epsilon}_{i1},\acute{\epsilon}_{i2}}$*, or*

iii) $\acute{O}_i$ *is the intersection of the bisector of* $\acute{\epsilon}_{i1}$*,* $\acute{\epsilon}_{i2}$*, and one boundary curve of the region.*

Proof. To prove by contradiction, assume that the Minmax-curve circle is tangent to exactly two non-parallel Voronoi curves $\acute{\epsilon}_{i1}$ and $\acute{\epsilon}_{i2}$, but the Minmax-curve centroid $\acute{O}_i$ is not the intersection of the bisector of $\acute{\epsilon}_{i1}$, $\acute{\epsilon}_{i2}$, and a boundary curve of the region (i.e., $\acute{O}_i$ is inside the region). Define:

$$\tilde{v} := \max_{\epsilon_{ij} \in \epsilon_i \backslash \{\acute{\epsilon}_{i1}, \acute{\epsilon}_{i2}\}} \Big\{ d(\acute{O}_i, \epsilon_{ij}) \Big\} \tag{5.15}$$

Define also $\delta^* = (\acute{r}_i - \tilde{v})/2$. Since $\Omega(\acute{O}_i, \acute{r}_i)$ is tangent to exactly two boundary curves, δ^* is strictly positive. If $\acute{O}_i$ is not in $\Psi^{min}_{\acute{\epsilon}_{i1},\acute{\epsilon}_{i2}}$, then one can choose a point inside the Minmax-circle and on the bisector of $\acute{\epsilon}_{i1}$ and $\acute{\epsilon}_{i2}$, say $\tilde{O}$, such that $d(\tilde{O}, \acute{\epsilon}_{i1}) = d(\tilde{O}, \acute{\epsilon}_{i2}) < \acute{r}_i$, and the distance between $\acute{O}_i$ and $\tilde{O}$ is equal to an arbitrary value $\delta \in (0, \delta^*]$ (see Fig. 5.12). According to Lemma 5.3:

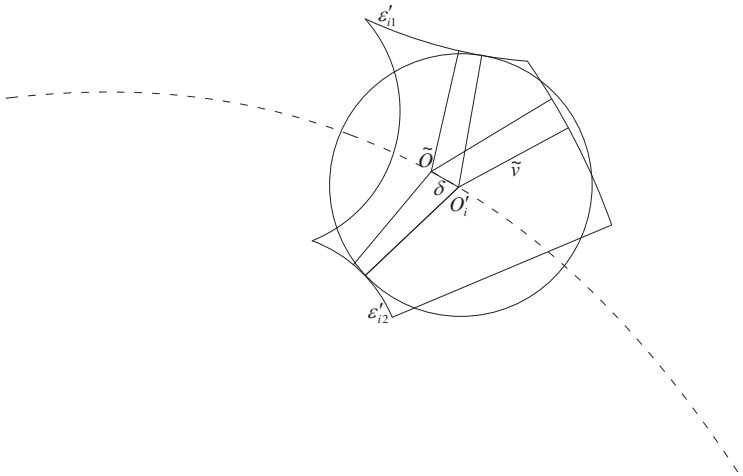


Figure 5.12: Figure for the proof of Lemma 5.5.

$$d(\tilde{O}, \epsilon_{ij}) \leq d(\acute{O}_i, \epsilon_{ij}) + \delta \leq \tilde{v} + \delta, \ \forall \epsilon_{ij} \in \epsilon_i \backslash \{\acute{\epsilon}_{i1}, \acute{\epsilon}_{i2}\} \tag{5.16}$$

It results from (5.16) and the relations $\tilde{v}+\delta \leq \acute{r}_i - \delta < \acute{r}_i$ and $d(\tilde{O}, \acute{\epsilon}_{i1}) = d(\tilde{O}, \acute{\epsilon}_{i2}) < \acute{r}_i$, that:

$$\max_{\epsilon_{ij} \in \epsilon_i} \left\{ d(\tilde{O}, \epsilon_{ij}) \right\} < \acute{r}_i \tag{5.17}$$

which contradicts the initial assumption that $\acute{O}_i$ is the Minmax-curve centroid. This completes the proof. ■

As noted earlier, the Minmax-curve circle is generically unique, and only in some special configurations there can be more than one such circle. The next lemma addresses the case where there are more than one Minmax-curve circle.

Lemma 5.6. *Assume a Minmax-curve circle is tangent to two parallel curves. If there are any other Minmax-curve circles, then all of them are tangent to these two parallel curves.*

Proof. Suppose one Minmax-curve circle, say Ω_1, is tangent to two parallel curves, say ϵ_{i1} and ϵ_{i2}, but there exists another Minmax-curve circle, say Ω_2, that is not tangent to ϵ_{i1} or ϵ_{i2}. Let the distance between ϵ_{i1} and ϵ_{i2} be denoted by $d(\epsilon_{i1}, \epsilon_{i2})$. From the definition of the Minmax-curve circle, the radius of Ω_1 is equal to $\frac{d(\epsilon_{i1}, \epsilon_{i2})}{2}$, and that of Ω_2 is grater than $\frac{d(\epsilon_{i1}, \epsilon_{i2})}{2}$. This contradicts the initial assumption that Ω_2 is a Minmax-curve circle, and completes the proof. ■

Remark 5.3. *Consider an MW-Voronoi region with at least three boundary curves, and assume two of them are parallel. If one of the Minmax-curve circles is tangent to these parallel curves, then all Minmax-curve circles are also tangent to these two curves. At least one of these circles is tangent to some other boundary curves too, and one of such circles is arbitrarily chosen as the Minmax-curve circle in this case.*

Definition 5.6. *For convenience of notation, the circle touching two curves ϵ_{ig} and ϵ_{ih} of the i-th MW-Voronoi region, centered at the intersection of the curve ϵ_{ik} and the bisector of ϵ_{ig} and ϵ_{ih}, is denoted by $\Omega^k_{g,h}$, for any $k, g, h \in \mathbf{e}_i := \{1, \ldots, e_i\}$.*

Also, the circle touching the two curves ϵ_{ir} and ϵ_{is} of the i-th MW-Voronoi region, centered at the point $A \in \Psi^{min}_{\epsilon_{ir},\epsilon_{is}}$, is denoted by $\Omega^{A,min}_{r,s}$, for any $r, s \in \mathbf{e}_i$. In addition, the circle touching the three boundary curves ϵ_{ip}, ϵ_{iq} and ϵ_{it} of region i is denoted by $\Omega_{p,q,t}$, $p, q, t \in \mathbf{e}_i$.

Theorem 5.2. *Consider an MW-Voronoi diagram, and suppose the i-th region has at least three boundary curves. Let $\hat{\mathbf{D}}_i$ and $\grave{\mathbf{D}}_i$ be, respectively, the sets of all circles $\Omega^k_{g,h}, \forall k, g, h \in \mathbf{e}_i$, and $\Omega^{A,min}_{r,s}, \forall r, s \in \mathbf{e}_i, A \in \Psi^{min}_{\epsilon_{ir},\epsilon_{is}}$ such that: (i) their centers lie inside the region or on its boundary, and (ii) they intersect or are tangent to all of the boundary curves of the region (or their extensions, as noted before). Let also $\tilde{\mathbf{D}}_i$ be the set of all circles such that: (i) they are tangent to at least three boundary curves of the i-th region; (ii) their centers lie inside the region or on its boundary, and (iii) they intersect or are tangent to all of the boundary curves of the MW-Voronoi region. Define $\mathbf{D}_i := \hat{\mathbf{D}}_i \cup \grave{\mathbf{D}}_i \cup \tilde{\mathbf{D}}_i$; then the Minmax-curve circle belongs to $\mathbf{D}_i$, and is the smallest circle in this set.*

Proof. The proof follows directly from Lemmas 5.5 and 5.6, Remark 5.3, and Definitions 5.2 and 5.6. ■

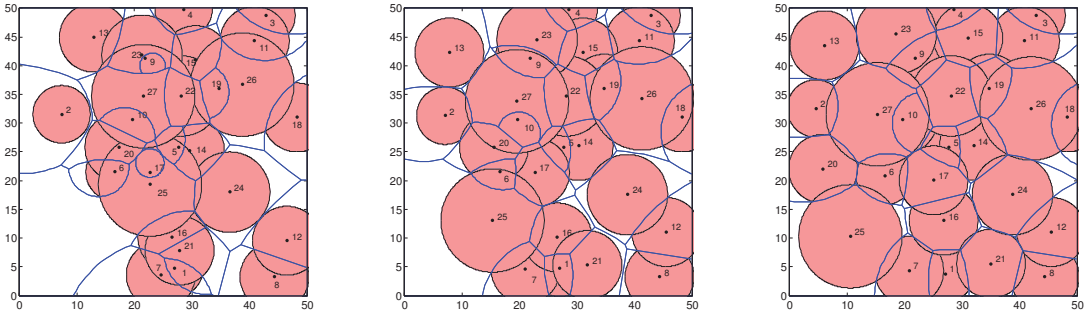
The result of Theorem 5.2 is used to develop an algorithm of complexity $O(e_i^4)$ for finding the Minmax-curve centroid of an MW-Voronoi region. Detailed steps are provided in Algorithm 5.5, and since typically an MW-Voronoi region does not have a "large" number of boundary curves, the computational complexity of the algorithm is normally not very high.

As an example, consider a sensor network with the same initial configuration as in Fig. 5.3(a), and let the Minmax-curve strategy be used. After the first round, the coverage is improved to 78.5%, and finally it reaches 90.8%, as depicted in Fig. 5.13.

Algorithm 5: An algorithm for finding the Minmax-curve centroid of the i-th MW-Voronoi region

begin

1) **for** $p = 1, 2, \ldots, e_i - 2$
   for $q = p + 1, p + 2, \ldots, e_i - 1$
   for $t = q + 1, q + 2, \ldots, e_i$
   calculate $\Omega_{p,q,t}$
   if $\Omega_{p,q,t}$ is centered inside the i-th MW-Voronoi region or on its boundary and intersects or is tangent to all of the boundary curves of the region, **then**
   record it.
   end
   end
   end
   end
2) **for** $r = 1, 2, \ldots, e_i - 1$
   for $s = r + 1, r + 2, \ldots, e_i$
   calculate $\Omega_{r,s}^{A,min}$
   if $\Omega_{r,s}^{A,min}$ is centered inside the i-th MW-Voronoi region or on its boundary and intersects or is tangent to all of the boundary curves of the region, **then**
   record it.
   end
   end
   end
3) **for** $g = 1, 2, \ldots, e_i - 1$
   for $h = g + 1, g + 2, \ldots, e_i$
   calculate $\Omega_{g,h}^{k}$
   if $\Omega_{g,h}^{k}$ intersects or is tangent to all of the boundary curves of the i-th region, **then**
   record it.
   end
   end
   end
4) The center of the smallest circle is the Minmax-curve centroid of the i-th MW-Voronoi region.



(a) (b) (c)

Figure 5.13: Snapshots of the execution of the Minmax-curve strategy. (a) Initial coverage; (b) coverage after the first round, and (c) final coverage.

5.2.3 Maxmin-Curve Strategy

The main idea behind the Maxmin-curve strategy is that normally for optimal coverage, no sensor should be too close to any of its Voronoi curves. The candidate location of a sensor under the Maxmin-curve strategy is a point inside the corresponding MW-Voronoi region which has the largest distance from the nearest curve. This point will be referred to as the *Maxmin-curve centroid*, and will be denoted by $\breve{O}_i$ for the i-th region, $i \in \mathbf{n}$. Furthermore, the distance between this point and the nearest curve to it will be represented by $\breve{r}_i$. The Maxmin-curve circle is defined next.

Definition 5.7. *The Maxmin-curve circle of an MW-Voronoi region is the largest circle that fits inside the region. This circle is, in fact,* $\Omega(\breve{O}_i, \breve{r}_i)$*, for the* i*-th region. Similar to the Minmax-curve circle, the Maxmin-curve circle is also generically unique, but in some special cases, as shown later, there can be infinitely many such circles.*

Lemma 5.7. *If an MW-Voronoi region has more than one boundary curve, then the corresponding Maxmin-curve circle is tangent to at least two of the curves.*

Proof. Let $\breve{\epsilon}_{i1}$ be the nearest boundary curve to the Maxmin-curve centroid of the i-th MW-Voronoi region. This means that $\breve{r}_i$ is equal to the distance between

$\breve{O}_i$ and $\breve{\epsilon}_{i1}$, which is denoted by $d(\breve{O}_i, \breve{\epsilon}_{i1})$; thus, $\Omega(\breve{O}_i, \breve{r}_i)$ is tangent to $\breve{\epsilon}_{i1}$. Define:

$$\hat{w} = \min_{\epsilon_{ij} \in \epsilon_i \backslash \{\breve{\epsilon}_{i1}\}} \left\{ d(\breve{O}_i, \epsilon_{ij}) \right\} \tag{5.18}$$

Define also $\delta^* = (\hat{w} - \breve{r}_i)/2$. Suppose that the Maxmin-curve circle is not tangent to any other boundary curve, and hence δ^* is strictly positive. Let M be a point on $\breve{\epsilon}_{i1}$ such that $M\breve{O}_i \perp \breve{\epsilon}_{i1}$. Let also $\hat{O}$ be a point on the extension of $M\breve{O}_i$ such that the distance between $\breve{O}_i$ and $\hat{O}$ is equal to an arbitrary value $\delta \in (0, \delta^*]$ (see Fig. 5.14). According to Lemma 5.3:

$$d(\hat{O}, \epsilon_{ij}) \geq d(\breve{O}_i, \epsilon_{ij}) - \delta \geq \hat{w} - \delta, \ \forall \epsilon_{ij} \in \epsilon_i \backslash \{\breve{\epsilon}_{i1}\} \tag{5.19}$$

From (5.19) and the relations $\hat{w} - \delta \geq \breve{r}_i + \delta > \breve{r}_i$ and $d(\hat{O}, \breve{\epsilon}_{i1}) > d(\breve{O}_i, \breve{\epsilon}_{i1})$, one can

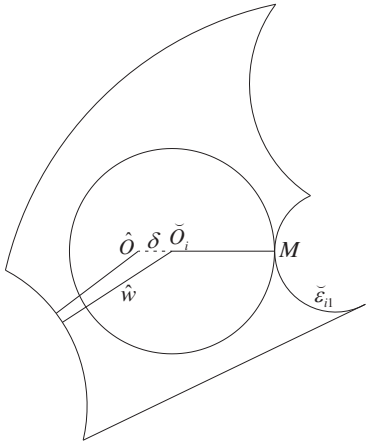


Figure 5.14: Figure for the proof of Lemma 5.7.

conclude that:

$$\min_{\epsilon_{ij} \in \epsilon_i} \left\{ d(\hat{O}, \epsilon_{ij}) \right\} > \breve{r}_i \tag{5.20}$$

which contradicts the fact that $\breve{O}_i$ is a Maxmin-curve centroid. This completes the proof. ■

Lemma 5.8. *Consider an MW-Voronoi diagram, and suppose that the i-th region has at least three boundary curves. If a Maxmin-curve circle is tangent to exactly two*

boundary curves, say $\breve{\epsilon}_{i1}, \breve{\epsilon}_{i2}$*, then these two curves are either parallel or* $\breve{O}_i \in \Psi^{max}_{\breve{\epsilon}_{i1},\breve{\epsilon}_{i2}}$.

Proof. Suppose $\breve{\epsilon}_{i1}$ and $\breve{\epsilon}_{i2}$ are not parallel. Define:

$$\tilde{w} := \min_{\epsilon_{ij} \in \epsilon_i \backslash \{\breve{\epsilon}_{i1}, \breve{\epsilon}_{i2}\}} \Big\{ d(\breve{O}_i, \epsilon_{ij}) \Big\} \tag{5.21}$$

Define also $\delta^* = (\tilde{w} - \breve{r})/2$. Since $\Omega(\breve{O}_i, \breve{r}_i)$ is tangent to exactly two boundary curves, δ^* is strictly positive. If $\breve{O}_i \notin \Psi^{max}_{\breve{\epsilon}_{i1},\breve{\epsilon}_{i2}}$, then one can choose a point $\tilde{O}$ inside the i-th region, on the bisector of $\breve{\epsilon}_{i1}$ and $\breve{\epsilon}_{i2}$, such that $d(\tilde{O}, \breve{\epsilon}_{i1}) = d(\tilde{O}, \breve{\epsilon}_{i2}) > \breve{r}_i$, and $\breve{O}_i\tilde{O} = \delta$, for some $\delta \in (0, \delta^*]$ (see Fig. 5.15). According to Lemma 5.3:

$$d(\tilde{O}, \epsilon_{ij}) \geq d(\breve{O}_i, \epsilon_{ij}) - \delta \geq \tilde{w} - \delta, \ \forall \epsilon_{ij} \in \epsilon_i \backslash \{\breve{\epsilon}_{i1}, \breve{\epsilon}_{i2}\} \tag{5.22}$$

It results from (5.22) and the relations $\tilde{w} - \delta \geq \breve{r}_i + \delta > \breve{r}_i$ and $d(\tilde{O}, \breve{\epsilon}_{i1}) = d(\tilde{O}, \breve{\epsilon}_{i2}) >$

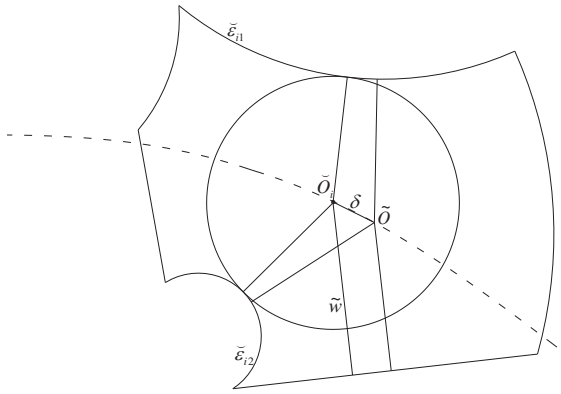


Figure 5.15: Figure for the proof of Lemma 5.8.

$\breve{r}_i$, that:

$$\min_{\epsilon_{ij} \in \epsilon_i} \Big\{ d(\tilde{O}, \epsilon_{ij}) \Big\} > \breve{r}_i \tag{5.23}$$

which contradicts the fact that $\breve{O}_i$ is a Maxmin-curve centroid. This completes the proof. ■

Definition 5.8. *For convenience of notation, the circle tangent to two curves* ϵ_{ir} *and* ϵ_{is} *of the i-th MW-Voronoi region, centered at the point* $A \in \Psi^{max}_{\epsilon_{ir},\epsilon_{is}}$*, is denoted by* $\Omega^{A,max}_{r,s}$*, for any* $r, s \in \mathbf{e}_i$.

Lemma 5.9. *Assume a Maxmin-curve circle is tangent to two parallel curves. If there are any other Maxmin-curve circles, then all of them are tangent to these two parallel curves.*

Proof. Suppose one Maxmin-curve circle, say Ω_1, is tangent to two parallel edges, say ϵ_{i1} and ϵ_{i2}, but there exists another Maxmin-curve circle, say Ω_2, that is not tangent to ϵ_{i1} or ϵ_{i2}. Note that the radius of the circle Ω_1 is equal to $\frac{d(\epsilon_{i1},\epsilon_{i2})}{2}$, and that of the circle Ω_2 is less than $\frac{d(\epsilon_{i1},\epsilon_{i2})}{2}$. This contradicts the initial assumption that Ω_2 is a Maxmin-curve circle. ■

Remark 5.4. *Consider an MW-Voronoi region with at least three boundary curves, two of which are parallel. If one of the Maxmin-curve circles is tangent to these parallel curves, then all Maxmin-curve circles are also tangent to these two curves. At least one of these circles is tangent to some other boundary curves too, and one of such circles is arbitrarily chosen as the Maxmin-curve circle in this case.*

Remark 5.5. *If an MW-Voronoi region has exactly one boundary curve, then this curve is a circle as pointed out before, and it is, in fact, the Maxmin-curve circle.*

Theorem 5.3. *Consider an MW-Voronoi diagram, and suppose that the i-th MW-Voronoi region has at least three boundary curves. Let* $\grave{\mathbf{Z}}_i$ *be the set of all circles* $\Omega_{r,s}^{A,max}, \forall r, s \in \mathbf{e}_i,\ A \in \Psi_{\epsilon_{ir},\epsilon_{is}}^{max}$, *inside the region. Let also* $\tilde{\mathbf{Z}}_i$ *be the set of all circles which: (i) are tangent to at least three boundary curves of the i-th region, and (ii) are inside the region. Define* $\mathbf{Z_i} := \grave{\mathbf{Z}}_\mathbf{i} \cup \tilde{\mathbf{Z}}_\mathbf{i}$*; then the Maxmin-curve circle belongs to* $\mathbf{Z}_i$*, and is the largest circle in this set.*

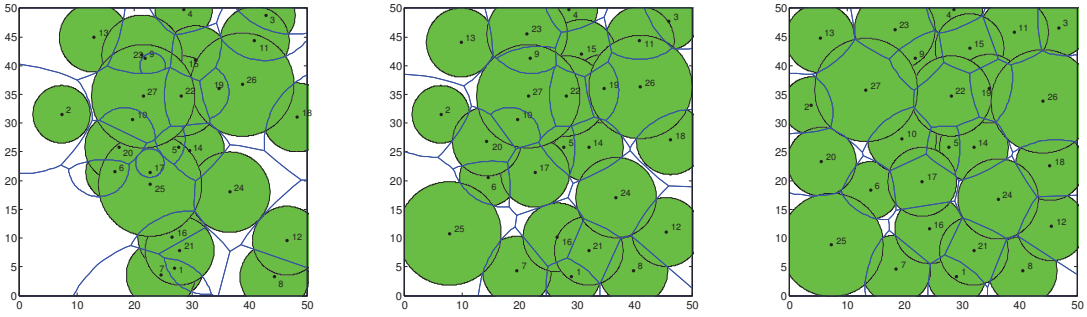
Proof. The proof follows directly from Lemmas 5.8 and 5.9, Definitions 5.7 and 5.8, and Remark 5.4. ■

Using the result of Theorem 5.3, Algorithm 5.6 is developed to calculate the Maxmin-curve centroid of an MW-Voronoi region. The complexity of the algorithm is $O(e_i^4)$, which is typically not very high.

Algorithm 6: An algorithm for finding the Maxmin-curve centroid of the i-th MW-Voronoi region

begin

1) **for** $p = 1, 2, \ldots, e_i - 2$
   for $q = p + 1, p + 2, \ldots, e_i - 1$
   for $t = q + 1, q + 2, \ldots, e_i$
   calculate $\Omega_{p,q,t}$
   if $\Omega_{p,q,t}$ is inside the i-th MW-Voronoi region, **then**
   record it.
   end
   end
   end
   end
2) **for** $r = 1, 2, \ldots, e_i - 1$
   for $s = r + 1, r + 2, \ldots, e_i$
   calculate $\Omega_{r,s}^{A,max}$
   if $\Omega_{r,s}^{A,max}$ is inside the i-th MW-Voronoi region, **then**
   record it.
   end
   end
   end
3) The center of the largest circle is the Maxmin-curve centroid of the i-th MW-Voronoi region.



(a) (b) (c)

Figure 5.16: Snapshots of the execution of the Maxmin-curve strategy. (a) Initial coverage; (b) coverage after the first round, and (c) final coverage.

Given a group of sensors with the same initial configuration as in of Fig. 5.3(a), let the Maxmin-curve strategy be used. It can be verified that after the first round the coverage increases to 84.9%, and eventually reaches 95.1%. This is depicted in Fig. 5.16, where it can be observed that after the final round the sensors are distributed more evenly than the initial configuration, and that the coverage increases considerably.

Theorem 5.4. *The proposed algorithms (WVB, Minmax-curve and Maxmin-curve) are convergent.*

Proof. Let the positions and sensing radii of the sensors in the k-th round be denoted by $\mathbf{P}(k) = \{P_1(k), P_2(k), \ldots, P_n(k)\}$ and $\mathbf{r}(k) = \{r_1(k), r_2(k), \ldots, r_n(k)\}$, respectively. Denote also the MW-Voronoi regions in the k-th round by $\Pi_1(k)$, $\Pi_2(k)$, ..., $\Pi_n(k)$, and the corresponding total covered area of the field by $\beta(k)$. If the k-th round is not the final round, then some sensors move and change their locations in the next round. Assume that the i-th sensor, $i \in \mathbf{n}$, moves to the new location $P_i(k+1) \neq P_i(k)$; if the coverage area w.r.t. this location is greater than the previous local coverage area, i.e. $\beta_{\Pi_i(k)}^{P_i(k+1)} > \beta_{\Pi_i(k)}^{P_i(k)}$, then according to Theorem 5.1 the total coverage in the network increases in this round, i.e. $\beta(k+1) > \beta(k)$. On the other hand, the total covered area is upper-bounded by the overall area of the

field, from which the convergence of the algorithms is implied. ■

It is worth mentioning that the convergence of the proposed algorithms may not be achieved in finite time. As mentioned earlier, in order to terminate the algorithm in finite time, a proper coverage improvement threshold ϵ is defined such that the algorithm will continue after the k-th round only if there is a sensor in the network whose coverage increases at least by ϵ in the following iteration, i.e. $\exists i \in \mathbf{n} : \beta_{\Pi_i(k)}^{P_i(k+1)} \geq \beta_{\Pi_i(k)}^{P_i(k)} + \epsilon$. Note that the choice of ϵ involves a trade-off between network coverage and deployment time. The following theorem provides an upper-bound on the number of rounds required to run the algorithm, as a function of ϵ.

Theorem 5.5. *Consider a set of n mobile sensors $\mathbf{S}$, randomly deployed in a 2D field. Using any of the proposed algorithms with the coverage improvement threshold ϵ, the number of required rounds to run the algorithm is at most $\frac{A_{total}}{\epsilon}$, where A_{total} is the overall area of the field.*

Proof. Let the number of rounds required to run the algorithm in order to meet the termination condition be denoted by ζ_f. Let also the total uncovered area of the field in the k-th round be represented by $\theta(k)$, and note that $\beta(k) = A_{total} - \theta(k)$. Denote the position of the sensors and their corresponding MW-Voronoi regions in the k-th round by $\mathbf{P}(k) = \{P_1(k), P_2(k), \ldots, P_n(k)\}$ and $\Pi_1(k), \Pi_2(k), \ldots, \Pi_n(k)$, respectively. From the properties of the MW-Voronoi diagram, one can conclude that:

$$\theta(k) = \sum_{i=1}^{n} \theta_{\Pi_i(k)}^{P_i(k)}, \ \forall 1 \leq k \leq \zeta_f \tag{5.24}$$

Define the *moving set of the k-th round* as the largest subset of $\mathbf{S}$ that moves in the k-th round, and denote the indices of the sensors in this set by $\mathbf{Indx}(k)$. Note that at least one sensor moves in the k-th round, i.e. $\mathbf{Indx}(k) \neq \emptyset$, $\forall 1 \leq k \leq \zeta_f$. Note also that the i-th sensor, $i \in \mathbf{Indx}(k)$, moves in the k-th round if $\beta_{\Pi_i(k)}^{P_i(k+1)} \geq \beta_{\Pi_i(k)}^{P_i(k)} + \epsilon$. This means that:

$$\theta_{\Pi_i(k)}^{P_i(k+1)} \leq \theta_{\Pi_i(k)}^{P_i(k)} - \epsilon, \quad \forall i \in \mathbf{Indx}(k) \tag{5.25}$$

On the other hand, some of the points in $\theta^{P_i(k+1)}_{\Pi_i(k)}$ might also be covered by another sensor located at $P_j(k+1)$, for some $j \in \mathbf{n}\backslash\{i\}$. Hence:

$$\theta(k+1) \leq \sum_{i=1}^{n} \theta^{P_i(k+1)}_{\Pi_i(k)} \tag{5.26}$$

From the last two relations and on noting that for any $i \in \mathbf{n}\backslash\mathbf{Indx}(k)$ the i-th sensor does not move (which implies $\theta^{P_i(k+1)}_{\Pi_i(k)} = \theta^{P_i(k)}_{\Pi_i(k)}$), one arrives at:

$$\theta(k+1) \leq \sum_{i=1}^{n} \theta^{P_i(k)}_{\Pi_i(k)} - |\mathbf{Indx}(k)|\,\epsilon \tag{5.27}$$

It is now concluded from (5.24) and (5.27) that:

$$\theta(k+1) \leq \theta(k) - |\mathbf{Indx}(k)|\,\epsilon \leq \theta(k) - \epsilon \tag{5.28}$$

or equivalently:

$$\beta(k+1) \geq \beta(k) + |\mathbf{Indx}(k)|\,\epsilon \geq \beta(k) + \epsilon \tag{5.29}$$

which implies that using the underlying sensor relocation scheme, in each round the total covered area increases by at least ϵ. Therefore, the total amount of increased coverage from the first round to the termination round is greater than or equal to $\zeta_f\epsilon$. Since the total covered area is always less than or equal to A_{total}, hence $A_{total} \geq \zeta_f\epsilon$ or equivalently $\frac{A_{total}}{\epsilon} \geq \zeta_f$. ■

Remark 5.6. *The importance of using the MW-Voronoi diagram for nonidentical sensors is that it guarantees the convergence of the proposed deployment algorithms. Note that the monotonically increasing characteristic of the total covered area is guaranteed for the MW-Voronoi partitioning, and not necessarily for conventional Voronoi partitioning. This means that using existing sensor deployment strategies (which are mainly for identical sensors) may lead to non-convergent sensor movements if the sensors are not identical.*

Remark 5.7. *Another important feature of the proposed algorithms is that the sensors will never collide, and hence there is no need to implement any collision avoidance strategy. Note that one of the important properties of the MW-Voronoi diagram*

is that first of all it partitions the field, and also there is exactly one sensor in each region. Since in the proposed algorithms the new location for each sensor is inside the corresponding MW-Voronoi region and each sensor moves within its region, the sensors will not collide.

Remark 5.8. *Note that when the sensors are identical, the Voronoi regions are polygons and one can solve an easy convex optimization problem to find the center of the largest circle inside a polygon, called Maxmin center (the maximum volume ellipsoid in a polyhedron is addressed in [131]). In a network of nonidentical sensors, on the other hand, the boundaries of the MW-Voronoi cells are parts of Apollonian circles (not straight lines). Hence, one cannot find the Minmax-curve and Maxmin-curve centroids by solving a convex optimization problem. Theorems 5.2 and 5.3 are important in the sense that they obtain these centroids using a geometric approach.*

5.3 Simulation Results

The results presented in this section for sensing coverage are all the average values obtained by using 20 random initial configurations for the sensors. Also, the coverage improvement threshold is set to $\varepsilon = 0.1\text{m}^2$, which means if the increase in the local coverage area by every sensor is less than 0.1m^2, then the termination condition is satisfied and there is no need to continue the iterations.

Example 1: In this example, 36 sensors are randomly deployed in a 50m by 50m flat space: 20 with a sensing radius of 6m, 8 with a sensing radius of 5m, 4 with a sensing radius of 7m, and 4 with a sensing radius of 9m. Moreover, the communication range of each sensor is assumed to be 10/3 times its sensing range. The coverage factor of the sensor network (defined as the ratio of the covered area to the overall area) in each round is depicted in Fig. 5.17 for the methods proposed in this chapter. It can be seen from this figure that under all three strategies,

coverage increases significantly in the first few rounds. It can also be observed that the Maxmin-curve strategy performs better than the other two strategies in this example, as far as coverage is concerned.

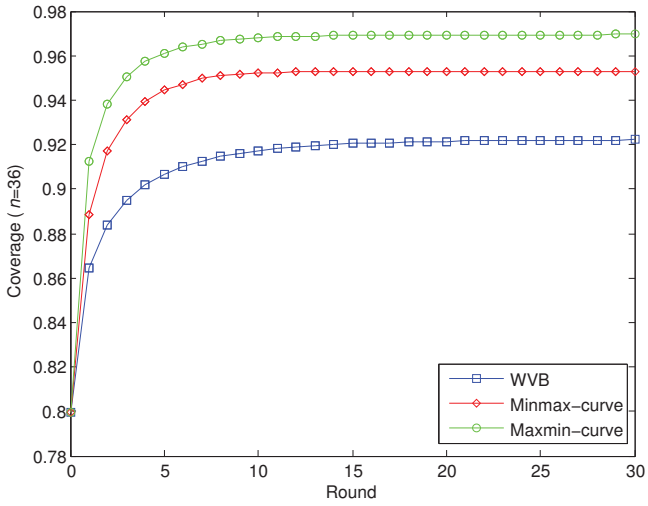


Figure 5.17: Network coverage in different rounds of the proposed algorithms for 36 sensors.

In order to compare the performance of the proposed algorithms for different number of sensors, consider three additional setups: n=18, 27 and 45. Let changes in the number of identical sensors in the new setups be proportional to the changes in the total number of sensors (e.g., for the case of n=45 there will be 25 sensors with a sensing radius of 6m, 10 with a sensing radius of 5m, 5 with a sensing radius of 7m, and 5 with a sensing radius of 9m). Coverage results for different number of sensors are given in Fig. 5.18. It can be observed from this figure that the sensing coverage using the Maxmin-curve algorithm is larger than that using the other two algorithms for different number of sensors. It can also be seen that although the WVB strategy provides better coverage compared to the Minmax-curve strategy when there are a relatively small number of sensors, it is outperformed by other strategies when the number of sensors increases. Note that the new candidate location for each sensor in the Maxmin-curve algorithm is the center of the largest circle inside the corresponding region. Note also that if the sensing radius of a sensor is less than

or equal to the radius of the Maxmin-curve circle of its region, then by moving the sensor to the Maxmin-curve centroid, its sensing circle will be completely inside the MW-Voronoi region, and consequently its local coverage is maximized. Hence, when there are a small number of sensors in the field and the MW-Voronoi regions are relatively large compared to the sensing circle of the sensors, it would be preferable that every sensor moves to the Maxmin-curve centroid of its region such that the sensing area of the sensor is completely inside the region. Thus, for small number of sensors the Maxmin-curve strategy outperforms the other two algorithms. On the other hand, when there are a large number of sensors in the field, the MW-Voronoi regions are relatively small compared to the sensing circle of the sensors, and hence the probability that each sensor covers its MW-Voronoi region by moving to either the Minmax-curve centroid or the Maxmin-curve centroid increases. As a result, for the case of 45 sensors, the performances of both Maxmin-curve and Minmax-curve algorithms are good (close to 100% coverage).

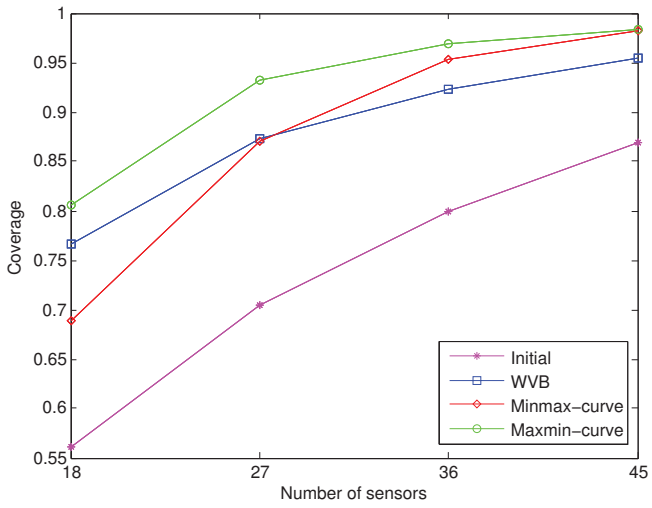


Figure 5.18: Network coverage for different number of sensors using the proposed algorithms.

Another important factor in the performance evaluation of different algorithms is how fast the desired coverage is achieved. Notice that sensor deployment time in each round of all algorithms is more or less the same. Hence, to compare the rate of

convergence of the algorithms, it suffices to check the number of rounds it takes for the sensors to provide certain coverage. Fig. 5.19 shows that in all three strategies the number of rounds required to satisfy a given termination condition increases by adding more sensors up to a certain point, and then starts to decrease. The reason is that the MW-Voronoi regions are relatively large (compared to the sensing circles) when there are a small number of sensors. As a result, it is likely that the sensing area of every sensor is contained within its MW-Voronoi region. Thus, the sensors' local coverage areas do not increase by moving the sensors. On the other hand, when the number of sensors is relatively large (such that the summation of sensing areas is much larger than the overall area of the field), it is likely that every sensor covers its MW-Voronoi region, which in turn means that the termination condition will be met relatively fast [100]. Note also that in the WVB strategy the number of rounds required for the termination of the algorithm is larger than the other strategies. The number of rounds in the Minmax-curve algorithm is relatively low, making it a good candidate as far as the deployment time is concerned.

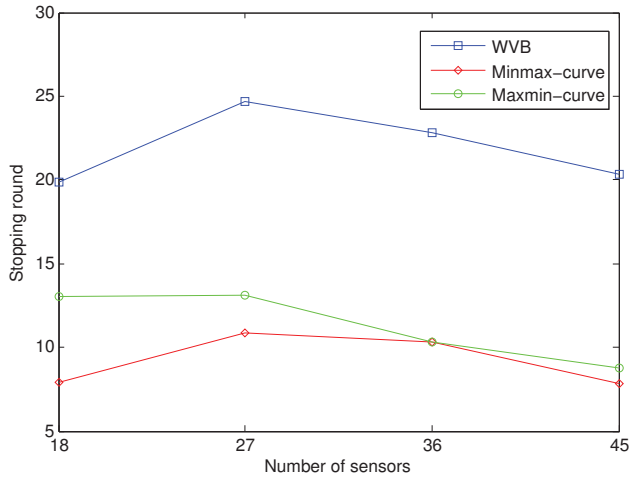


Figure 5.19: The number of rounds required to reach the termination condition for different number of sensors using the proposed algorithms.

Energy consumption of sensors is another important measure of performance in sensor deployment algorithms. The energy consumption of a mobile sensor highly

depends on its traveling distance, as well as the number of times it stops before arriving at the destination (the latter is due to static friction). Hence, one should take the traveling distance as well as the number of movements of each sensor into account in order to compare the energy-efficiency of different sensor deployment algorithms. The average moving distance is provided in Fig. 5.20 for different number of sensors in this example. It is observed from this figure that by increasing the number of sensors, the average moving distance of the sensors decreases in all scenarios. This observation is justified for each algorithm as follows. In the WVB strategy, when the number of sensors increases, the distance between each sensor and its final position decreases, resulting in a decrease in the average moving distance. In the other two algorithms, on the other hand, when the number of sensors increases, the MW-Voronoi regions become smaller. As a result, the distance between each sensor and its destination point in the corresponding MW-Voronoi region decreases, which in turn leads to a decrease in the average moving distance. It can be concluded from Fig. 5.20 that when there are a large number of sensors in the field, the average moving distances in all three strategies are approximately equal. The number of movements versus the number of sensors is given in Fig. 5.21. It can be observed from this figure that when the number of sensors is higher than a certain value (which is different for the three algorithms), the number of movements decreases. This is due to the fact that for a large number of sensors the MW-Voronoi regions become smaller, which helps the sensors cover their MW-Voronoi regions (as noted earlier). As a result, the coverage holes will be covered in a shorter period of time, decreasing the number of movements.

Assume that the energy required for a sensor to travel 1m is equal to 8.268J [130], [135]. Assume also that the energy required to stop a sensor and then overcome its static friction after a complete stop is equal to the energy required for continuously moving the sensor 1m [64], [65]. The energy consumption results for this case are

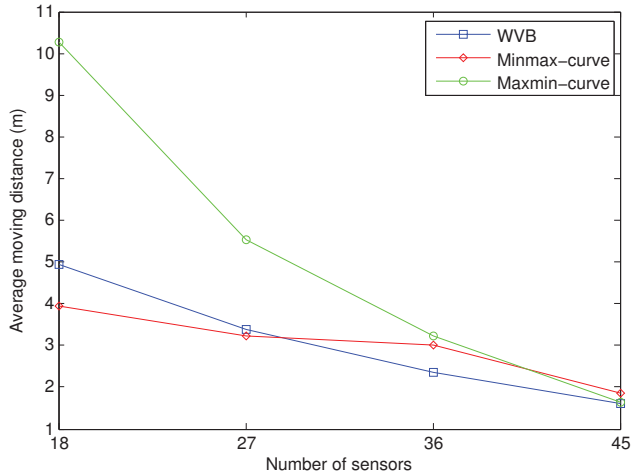


Figure 5.20: The average distance each sensor travels using the proposed algorithms with different number of sensors.

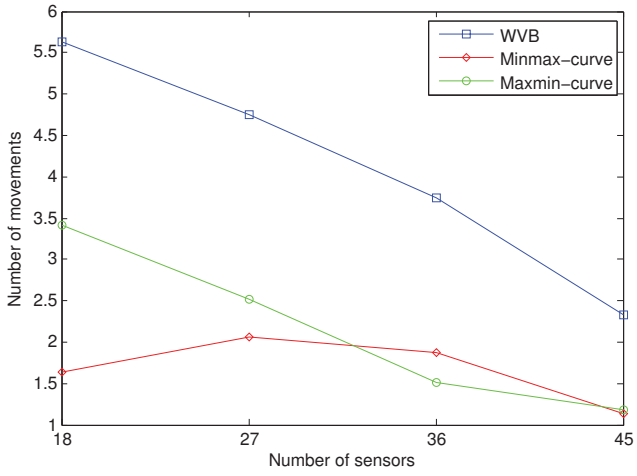


Figure 5.21: The number of movements required to reach the termination condition using the proposed algorithms with different number of sensors.

Table 5.1: The energy consumption of the network in Joule for different number of sensors using the proposed algorithms.

	$n = 18$	$n = 27$	$n = 36$	$n = 45$
WVB	87.4813 J	67.2180 J	50.4572 J	32.6382 J
Minmax-curve	46.1704 J	43.8725 J	40.2928 J	24.6369 J
Maxmin-curve	112.9560 J	66.5397 J	39.0830 J	23.3077 J

summarized in Table I. The results show that for a small number of sensors the Minmax-curve strategy is more efficient than the other two strategies in terms of energy consumption. The Minmax-curve and Maxmin-curve strategies, on the other hand, perform more or less similarly (and better than the WVB strategy) in terms of energy efficiency when there are a large number of sensors in the field.

Remark 5.9. *It is worth mentioning that the algorithms proposed in this chapter differ only in the way the new locations of the sensors are determined. Since the complexity of the algorithm to find the new location of a sensor in the WVB strategy is less than that in the Maxmin-curve and Minmax-curve strategies, the WVB algorithm outperforms the other two algorithms as far as the computational complexity is concerned.*

Example 2: In this example, the performance of the proposed algorithms is evaluated in terms of network coverage in a larger sensing field with a higher number of sensors. Let the sensing field be a 100m $\times$ 100m flat space. Four different settings are considered in the sequel. In the first setting, 60 sensors are considered: 3 with a sensing radius of 5m, 24 with a sensing radius of 6m, 12 with a sensing radius of 7m, and 21 with a sensing radius of 9m. In the second, third and fourth settings, 80, 100 and 120 sensors are considered, respectively, with an increase in the number of sensors of identical sensing radius proportional to the increase in the total number of sensors. In all scenarios, it is assumed that the communication range of each

sensor is 10/3 times its sensing range (e.g, a sensor with a sensing range of 6m has a communication range of 20m). Fig. 5.22 depicts the coverage factor of the sensors in each round for the third scenario (100 sensors). Also, Fig. 5.23 provides the coverage results for different number of sensors for comparison. It can be observed that the results in these figures are very similar to the ones given in Example 1 (Figs. 5.17 and 5.18). Hence, the discussion given in the previous example is also valid here.

The simulation results can be summarized as follows:

- The Maxmin-curve strategy is more desirable in terms of network coverage.
- The WVB strategy is more preferable as far as the computational complexity is concerned.
- The Minmax-curve strategy outperforms the other two algorithms in terms of deployment time.
- As far as the energy consumption is concerned:
  - if there are a large number of sensors in the field, then the Maxmin-curve strategy is more efficient.
  - when the number of sensors in the field is relatively small, the Minmax-curve algorithm outperforms the other two strategies.

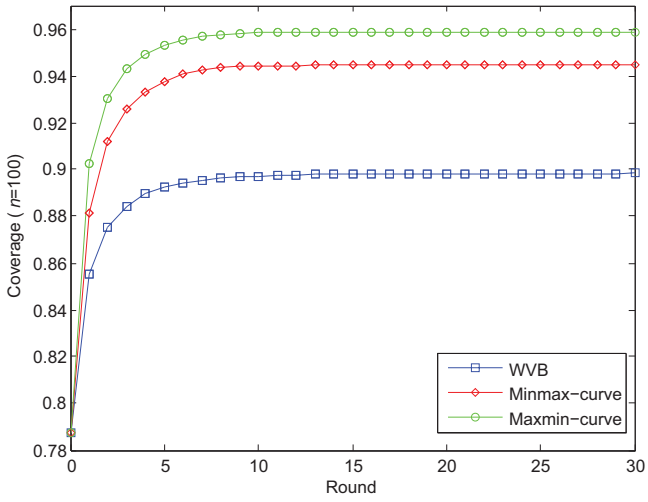


Figure 5.22: Network coverage per round for 100 sensors in Example 2.

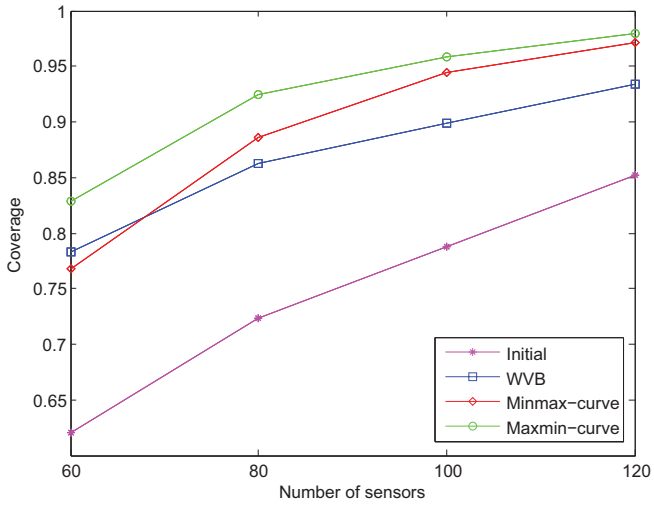


Figure 5.23: Network coverage for different number of sensors using the proposed algorithms in Example 2.

Chapter 6

Self-Deployment Algorithms for Field Coverage in a Network of Nonidentical Mobile Sensors: Vertex-Based Approach

In this chapter, distributed deployment algorithms are proposed for efficient coverage in a mobile sensor network. The proposed algorithms calculate the position of the sensors iteratively based on existing coverage holes in the field. To this end, the multiplicatively weighted Voronoi (MW-Voronoi) diagram is used to partition the field, as it is assumed that the sensors have different sensing ranges. Under the proposed procedures, the sensors move in such a way that the coverage holes in the network are reduced. Simulation results are provided to demonstrate the effectiveness of the deployment schemes proposed in this chapter.

The rest of the chapter is organized as follows. Section 6.1 provides the new algorithms for efficient network coverage, as the main contribution of the chapter, and simulations are given in Section 6.2.

6.1 Deployment Protocols

In this section, three different protocols are developed for a distributed sensor network. The proposed algorithms are iterative, where in each iteration every sensor S_i, $i \in \mathbf{n}$, first broadcasts its sensing radius r_i and location P_i to other sensors, and then constructs its MW-Voronoi region Π_i based on the information it receives from them. It checks the region subsequently to detect the possible coverage holes. If any coverage hole exists, the sensor calculates its target location (but does not move there) in such a way that by moving there the coverage hole would be eliminated, or at least its size would be reduced by a certain threshold. Once the new target location $\acute{P}_i$ is calculated, the coverage area w.r.t. this location, i.e., $\beta_{\Pi_i}^{\acute{P}_i}$ is obtained. If this coverage area is greater than the previous local coverage area, i.e. $\beta_{\Pi_i}^{\acute{P}_i} > \beta_{\Pi_i}^{P_i}$, the sensor moves to the new location; otherwise it remains in its current position. In order to terminate the algorithm in finite time, a proper coverage improvement threshold ϵ is defined such that if the increase in the coverage area by each sensor within its MW-Voronoi region is not sufficiently large (as specified by ϵ), there is no need to continue the iterations.

Theorem 6.1. *Consider a set of n mobile sensors randomly deployed in a 2D field. Using the proposed sensor deployment procedure with the coverage improvement threshold ε, the total coverage in the network increases. Furthermore, the algorithm converges in at most $\frac{A_{total}}{\varepsilon}$ rounds, where A_{total} is the overall area of the field.*

Proof. The proof is similar to the proof of Theorem 5.5 (using also the results of Theorems 5.1 and 5.4), and is omitted here. ■

Notation 6.1. *In the remainder of this chapter, $\mathcal{V}$ represents an MW-Voronoi diagram with n regions (each one corresponding to a sensor). Furthermore, the number of corners of the i-th region is denoted by m_i, for any $i \in \mathbf{n}$.*

Notation 6.2. *Consider a circle of radius r centered at O, denoted hereafter by $\Omega(O,r)$, and a point V in the plane. The intersection of Ω and the extension of VO from O is denoted by $T^V_{\Omega(O,r)}$. The other intersection point of $\Omega(O,r)$ and VO (or its extension) is denoted by $\bar{T}^V_{\Omega(O,r)}$.*

Notation 6.3. *As mentioned before, the boundary curves of an MW-Voronoi region are the segments of some Apollonian circles. The set of all such Apollonian circles for the i-th MW-Voronoi region is denoted by $\mathbf{\Omega}_i$. The sets $\bar{\mathbf{\Omega}}_i$ and $\tilde{\mathbf{\Omega}}_i$ are defined as follows:*

$$\bar{\mathbf{\Omega}}_i = \{\Omega \in \mathbf{\Omega}_i | S_i \in \Omega\}$$

$$\tilde{\mathbf{\Omega}}_i = \{\Omega \in \mathbf{\Omega}_i | S_i \notin \Omega\}$$

The above-mentioned procedure will be used in the next three subsections to develop the farthest point boundary, Maxmin-vertex, and Minmax-vertex algorithms. The target point for each sensor in this procedure is defined in the corresponding subsections.

6.1.1 Farthest Point Boundary Strategy (FPB)

In this algorithm, each sensor moves toward the farthest point in its MW-Voronoi region such that any existing coverage hole in its region can be covered. This point is denoted by $X_{i,far}$ for the i-th region. In fact, once a sensor detects a coverage hole, it calculates the farthest point (using the information about its MW-Voronoi region as well as the coverage holes in that region, as it will be shown later) and moves toward it continuously until $X_{i,far}$ is covered. The following definition is used to calculate the farthest point in each MW-Voronoi region.

Definition 6.1. *The corner points of the i-th MW-Voronoi region (i.e., the intersection of its boundary curves) are denoted by $V_{i1}, V_{i2}, \ldots, V_{im_i}$. These points will*

hereafter be referred to as the MW-Voronoi vertices for the i-th MW-Voronoi region (note that a region may have no vertex). It is to be noted that the farthest point in each MW-Voronoi region lies on the boundary of the region.

Lemma 6.1. *Let E and F be two points on the circle Ω, and V be an arbitrary point in the plane such that T_Ω^V is closer to E than to F (see Fig. 6.1). Then, $VE > VF$.*

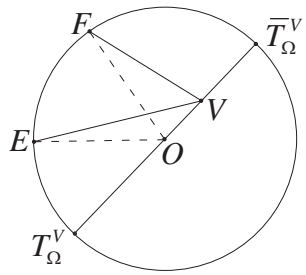


Figure 6.1: An illustrative figure for Lemma 6.1.

Proof. From the law of cosines in triangles VOE and VOF, it results that:

$$VE^2 = VO^2 + OE^2 - 2VO \times OE \times \cos\angle VOE \tag{6.1}$$

$$VF^2 = VO^2 + OF^2 - 2VO \times OF \times \cos\angle VOF \tag{6.2}$$

Since $0 \leq \angle VOF < \angle VOE \leq 180$, hence $\cos\angle VOE < \cos\angle VOF$. From (6.1), (6.2) as well as the relations $OE = OF$ and $\cos\angle VOE < \cos\angle VOF$ it can be concluded that $VE > VF$. This completes the proof. ■

Lemma 6.2. *Given a positive constant $k \neq 1$, let E and F be two points on $\Omega_{AB,k}$ such that $T^A_{\Omega_{AB,k}}$ is closer to E than to F (see Fig. 6.2). Then, $AE > AF$ and $BE > BF$.*

Proof. The proof follows immediately from Lemma 6.1, on noting that $\frac{AE}{BE} = \frac{AF}{BF} = k$. ■

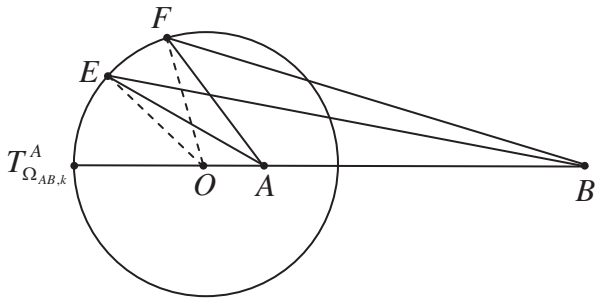


Figure 6.2: An illustrative figure for Lemma 6.2.

Remark 6.1. *It is implied from Lemma 6.2 that for any positive constant $k \neq 1$, $T^A_{\Omega_{AB,k}}$ is the farthest point to A and B, and that $\bar{T}^A_{\Omega_{AB,k}}$ is the nearest point to A and B, among all points on $\Omega_{AB,k}$. For convenience of notation, $T^A_{\Omega_{AB,k}}$ and $\bar{T}^A_{\Omega_{AB,k}}$ will hereafter be denoted by $T_{AB,k}$ and $\bar{T}_{AB,k}$, respectively.*

Lemma 6.3. *Let D be a point and AB be a segment in the plane. Among all points on AB, the farthest point from D is either A or B.*

Proof. The proof is straightforward and is omitted here. ■

Theorem 6.2. *Let $\mathbf{A}_i$ be the set of all vertices for the i-th region ($i \in \mathbf{n}$) of the MW-Voronoi diagram $\mathcal{V}$, and define the set $\mathbf{B}_i$ as follows:*

$$\mathbf{B}_i = \left\{ T_{S_iS_j,k} \mid k = \frac{w_i}{w_j},\ 1 \leq j \leq n,\ S_j \in \mathbf{N}_i \right\}$$

where $\mathbf{N}_i$ is the set of all neighbors of the i-th sensor. Then the farthest point in the i-th region belongs to the union of the sets $\mathbf{A}_i$ and $\mathbf{B}_i$; i.e., $X_{i,far} \in \mathbf{A}_i \cup \mathbf{B}_i$.

Proof. As noted earlier, $X_{i,far}$ lies on the boundary of the i-th region. Consider the following two cases:

Case 1: $X_{i,far}$ is on the boundary curve $V_{i1}V_{i2}$ such that $V_{i1}V_{i2} \in \Omega_{S_iS_g,\frac{w_i}{w_g}}$. If $T_{S_iS_g,\frac{w_i}{w_g}}$ is on the boundary curve $V_{i1}V_{i2}$, then according to Remark 6.1, $X_{i,far} \in \mathbf{B}_i$; otherwise, since among all points on the boundary curve $V_{i1}V_{i2}$ either V_{i1} or V_{i2} is

the nearest point to $T_{S_iS_g,\frac{w_i}{w_g}}$, hence according to Lemma 6.2 $X_{i,far}$ is equal to either V_{i1} or V_{i2}. This means that $X_{i,far} \in \mathbf{A}_i$.

Case 2: $X_{i,far}$ is on the boundary segment $V_{i3}V_{i4}$. In this case, it follows from Lemma 6.3 that $X_{i,far} \in \mathbf{A}_i$.

Therefore, in both cases considered above $X_{i,far} \in \mathbf{A}_i \cup \mathbf{B}_i$. ■

From Theorem 6.2 and on noting that the number of vertices (or boundary curves) of the i-th region is equal to the number of the i-th sensor's neighbors (i.e., $m_i = dim(\mathbf{N}_i)$), one can develop the following algorithm of complexity $O(m_i)$ to calculate the farthest point in the i-th MW-Voronoi region.

Algorithm 7: An algorithm for finding the farthest point in the i-th MW-Voronoi region

---

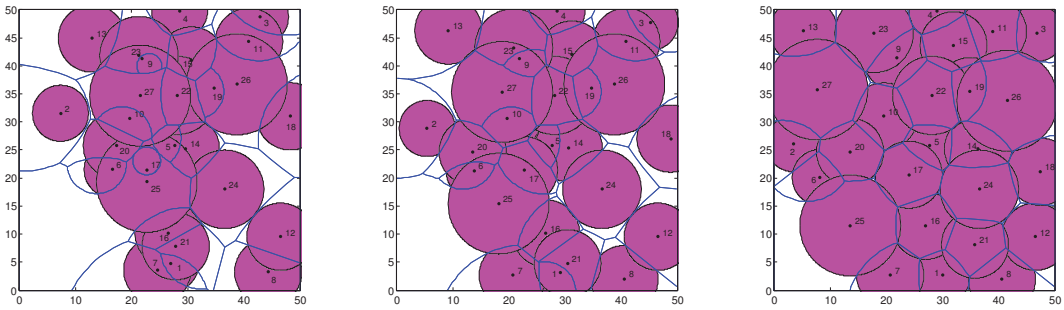
begin

1) **for** all $S_j \in \mathbf{N}_i$
   calculate $T_{S_iS_j,k}$.
   if $T_{S_iS_j,k}$ lies on the boundary of the
   i-th MW-Voronoi region, **then**
   record it.
   end
   end
2) **for** $j = 1, 2, \ldots, m_i$
   record V_{ij}.
   end
3) The point whose distance from S_i is maximum is the farthest point in the i-th MW-Voronoi region.

Since typically an MW-Voronoi region does not have "too many" vertices, the computational complexity of calculating the farthest point is usually not very high.

Fig. 6.3 shows an operational example of FPB Algorithm. In this example, 27 sensors are randomly deployed in a 50m × 50m flat space: 15 with a sensing radius of 6m, 6 with a sensing radius of 5m, 3 with a sensing radius of 7m, and 3 with a sensing radius of 9m. Moreover, the communication range of each sensor is assumed to be 10/3 times its sensing range. In this figure, three snapshots are provided, and in each one the sensing areas of the sensors (filled circles) as well as the MW-Voronoi

regions are depicted. The initial coverage is 66.7%, but after the first round it is improved to 77.2%, and the final coverage is 92.9%. As it can be seen, after the final round the sensors are distributed more evenly than the initial deployment, and that the coverage increases considerably.



(a) (b) (c)

Figure 6.3: Snapshots of the execution of the movement of the sensors under the FPB algorithm. (a) Initial coverage; (b) field coverage after the first round, and (c) final coverage.

6.1.2 Maxmin-Vertex Strategy

The idea behind the Maxmin-vertex strategy is that normally for a good coverage result, none of the sensors should be too close to any of its vertices. In this strategy, the destination for each sensor is selected as a point inside the corresponding MW-Voronoi region whose distance from the nearest vertex is maximized. This point will be referred to as the *Maxmin-vertex centroid*, and will be denoted by $\bar{O}_i$ for the i-th MW-Voronoi region ($i \in \mathbf{n}$). Let the distance between this point and the nearest vertex to it in the i-th region be represented by $\bar{r}_i$. The Maxmin-vertex circle is defined next.

Definition 6.2. *The Maxmin-vertex circle of a region in the MW-Voronoi diagram $\mathcal{V}$ is defined as the largest circle centered inside that region such that all of the vertices of the region are either outside the circle, or on it. This circle is, in fact, $\Omega(\bar{O}_i, \bar{r}_i)$ for the i-th region ($i \in \mathbf{n}$).*

Remark 6.2. *If an MW-Voronoi region has exactly one boundary curve, then this curve is a circle which is also the Maxmin-vertex circle in the Maxmin-vertex strategy.*

Lemma 6.4. *Suppose the i-th region ($i \in \mathbf{n}$) of the MW-Voronoi diagram $\mathcal{V}$ has more than one boundary curve. If the Maxmin-vertex circle passes through exactly one vertex, say V_{i1}, then $\bar{O}_i$ is $T_\Omega^{V_{i1}}$ for some $\Omega \in \mathbf{\Omega}_i$; otherwise, the Maxmin-vertex circle passes through at least two vertices.*

Proof. Let $\bar{V}_{i1}$ be the nearest vertex of the i-th MW-Voronoi region to $\bar{O}_i$, and define:

$$\hat{u} := \min_{V \in \mathbf{V}_i - \{\bar{V}_{i1}\}} \left\{ d(\bar{O}_i, V) \right\}, \quad i \in \mathbf{n} \tag{6.3}$$

where $\mathbf{V}_i$ is the set of vertices of the i-th MW-Voronoi region in the MW-Voronoi diagram.

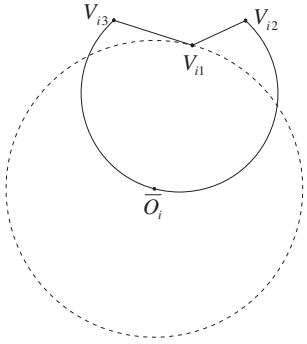


Figure 6.4: An example of the Maxmin-vertex circle, when it passes through exactly one vertex.

Suppose $\bar{O}_i$ and $T_\Omega^{\bar{V}_{i1}}$ are disjoint for any $\Omega \in \mathbf{\Omega}_i$. Suppose also that the Maxmin-vertex circle does not pass through any vertex other than $\bar{V}_{i1}$, and hence the parameter $\delta^* = (\hat{u} - \bar{r}_i)/2$ is strictly positive. There are two possible cases, as discussed below.

Case 1: $\bar{O}_i$ is inside the i-th MW-Voronoi region. Let $\hat{O}$ be a point inside the i-th MW-Voronoi region and on the line $\bar{V}_{i1}\bar{O}_i$, but closer to $\bar{O}_i$, such that the distance

between $\bar{O}_i$ and $\hat{O}$ is equal to a given value $\delta \in (0, \delta^*]$ (see Fig. 6.5(a)).

Case 2: $\bar{O}_i$ is on the boundary of the i-th MW-Voronoi region. Suppose $\bar{O}_i$ is on the curve ϵ. Since $\bar{O}_i$ and $T_\Omega^{\bar{V}_{i1}}$ are distinct for any $\Omega \in \mathbf{\Omega}_i$, one can choose a point $\hat{O}$ on ϵ such that $d(\hat{O}, \bar{V}_{i1}) > d(\bar{O}_i, \bar{V}_{i1})$ and the distance between $\bar{O}_i$ and $\hat{O}$ is equal to a given value $\delta \in (0, \delta^*]$ (see Fig. 6.5(b)).

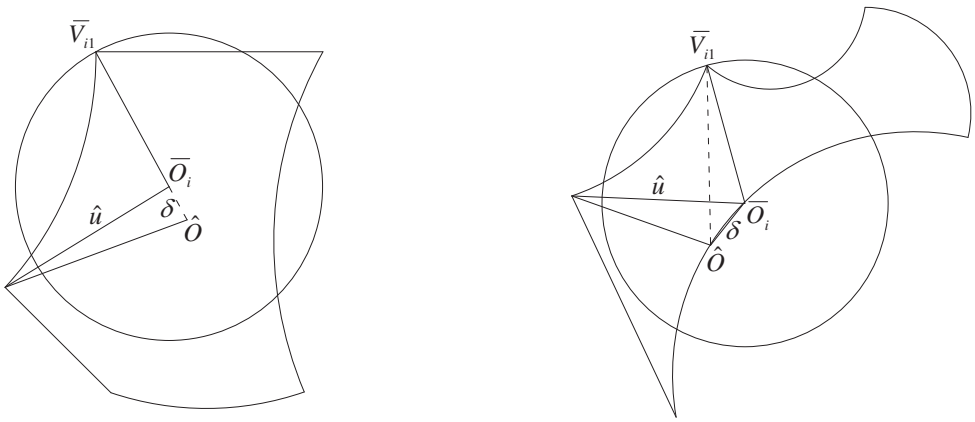
In both cases, according to the triangle inequality:

$$d(\hat{O}, V) \geq d(\bar{O}_i, V) - \delta \geq \hat{u} - \delta, \quad \forall V \in \mathbf{V}_i - \{\bar{V}_{i1}\} \tag{6.4}$$

From the above relation and on nothing that $\hat{u} - \delta \geq \bar{r}_i + \delta > \bar{r}_i$ and $d(\hat{O}, \bar{V}_{i1}) > d(\bar{O}_i, \bar{V}_{i1})$, it can be concluded that

$$\min_{V \in \mathbf{V}_i} \left\{ d(\hat{O}, V) \right\} > \bar{r}_i, \quad i \in \mathbf{n} \tag{6.5}$$

which contradicts the initial assumption that $\bar{O}_i$ is the Maxmin-vertex centroid. This means that there is at least one more vertex on the Maxmin-vertex circle. ■



(a) (b)

Figure 6.5: The Maxmin-vertex centroid, when it is: (a) inside an MW-Voronoi region, and (b) on the boundary of an MW-Voronoi region.

Lemma 6.5. *Consider an MW-Voronoi diagram $\mathcal{V}$, and assume that the Maxmin-vertex circle of one of the regions, say region i ($i \in \mathbf{n}$), passes through exactly two*

vertices, say $\bar{V}_{i1}$ and $\bar{V}_{i2}$. Then $\bar{O}_i$ is the intersection point of the perpendicular bisector of $\bar{V}_{i1}\bar{V}_{i2}$ and the boundary of the i-th MW-Voronoi region.

Proof. Suppose $\bar{O}_i$ is not the intersection point of the perpendicular bisector of $\bar{V}_{i1}\bar{V}_{i2}$ and the boundary of the i-th MW-Voronoi region, i.e., $\bar{O}_i$ is inside the i-th region. Define:

$$\tilde{u} := \min_{V \in \mathbf{V}_i - \{\bar{V}_{i1}, \bar{V}_{i2}\}} \left\{ d(\bar{O}_i, V) \right\}, \quad i \in \mathbf{n} \tag{6.6}$$

Since $\Omega(\bar{O}_i, \bar{r}_i)$ passes through exactly two vertices, thus $\delta^* = (\tilde{u} - \bar{r}_i)/2$ is strictly positive. Let $\tilde{O}$ be a point on the perpendicular bisector of $\bar{V}_{i1}\bar{V}_{i2}$ and outside the triangle $\bar{V}_{i1}\bar{V}_{i2}\bar{O}_i$, but closer to $\bar{O}_i$, such that the distance between the points $\bar{O}_i$ and $\tilde{O}$ is equal to a given value $\delta \in (0, \delta^*]$ (see Fig. 6.6). Using the triangle inequality, one can write:

$$d(\tilde{O}, V) \geq d(\bar{O}_i, V) - \delta \geq \tilde{u} - \delta, \quad \forall V \in \mathbf{V}_i - \{\bar{V}_{i1}, \bar{V}_{i2}\} \tag{6.7}$$

Using (6.7) along with the relations $\tilde{u} - \delta \geq \tilde{u} - \delta^* = \bar{r}_i + \delta^* > \bar{r}_i$ and $d(\tilde{O}, \bar{V}_{i1}) =$

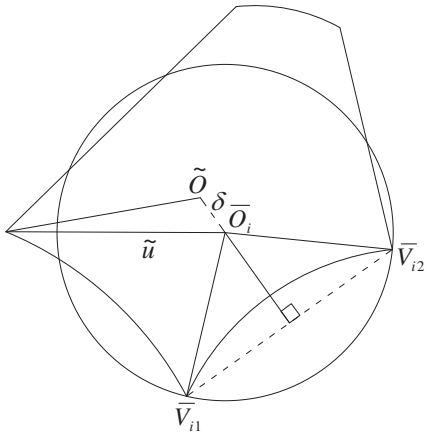


Figure 6.6: An illustrative figure used in the proof of Lemma 6.5.

$d(\tilde{O}, \bar{V}_{i2}) > \bar{r}_i$, one arrives at:

$$\min_{V \in \mathbf{V}_i} \left\{ d(\tilde{O}, V) \right\} > \bar{r}_i, \quad i \in \mathbf{n} \tag{6.8}$$

which contradicts the initial assumption that $\bar{O}_i$ is the Maxmin-vertex centroid. This completes the proof. ■

Definition 6.3. *For convenience of notation, the circle passing through two vertices V_p and V_q of region i in the MW-Voronoi diagram $\mathcal{V}$, centered at the intersection of the perpendicular bisector of V_pV_q and the boundary curve V_kV_l, is denoted by $\Omega^{k,l}_{p,q}$, $k,l,p,q \in \mathbf{m}_i$. Also, the circle passing through three vertices V_p, V_q and V_r of region i is denoted by $\Omega_{p,q,r}$, for $p,q,r \in \mathbf{m}_i$. In addition, the circle passing through one vertex V_r of MW-Voronoi region i, centered at $T^{V_r}_{\Omega}$, is denoted by $\Theta^{V_r}_{\Omega}$, for any $r \in \mathbf{m}_i$ and $\Omega \in \mathbf{\Omega}_i$.*

Theorem 6.3. *Consider an MW-Voronoi diagram $\mathcal{V}$, and suppose that the i-th region ($i \in \mathbf{n}$) has more than one boundary curve. Let $\hat{\mathbf{C}}_i$ and $\grave{\mathbf{C}}_i$ be the sets of all circles $\Omega^{k,l}_{p,q}$, $\forall k,l,p,q \in \mathbf{m}_i$ and $\Theta^{V_r}_{\Omega}$, $\forall r \in \mathbf{m}_i$, $\Omega \in \mathbf{\Omega}_i$, respectively, whose centers are on the boundary of the i-th region, and do not enclose any of the vertices of this region. Let also $\tilde{\mathbf{C}}_i$ be the set of all circumcircles of any three vertices, centered inside the i-th MW-Voronoi region or on its boundary, which do not enclose any of the vertices of this region. Define $\mathbf{C}_i = \hat{\mathbf{C}}_i \cup \grave{\mathbf{C}}_i \cup \tilde{\mathbf{C}}_i$. Then the circle $\Omega(\bar{O}_i, \bar{r}_i)$ belongs to $\mathbf{C}_i$, and it is the largest circle in this set.*

Proof. If $\Omega(\bar{O}_i, \bar{r}_i) \notin \grave{\mathbf{C}}_i$, then according to Lemma 6.4 the Maxmin-vertex circle passes through at least two vertices. If it passes through exactly two vertices, say V_1, V_2, then according to Lemma 6.5, there exist $k,l \in \mathbf{m}_i$ such that $\Omega(\bar{O}_i, \bar{r}_i) = \Omega^{k,l}_{1,2}$. Hence, in this case $\Omega(\bar{O}_i, \bar{r}_i) \in \mathbf{C}_i$, and from Definition 6.2, $\bar{r}_i = \max\{r \,|\Omega(O,r) \in \mathbf{C}_i$. If, on the other hand, the Maxmin-vertex circle passes through three or more Voronoi vertices, then it is the circumcircle of those vertices. Therefore, $\Omega(\bar{O}_i, \bar{r}_i) \in \mathbf{C}_i$, and again it is deduced from Definition 6.2 that $\bar{r}_i = \max\{r \,|\Omega(O,r) \in \mathbf{C}_i\}$. ■

Using the result of Theorem 6.3, the following algorithm of complexity $O(m_i^4)$ is developed to find the Maxmin-vertex centroid in the i-th MW-Voronoi region.

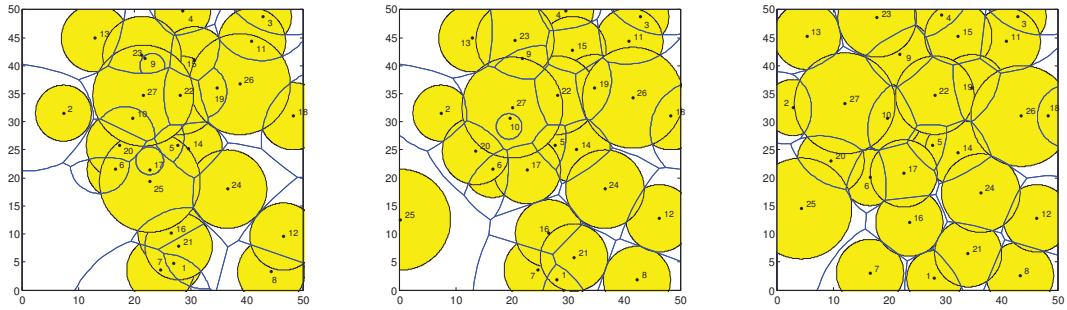
Algorithm 8: An algorithm for finding the Maxmin-vertex centroid in the i-th MW-Voronoi region

begin

1) **for** $p = 1, 2, \ldots, m_i - 2$
   for $q = p + 1, p + 2, \ldots, m_i - 1$
   for $r = q + 1, q + 2, \ldots, m_i$
   calculate $\Omega_{p,q,r}$
   if $\Omega_{p,q,r}$ is centered inside the i-th MW-Voronoi region or on its boundary and does not enclose any of the vertices of the region, **then**
   record it.
   end
   end
   end
   end
2) **for** $p = 1, 2, \ldots, m_i - 1$
   for $q = p + 1, p + 2, \ldots, m_i$
   calculate $\Omega_{p,q}^{k,l}$
   if $\Omega_{p,q}^{k,l}$ does not enclose any of the vertices of the i-th region, **then**
   record it.
   end
   end
   end
3) **for** $r = 1, 2, \ldots, m_i$
   calculate $\Theta_{\Omega}^{V_r}$
   if $\Theta_{\Omega}^{V_r}$ does not enclose any of the vertices of the i-th region, **then**
   record it.
   end
   end
4) The center of the largest circle is the Maxmin-vertex centroid in the i-th MW-Voronoi region.

As in the case of calculating the farthest point, since typically an MW-Voronoi region does not have "too many" vertices, the computational complexity for calculating the Maxmin-vertex centroid is usually not very high. Consider the initial deployment of Fig. 6.3(a), and this time let the Maxmin-vertex algorithm be used.

After the first round, the coverage is improved to 75.5%, and finally it reaches 89.5% (see Fig. 6.7).



(a) (b) (c)

Figure 6.7: Snapshots of the execution of the movement of the sensors under the Maxmin-vertex algorithm. (a) Initial coverage; (b) field coverage after the first round, and (c) final coverage.

6.1.3 Minmax-Vertex Strategy

The idea behind the Minmax-vertex technique is that normally for optimal coverage, each sensor should not be "too far" from any of its MW-Voronoi vertices. The Minmax-vertex strategy selects the target location for each sensor as a point inside the corresponding MW-Voronoi region whose distance from the farthest vertex is minimized. This point will be referred to as the *Minmax-vertex centroid*, and will be denoted by $\check{O}_i$ for the i-th region ($i \in \mathbf{n}$). Furthermore, the distance between this point and the farthest vertex from it in the i-th region will be represented by $\check{r}_i$. The Minmax-vertex circle is defined next.

Definition 6.4. *The Minmax-vertex circle of an MW-Voronoi region is defined as the smallest circle centered inside the region such that all of the vertices of the region are either inside the circle or on it. This circle is, in fact,* $\Omega(\check{O}_i, \check{r}_i)$*, for the i-th region (*$i \in \mathbf{n}$*).*

Remark 6.3. *If an MW-Voronoi region has exactly one boundary curve, then this*

curve is a circle which is also the Minmax-vertex circle for that region in the Minmax-vertex strategy.

Lemma 6.6. *If an MW-Voronoi region has more than one boundary curve, then the corresponding Minmax-vertex circle passes through at least two vertices.*

Proof. Let $\check{V}_{i1}$ be the farthest vertex to $\check{O}_i$ on the boundary of the i-th MW-Voronoi region, and define:

$$\hat{z} := \max_{V \in \mathbf{V}_i - \{\check{V}_{i1}\}} \left\{ d(\check{O}_i, V) \right\}, \quad i \in \mathbf{n} \tag{6.9}$$

Suppose that the Minmax-vertex circle does not pass through any vertex other than $\check{V}_{i1}$, and hence $\delta^* = (\check{r}_i - \hat{z})/2$ is strictly positive. There are two possible cases, as discussed below.

Case 1: $\check{O}_i$ is inside the i-th MW-Voronoi region. Let $\hat{O}$ be a point inside the i-th MW-Voronoi region and on the line $\check{V}_{i1}\check{O}_i$ such that the distance between $\check{O}_i$ and $\hat{O}$ is equal to a given value $\delta \in (0, \delta^*]$ (see Fig. 6.8(a)).

Case 2: $\check{O}_i$ is on the boundary of the MW-Voronoi region. Suppose $\check{O}_i$ is on the curve ϵ. Let $\hat{O}$ be a point on ϵ or inside the i-th MW-Voronoi region such that $d(\hat{O}, \check{V}_{i1}) < d(\check{O}_i, \check{V}_{i1})$, and the distance between $\check{O}_i$ and $\hat{O}$ is equal to a given value $\delta \in (0, \delta^*]$ (see Fig. 6.8(b)).

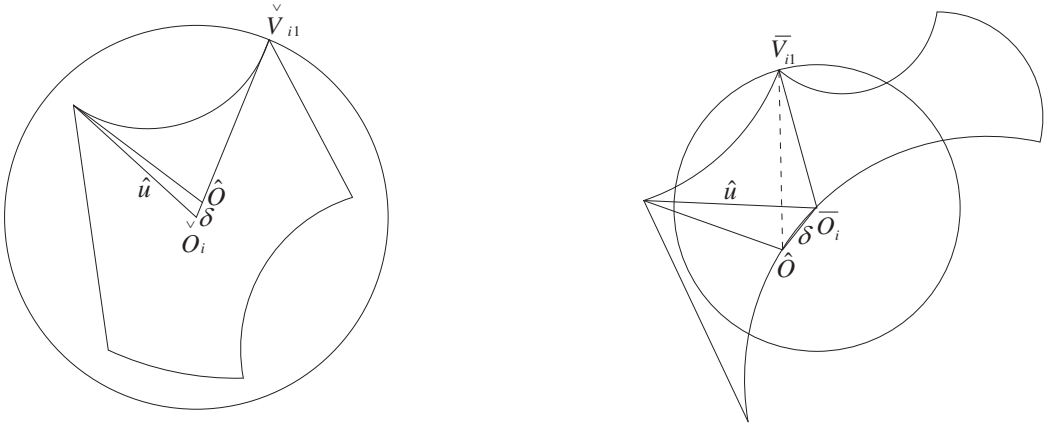
In both cases, according to the triangle inequality:

$$d(\hat{O}, V) \leq d(\check{O}_i, V) + \delta \leq \hat{z} + \delta, \quad \forall V \in \mathbf{V}_i - \{\check{V}_{i1}\} \tag{6.10}$$

From the above relation and on noting that $\hat{z} + \delta \leq \check{r}_i - \delta < \check{r}_i$ and $d(\hat{O}, \check{V}_{i1}) < d(\check{O}_i, \check{V}_{i1})$, it can be concluded that

$$\max_{V \in \mathbf{V}_i} \left\{ d(\hat{O}, V) \right\} < \check{r}_i, \ i \in \mathbf{n} \tag{6.11}$$

which contradicts the initial assumption that $\check{O}_i$ is the Minmax-vertex centroid. This means that there is at least one more vertex on the Minmax-vertex circle. ■



(a) (b)

Figure 6.8: Minmax-vertex centroid, when it is: (a) inside an MW-Voronoi region, and (b) on the boundary of an MW-Voronoi region.

Lemma 6.7. *Consider an MW-Voronoi diagram $\mathcal{V}$, and assume that the Minmax-vertex circle of one region, say region i ($i \in \mathbf{n}$), passes through exactly two vertices, say $\check{V}_{i1}$ and $\check{V}_{i2}$. Then $\check{O}_i$ is the intersection point of the perpendicular bisector of $\check{V}_{i1}\check{V}_{i2}$ and the boundary of the i-th MW-Voronoi region.*

Proof. Suppose $\check{O}_i$ is not the intersection point of the perpendicular bisector of $\check{V}_{i1}\check{V}_{i2}$ and the boundary of the i-th MW-Voronoi region, i.e., $\check{O}_i$ is inside the i-th region. Define:

$$\tilde{z} := \max_{V \in \mathbf{V}_i - \{\check{V}_{i1}, \check{V}_{i2}\}} \left\{ d(\check{O}_i, V) \right\}, \ i \in \mathbf{n} \tag{6.12}$$

Since $\Omega(\check{O}_i, \check{r}_i)$ passes through exactly two vertices, thus $\delta^* = (\check{r}_i - \tilde{z})/2$ is strictly positive. Let $\tilde{O}$ be a point on the perpendicular bisector of $\check{V}_{i1}\check{V}_{i2}$ and inside the triangle $\check{V}_{i1}\check{V}_{i2}\check{O}_i$, but closer to $\check{O}_i$, such that the distance between the points $\check{O}_i$ and $\tilde{O}$ is equal to a given value $\delta \in (0, \delta^*]$ (see Fig. 6.9). Using the triangle inequality, one can write:

$$d(\tilde{O}, V) \leq d(\check{O}_i, V) + \delta \leq \tilde{z} + \delta, \ \forall V \in \mathbf{V}_i - \{\check{V}_{i1}, \check{V}_{i2}\} \tag{6.13}$$

Using (6.13) along with the relations $\tilde{z} + \delta \leq \tilde{z} + \delta^* = \check{r}_i - \delta^* < \check{r}_i$ and $d(\tilde{O}, \check{V}_{i1}) =$

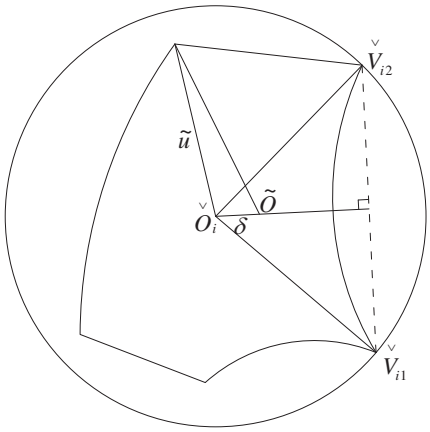


Figure 6.9: An illustrative figure used in the proof of Lemma 6.7.

$d(\tilde{O}, \check{V}_{i2}) < \check{r}_i$, one can conclude that:

$$\max_{V \in \mathbf{V}_i} \left\{ d(\tilde{O}, V) \right\} < \check{r}_i, \ i \in \mathbf{n} \tag{6.14}$$

which contradicts the initial assumption that $\check{O}_i$ is the Minmax-vertex centroid. This completes the proof. ■

Theorem 6.4. *Given an MW-Voronoi diagram $\mathcal{V}$, let $\hat{\mathbf{W}}_i$ be the set of all circles $\Omega^{k,l}_{p,q}$, $\forall k, l, p, q \in \mathbf{m}_i$, whose centers are on the boundary of the i-th region, and all vertices of the region are either inside or on them. Let also $\tilde{\mathbf{W}}_i$ be the set of all circumcircles of any three vertices, centered inside or on the i-th region, with all vertices of the region either inside or on them. Define $\mathbf{W}_i := \hat{\mathbf{W}}_i \cup \tilde{\mathbf{W}}_i$. Then the circle $\Omega(\check{O}_i, \check{r}_i)$ belongs to $\mathbf{W}_i$, and it is the smallest circle in this set.*

Proof. According to Lemma 6.6, the Minmax-vertex circle passes through at least two Voronoi vertices. If it passes through exactly two Voronoi vertices, say V_{i1}, V_{i2}, then according to Lemma 6.7, there exist $k, l \in \mathbf{m}_i$ such that $\Omega(\check{O}_i, \check{r}_i) = \Omega^{k,l}_{1,2}$. Hence, in this case $\Omega(\check{O}_i, \check{r}_i) \in \mathbf{W}_i$, and from Definition 6.4, $\check{r}_i = \min\{r \,|\Omega(O, r) \in \mathbf{W}_i\}$. If, on the other hand, the Minmax-vertex circle passes through three or more

Voronoi vertices, then it is the circumcircle of those vertices. Therefore, $\Omega(\check{O}_i, \check{r}_i) \in \mathbf{W}_i$, and again it is deduced from Definition 6.4 that $\check{r}_i = \min\{r \,|\Omega(O, r) \in \mathbf{W}_i\}$. ■

Using the result of Theorem 6.4, the following algorithm of complexity $O(m_i^4)$ is developed to calculate the Minmax-vertex centroid of the i-th MW-Voronoi region.

Algorithm 9: An algorithm for finding the Minmax-vertex centroid in the i-th MW-Voronoi region

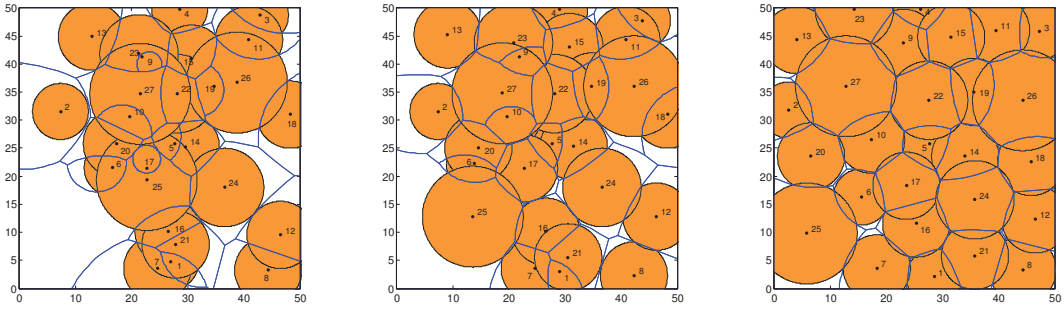
---

begin

1) **for** $p = 1, 2, \ldots, m_i - 2$
   for $q = p + 1, p + 2, \ldots, m_i - 1$
   for $r = q + 1, q + 2, \ldots, m_i$
   calculate $\Omega_{p,q,r}$.
   if $\Omega_{p,q,r}$ is centered inside the i-th MW-Voronoi region or on its boundary, and all the corresponding vertices are either inside the region or on its boundary, **then**
   record it.
   end
   end
   end
   end
2) **for** $p = 1, 2, \ldots, m_i - 1$
   for $q = p + 1, p + 2, \ldots, m_i$
   calculate $\Omega_{p,q}^{k,l}$.
   if all vertices of the i-th region are either inside it or on the boundary of $\Omega_{p,q}^{k,l}$, **then**
   record it.
   end
   end
   end
3) The center of the smallest circle is the Minmax-vertex centroid in the i-th MW-Voronoi region.

As in the two methods presented earlier, since typically a MW-Voronoi region does not have "too many" vertices, the computational complexity for calculating the Minmax-vertex centroid is normally not very high. Using this algorithm with the initial setting of Fig. 6.3(a), the coverage after the first round is improved to 79.9%, and it finally reaches 97.1% (see Fig. 6.10).

Assumption 6.1. *It is implicitly assumed that a synchronization protocol (similar*



(a) (b) (c)

Figure 6.10: Snapshots of the execution of the movement of the sensors under the Minmax-vertex algorithm. (a) Initial coverage; (b) field coverage after the first round, and (c) final coverage.

to the one in [137]) is implemented to guarantee that all sensors start each round at the same time. Furthermore, the coverage rounds are assumed to be sufficiently long, so that each sensor can complete the process of calculating the new location and moving there (if necessary) in one round.

In the next section, it will be shown that all proposed algorithms result in a satisfactory final coverage. In addition, the performance of these algorithms will be compared in terms of the energy consumption of the sensors and deployment speed in reaching the desired coverage level for the sensing field.

6.2 Simulation Results

The three algorithms proposed in Section 6.1 are applied to a flat space of size 50m $\times$ 50m in this section. In each simulation, the algorithm is terminated when none of the sensors' coverage in its corresponding MW-Voronoi region is improved by more than 0.1m^2 in the next move. The results presented in this section for field coverage are all the average values obtained by using 20 random initial deployments for the sensors.

Assume first the same 27 sensors of the example given in Section 6.1. The

coverage factor (defined as the ratio of the covered area to the overall area) of the sensors in each round is depicted in Fig. 6.11 for the three algorithms proposed in this chapter. It can be observed from this figure that all three strategies result in a satisfactory coverage level of the sensing field in the first few rounds of the corresponding algorithms. The resultant curves also show that the Minmax-vertex algorithm performs better than the other algorithms as far as coverage is concerned.

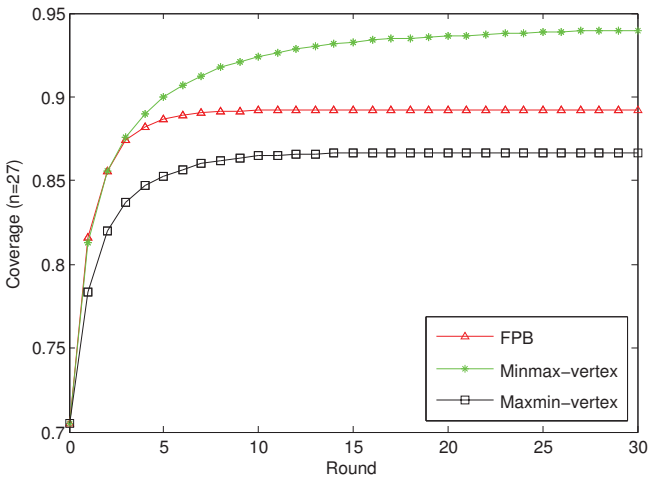


Figure 6.11: Network coverage per round for 27 sensors.

It is desired now to compare the performance of the proposed algorithms in terms of the number of deployed sensors n. To this end, consider three more setups: n=18, 36, and 45 (in addition to n=27 discussed above). Let changes in the number of identical sensors in the new setups be proportional to the changes in the total number of sensors (e.g., for the case of n=18 there will be 10 sensors with a sensing radius of 6m, 4 with a sensing radius of 5m, 2 with a sensing radius of 7m, and 2 with a sensing radius of 9m). Fig. 6.12 provides the coverage results for different number of sensors. It can be seen from this figure that the network coverage in Minmax-vertex algorithm is larger than that in the other algorithms for different number of sensors.

The time it takes for the sensors to provide the desired coverage level is another important measure of the efficiency of the algorithms. Since the deployment time of

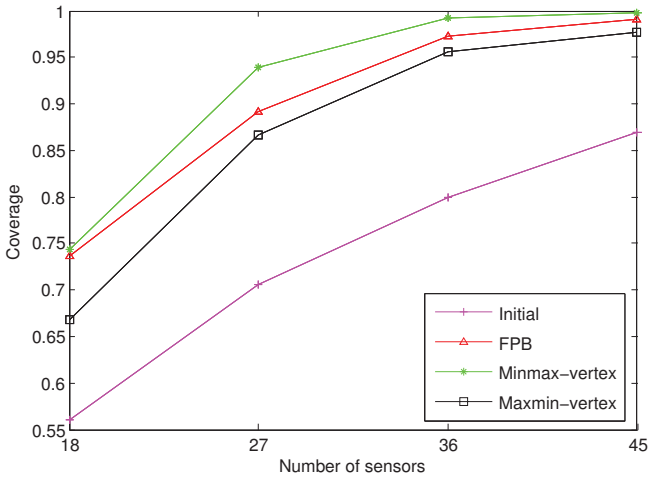


Figure 6.12: Network coverage for different number of sensors using the proposed algorithms.

the sensors in each round is almost equal for all algorithms, the number of rounds required for the sensors to reach a certain coverage level is used to evaluate the time efficiency. It is shown in Fig. 6.13 that in all three algorithms the number of rounds required to meet the termination condition specified earlier, increases by increasing the number of sensors up to a certain value (which varies for different algorithms), and then starts to decrease by adding more sensors. This is mainly because when there are a small number of sensors in the field, the MW-Voronoi regions are large in comparison with the corresponding sensing circles. Hence, there is a good chance that each sensor's local coverage area is completely inside its MW-Voronoi region, which means that the sensor does not need to move in order to increase its coverage area. On the other hand, when there are a large number of sensors in the field, there is a good chance that each sensor covers its MW-Voronoi region (and hence there are no coverage holes), which implies that the termination condition will be satisfied in a short period of time. It can be seen from Fig. 6.13 that in the Minmax-vertex algorithm the number of rounds required for the termination of the algorithm is larger than the other strategies. When the number of sensors in the field is not large, the number of rounds in the FPB algorithm is smaller than the

other algorithms. Hence, the FPB algorithm is more efficient in such cases, as far as the deployment time is concerned. On the other hand, if there are a large number of sensors in the field, the Maxmin-vertex outperforms the other two algorithms in terms of deployment time.

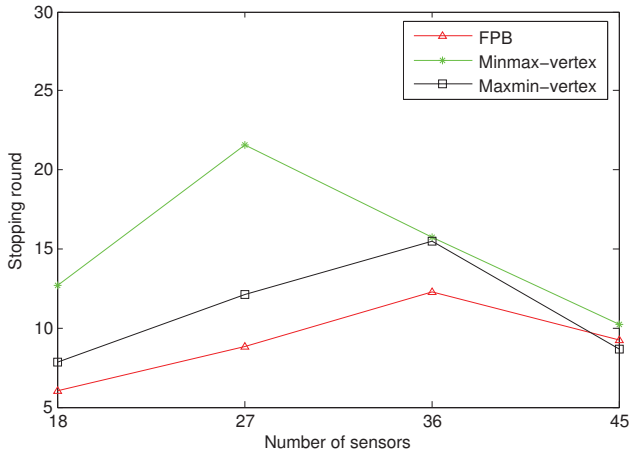


Figure 6.13: The number of rounds required to reach the termination conditions for different number of sensors using the proposed algorithms.

Another important means of assessing the performance of the sensor deployment algorithms is the energy consumption of the sensors. The consumed movement energy of each sensor is known to be directly related to its traveling distance, as well as the number of times it stops (the latter one is related to the static friction). Thus, to compare the proposed methods in terms of energy consumption, the traveling distance and the number of movements should be taken into consideration. Fig. 6.14 depicts the average moving distance for different number of sensors, using the three algorithms. This figure shows that by increasing the number of sensors, the average moving distance is decreased in all scenarios. This is due to the fact that the MW-Voronoi regions become smaller when the number of sensors increases. Note that a decrease in the size of an MW-Voronoi region translates to a smaller distance between the corresponding sensor and its destination point in that region. This in turn leads to a decrease in the average moving distance. It is shown in Fig. 6.14 that

the traveling distance in the FPB algorithm is shorter than that in the other two algorithms. It can be seen from Fig. 6.14 that the average moving distances for all three algorithms are more or less the same when there are a large number of sensors in the field. The number of movements versus the number of sensors is depicted in Fig. 6.15, where it is shown that if the number of sensors increases from 18 to 27, the number of movements increases as well. It can also be observed from this figure that when the number of sensors increases beyond 36, the number of movements decreases. This is due to the fact that for large number of sensors the MW-Voronoi regions become smaller, and hence the sensors will likely cover their MW-Voronoi regions and will not need to move. As it can be observed from Fig. 6.15, when there are a relatively large number of sensors in the field, the number of movements in Maxmin-vertex algorithm is less than that in the other algorithms.

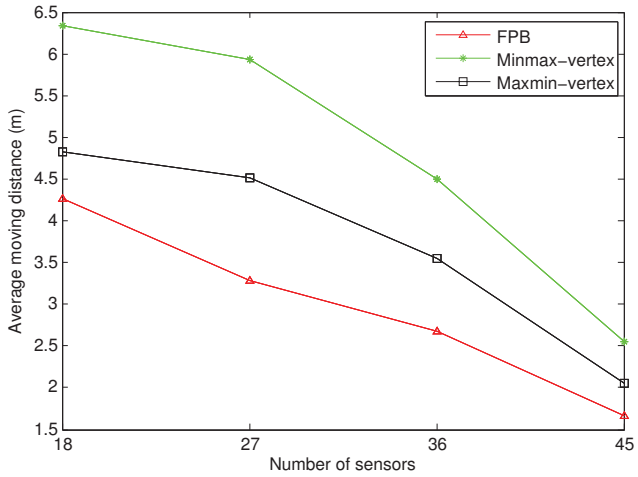


Figure 6.14: The average distance each sensor travels for different number of sensors, using the proposed algorithms.

Let the required energy for traveling 1m (without stopping) be 8.268J (or 0.210J/inch) [130], [135]. Consider two scenarios, where the energy required to stop a sensor and then overcome its static friction after a complete stop is equal to the energy required to continuously move the sensor 1m (first scenario) and 4m (second scenario) [64], [65]. Tables 6.1 and 6.2 provide a summary of the energy

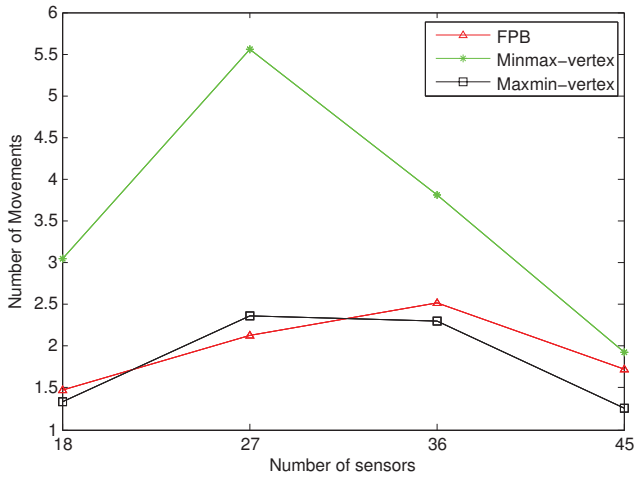


Figure 6.15: The number of movements required for different number of sensors using the proposed algorithms.

consumption results for these two cases. Define α as the ratio of energy consumption due to one stop followed by one move from complete stop to energy consumption due to one meter move. Now, if there are a large number of sensors in the field and the power required to overcome static friction of a sensor is much larger than that required to move it (per unit), the Maxmin-vertex algorithm outperforms the other two algorithms in terms of energy consumption. If, on the other hand, the power required to overcome static friction of a sensor is much smaller than that required to move it, then regardless of the number of sensors the FPB algorithm performs better than the other two algorithms in terms of energy consumption.

Remark 6.4. *Note that the algorithms introduced in this chapter differ only in the way the new locations of the sensors are determined. As mentioned before, the complexity of the algorithm to find the new location of the i-th sensor in the FPB strategy is of order $O(m_i)$, while it is of order $O(m_i^4)$ in the Minmax-vertex and Maxmin-vertex algorithms. Hence, the FPB algorithm outperforms the other two algorithms as far as the computational complexity is concerned.*

The above discussion is summarized below (logic and's in these statements are capitalized):

Table 6.1: The energy consumption in Joule for different number of sensors using the proposed algorithms for the case when the energy required to stop a sensor and then overcome static friction after a complete stop is equal to the energy required to move the sensor 1m non-stop.

	$n = 18$	$n = 27$	$n = 36$	$n = 45$
FPB	47.4234 J	44.7480 J	42.9542 J	28.1041 J
Minmax-vertex	77.6335 J	95.0093 J	68.6495 J	37.0676 J
Maxmin-vertex	50.9095 J	56.8741 J	48.2570 J	27.3611 J

1. The Minmax-vertex algorithm is more preferable as far as network coverage is concerned.

2. The Maxmin-vertex algorithm outperforms the other two algorithms when there are a large number of sensors in the field, AND:

   - the deployment time is the main concern.
   - the energy consumption is the main concern, AND the power required to overcome the static friction of a sensor is much larger than that required to move it (per unit).

3. The FPB algorithm is more desirable when:

   - the deployment time is the main concern AND the number of sensors in the field is not large.
   - the energy consumption is the main concern, AND the power required to overcome the static friction of a sensor is much smaller than that required to move it (per unit).
   - the computational complexity is concerned.

Table 6.2: The energy consumption in Joule for different number of sensors using the proposed algorithms for the case when the energy required to stop a sensor and then overcome static friction after a complete stop is equal to the energy required to move the sensor 4m non-stop.

	$n = 18$	$n = 27$	$n = 36$	$n = 45$
FPB	83.8715 J	97.3876 J	105.4810 J	71.0150 J
Minmax-vertex	153.3546 J	232.9012 J	163.1114 J	84.8015 J
Maxmin-vertex	83.8437 J	115.5310 J	105.1340 J	58.4763 J

Chapter 7

Distributed Deployment Algorithms for Coverage Improvement in a Network of Wireless Mobile Sensors: Relocation by Virtual Force

Efficient deployment algorithms are developed in this chapter to increase coverage in a network of wireless mobile sensors. The proposed strategies iteratively compute the position of the sensors based on existing coverage holes. These holes are obtained using a Voronoi diagram for the case of identical sensors, and a multiplicatively weighted Voronoi (MW-Voronoi) diagram for the case of sensors with different sensing ranges. Each sensor is driven by virtual forces applied to it from the vertices and boundaries of its Voronoi cell. These forces are obtained in such a way that when the sensor is relocated, the covered area of the corresponding cell increases. Simulation results demonstrate the efficacy of the proposed strategies,

and their superiority to existing algorithms.

The plan of the rest of the chapter is as follows. Section 7.1 provides the new algorithms for efficient coverage in a network of identical sensors. These algorithms are extended to the case of nonidentical sensors in Section 7.2. Simulations are given in Section 7.3, which demonstrate the efficacy of the proposed deployment strategies.

7.1 Deployment Protocols for Identical Sensors

Three sensor deployment algorithms are introduced in this section for efficient coverage in a network of identical sensors. The proposed algorithms are iterative, and each iteration consists of four phases. In the first phase, every sensor S_i, $i \in \mathbf{n}$, transmits its position information P_i to other sensors, receives similar information from other sensors, and then constructs its Voronoi polygon. In the second phase, every sensor checks its Voronoi polygon to find its local coverage hole. If a coverage hole exists in a polygon, say the i-th polygon, then a proper scheme is used to find a point $\acute{P}_i$ in it such that by placing the sensor there, the coverage hole would be eliminated, or at least its size would be reduced by a certain threshold. Once the new destination is found, the coverage area w.r.t. this location (i.e. $\beta_{\Pi_i}^{\acute{P}_i}$) is obtained in the third phase. If the coverage area w.r.t. the new destination is greater than the local coverage area, i.e. $\beta_{\Pi_i}^{\acute{P}_i} > \beta_{\Pi_i}^{P_i}$, the sensor moves there; otherwise, it stays at its present location. In order to terminate the algorithm in finite time, a proper threshold ϵ is considered such that if the increase in the coverage area by each sensor within its Voronoi polygon is not greater than ϵ, the iterations stop. Note that the algorithms developed in this section differ in the second phase only. In each round, the new destination for each sensor is computed based on a proper deployment strategy.

As noted above, under the strategies proposed in this work each sensor moves to its new destination only if its coverage area w.r.t. the new location in the old Voronoi polygon increases.

7.1.1 Vertex Virtual Forces (VVF) Strategy

The movement of each sensor under the VVF strategy depends on the vector sum of all virtual forces applied to it from the vertices of its polygon. Denote the vertices of the i-th Voronoi polygon by $\mathbf{V}_i = \{V_{i1}, V_{i2}, \ldots, V_{il}\}$. Also, denote by $d_{V_{ij}}^{S_i}$ the distance between sensor S_i and the j-th vertex of its Voronoi polygon. Let the sensing radius of every sensor be r. In this strategy, if the distance between sensor S_i and vertex V_{ij} is less than r, then a virtual force form V_{ij} will push S_i, tending to move the sensor away by $r - d_{V_{ij}}^{S_i}$. If on the other hand $d_{V_{ij}}^{S_i} > r$, then a virtual force form V_{ij} will pull the sensor, tending to move it toward V_{ij} by $d_{V_{ij}}^{S_i} - r$. Eventually, each sensor moves in the direction of the vector sum of all virtual forces, $\overrightarrow{V_v^i}$, applied to it from the vertices of the corresponding Voronoi polygon. The new destination $\acute{P}_i$ is equal to $P_i + \alpha\overrightarrow{V_v^i}$, where α is a parameter (not necessarily constant) which is to be chosen properly. For example, a line search procedure can be used to find the optimal value for α in order to maximize the coverage area w.r.t. the new destination $\acute{P}_i$. However, in this chapter α is chosen as $\frac{1}{4}$ based on simulation to reduce the computational complexity of the strategy [102].

Fig. 7.1 shows an illustrative example of the VVF strategy. In this figure, the virtual forces applied to the sensor are depicted by dashed vectors, and the displacement $\frac{1}{4}\overrightarrow{V_v^i}$ is shown by a red vector.

7.1.2 Edge Virtual Forces (EVF) Strategy

The sensor deployment strategy introduced in the previous subsection is vertex-based, as it operates according to the distances of the sensors from the vertices

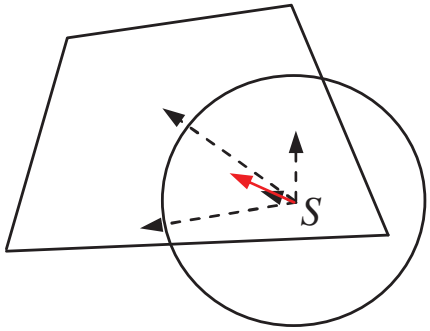


Figure 7.1: An illustrative example of the VVF strategy.

of their Voronoi polygons. While this algorithm proves effective in many practical scenarios, it may not be as effective for specific sensor configurations. For example, consider the polygon in Fig. 7.2, where the sensor is denoted by S. It can be easily verified that in order to increase the coverage area, the sensor should move in the up-left direction. However, under the VVF strategy the sensor is forced (by the corresponding virtual forces) to move in almost the opposite direction (more precisely, to point A, in this specific configuration), although the movement adjustment scheme described earlier does not allow the sensor to move. To address this shortcoming of the VVF algorithm, an edge-based method is presented in the sequel.

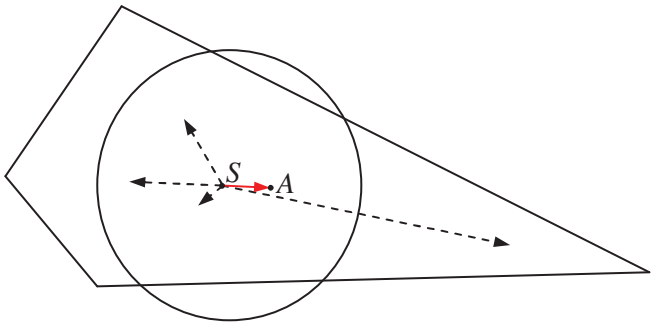


Figure 7.2: A configuration for which the VVF technique is not as effective.

Denote the set of the edges of the i-th Voronoi polygon by $\mathbf{E}_i = \{E_{i1}, E_{i2}, \ldots, E_{il}\}$, and the distance between sensor S_i and the j-th edge of its Voronoi polygon by $d_{E_{ij}}^{S_i}$. In the EVF method, the movement of each sensor results from the vector

sum of all virtual forces applied to it from the edges of its polygon. If the distance between sensor S_i and edge E_{ij} is greater than the sensing radius r, then a virtual force form E_{ij} will pull S_i, tending to move the sensor toward the edge by $d_{E_{ij}}^{S_i} - r$. If on the other hand $d_{E_{ij}}^{S_i} < r$, then a virtual force form E_{ij} will push S_i, tending to move it away by $r - d_{E_{ij}}^{S_i}$. Similar to the VVF strategy, the movement of each sensor is proportional to the vector sum of all virtual forces applied to it from the edges of the corresponding Voronoi polygon, i.e. $\acute{P}_i = P_i + \beta \overrightarrow{V_e^i}$, where β is a given constant. Fig. 7.3 shows an illustrative example of the EVF strategy, for $\beta = \frac{1}{4}$. In this figure, the virtual forces applied from the edges to the sensor are shown by dotted vectors, and the displacement $\frac{1}{4}\overrightarrow{V_e^i}$ is depicted by a red vector.

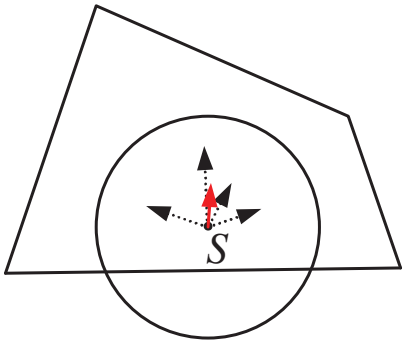


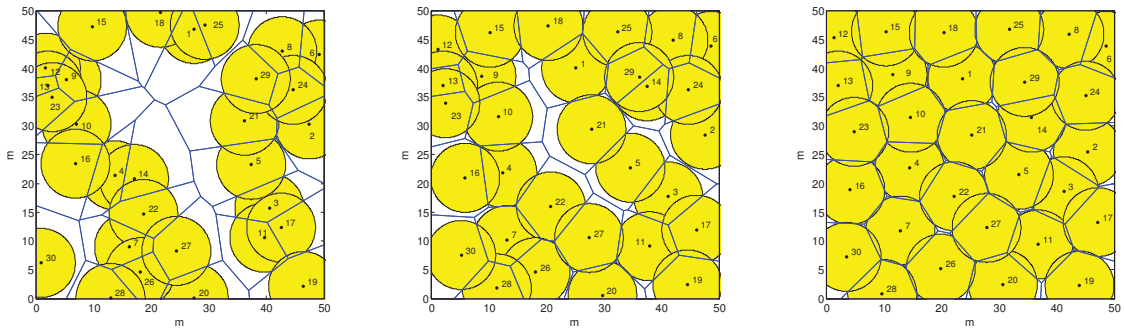
Figure 7.3: An illustrative example of the EVF strategy.

7.1.3 Vertex-Edge Virtual Forces (VEVF) Strategy

The effectiveness of each of the two deployment strategies described so far depends on the relative position of sensors w.r.t. each other. One can take advantage of the strengths of both techniques, by developing a new algorithm as a combination of the VVF and EVF strategies. In this algorithm, which is referred to as the VEVF strategy, every sensor selects two points in each round, as its potential new location: one point according to the VVF strategy, and the other one according to the EVF technique. Any of these two points from which the sensor coverage improves the

most is subsequently selected as the target location of the sensor.

An operational example of the VEVF strategy is given in Fig. 7.4, where 30 sensors with a communication radius of 20m and sensing radius of 6m are randomly deployed in a 50m by 50m plane. The positions of the sensors are shown for three snapshots along with their sensing areas (filled circles) and the resultant Voronoi diagrams. After the first round of the algorithm, the coverage increases from the initial value of 69.05% to 87.76%, and in the final round (where the termination condition is satisfied) the coverage is 98.14%. The figure also demonstrates that the sensors become more evenly distributed as network coverage increases.



(a) (b) (c)

Figure 7.4: Snapshots of the execution of the VEVF strategy. (a) Initial coverage; (b) coverage after the first round, and (c) final coverage.

7.2 Deployment Protocols for Nonidentical Sensors

When sensors have different sensing radii, a point which is not covered by the generating sensor of the polygon containing that point, may be covered by a neighboring sensor. This means that for a network of nonidentical mobile sensors, the conventional Voronoi diagram is not as useful for development and analysis of sensor

deployment strategies. The multiplicatively weighted Voronoi (MW-Voronoi) diagram is used in such networks, as described in the next subsection.

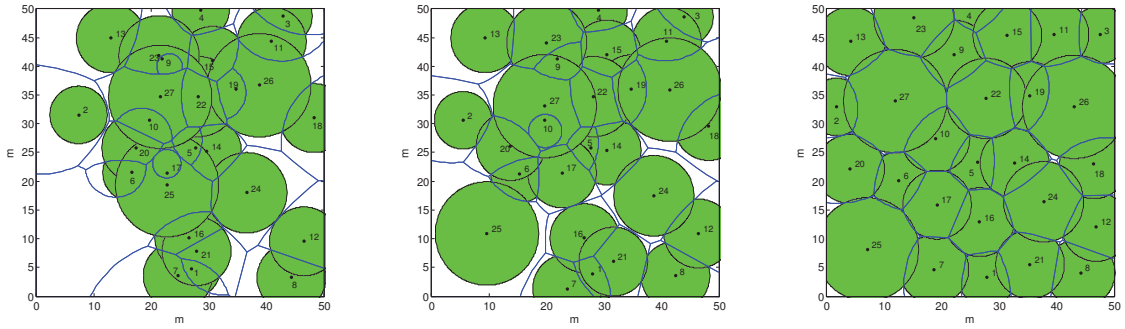
7.2.1 Deployment Protocols

Deployment algorithms similar to the ones developed in the previous section for a network of identical mobile sensors can also be developed for the case of a network of nonidentical sensors by using the MW-Voronoi partitions. Note that the boundaries of the regions in this case are parts of some Apollonian circles, and are not straight edges in general. Note also that an MW-Voronoi region will not have any vertices when the region is a circle, in which case the center of this circle is considered as the new location of the corresponding sensor. The corner points and boundary curves of an MW-Voronoi region can be regarded as the vertices and edges of that region, respectively. Then, analogously to the VVF, EVF and VEVF strategies developed in the previous section, one can introduce the corner point virtual forces (CPVF), boundary curve virtual forces (BCVF), and point-curve virtual forces (PCVF) strategies, respectively.

Remark 7.1. *The complexity of calculating a new sensor destination in all algorithms proposed in this chapter is of order $O(m_i)$ or $O(e_i)$, where m_i and e_i are the number of vertices and edges of the i-th Voronoi polygon (or MW-Voronoi region), respectively. Since typically a Voronoi polygon (or MW-Voronoi region) does not have a large number of vertices and edges, the complexity of the proposed techniques for computing the new sensor destinations is usually not very high.*

An operational example of the CPVF Algorithm is shown in Fig. 7.5. In this example, 27 sensors are randomly placed in a 50m $\times$ 50m flat field: 3 with a sensing radius of 9m, 3 with a sensing radius of 7m, 6 with a sensing radius of 5m, and 15 with a sensing radius of 6m. Furthermore, the communication range of each sensor is assumed to be 10/3 times its sensing range. Three snapshots are given

in this figure, where the sensing areas of the sensors (filled circles) as well as the MW-Voronoi regions are sketched in each snapshot. Using this algorithm with the initial setting shown in Fig. 7.5(a), the coverage increases from 66.7% to 82.6% after the first round, and eventually converges to 98.6%.



(a) (b) (c)

Figure 7.5: Snapshots of the execution of the movement of the sensors under the CPVF algorithm. (a) Initial coverage; (b) field coverage after the first round, and (c) final coverage.

7.3 Simulation Results

Simulations are often used for the evaluation and comparison of different sensor deployment algorithms. In this section, the performance of the proposed sensor deployment algorithms are evaluated and compared using several simulations with random initial sensor configurations. In the examples given below, the average results are depicted by performing 100 simulations with random initial configurations for Examples 1 and 2, and 20 simulations (also with random initial configurations) for Example 3.

Example 1: In this example, the performance of the strategies introduced in Section 7.1 will be compared for different number of sensors: n=20, 30, 40, and 50. The sensors are randomly deployed in a 50m $\times$ 50m flat field, and their sensing and communication ranges are 6m and 20m, respectively. The algorithms used in

this example are terminated when no sensor's coverage in its corresponding Voronoi polygon increases by more than 1% in the next move. In Fig. 7.6, the final coverage factor (defined as the ratio of the covered area to the overall area) is depicted under the three algorithms introduced in Section 7.1, for different number of sensors. It can be observed from this figure that the coverage area under the VEVF strategy is larger than that under the VVF and EVF algorithms, for different number of sensors.

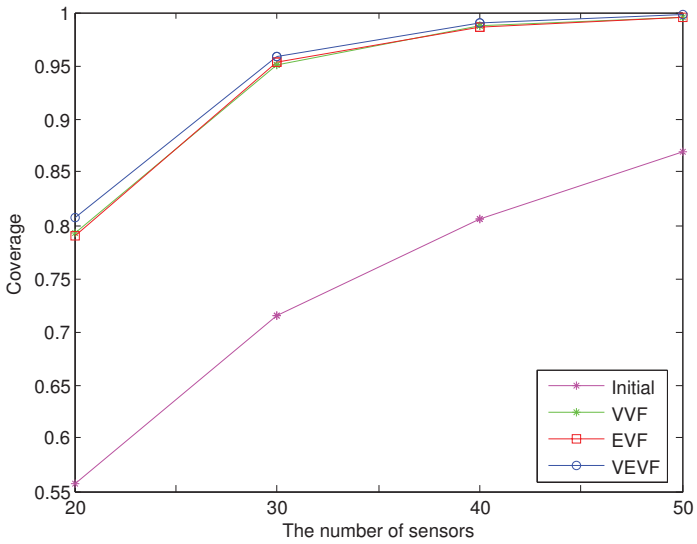


Figure 7.6: Coverage factor for different number of sensors in Example 1 using the proposed algorithms.

Convergence rate is another important issue in the performance evaluation of sensor deployment algorithms. Since the sensor deployment time in each round of different algorithms is almost equal, the number of rounds required for the sensors to meet the termination condition can be used to assess time efficiency. It is shown in Fig. 7.7 that in all three algorithms, the number of rounds (required to meet a certain termination condition) decreases as the number of sensors increases between 30 and 50. This is due to the fact that when the number of sensors in the target field is large, the Voronoi polygons are small compared to the corresponding sensing circles. As a result, it is likely that each sensor covers a large portion of its Voronoi

polygon; thus, the algorithm reaches the termination condition faster. The number of rounds in the EVF algorithm is relatively low, and hence it is more desirable as far as the convergence rate is concerned.

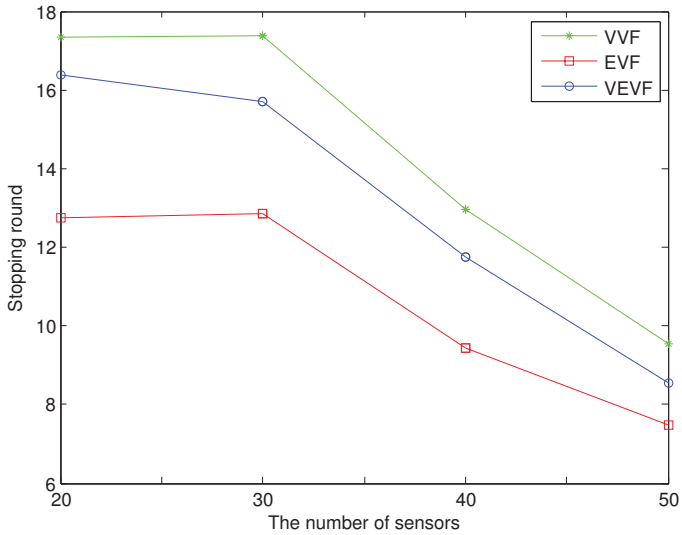


Figure 7.7: The number of rounds required to meet the termination condition in Example 1 for different number of sensors using the proposed strategies.

The energy consumed by sensors in order to provide the desired coverage level is another important issue which needs to be taken into consideration when evaluating the efficiency of different algorithms. The energy consumption of the network highly depends on the traveling distance of the sensors, and also the number of times they stop before arriving at their next position (note that once a sensor stops, it has to overcome the static friction in order to move again). Thus, to compare the energy-efficiency of the proposed methods, one should take the traveling distance and number of movements into account. Fig. 7.8 provides the average moving distance vs. the number of sensors for all three algorithms. This figure shows that the average moving distance decreases by increasing the number of sensors in all three algorithms. This results from the fact that when there are a large number of sensors, the Voronoi polygons are relatively small. This, in turn, decreases the distance between each sensor and its destination point in the corresponding Voronoi

polygon. Therefore, the average moving distance of the sensors under all three algorithms decreases. It can also be observed from Fig. 7.8 that the average moving distance using the VVF method is less than that using the other two techniques.

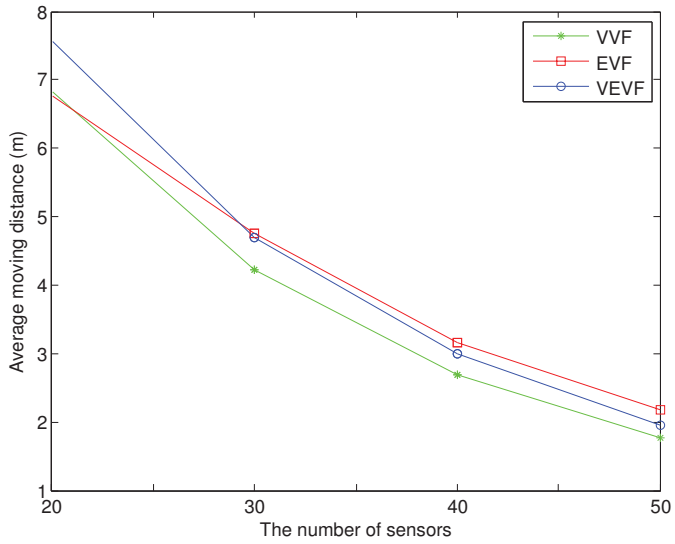


Figure 7.8: The average travel distance for different number of sensors in Example 1, using the proposed algorithms.

The number of sensor movements for different number of sensors is given in Fig. 7.9, which shows that the number of movements in all scenarios decreases by increasing the number of sensors. This results from the fact that when there are a large number of sensors in the field, the Voronoi polygons are small, and hence it is likely that the sensors cover a large area of their Voronoi polygons. Thus, the termination condition will be satisfied in a shorter period of time, which, in turn, decreases the number of sensor movements. It can also be seen form Fig. 7.9 that the smallest number of movements results from the EVF strategy.

Let the energy that a sensor spends to travel 1m (with no stop) be 8.268J [130], [135]. Consider two cases, where the energy required to stop a sensor and then overcome its static friction after a complete stop is equal to the energy that the sensor spends to travel 1m (first case) and 4m (second case) [64], [65]. Tables 7.1 and 7.2 give a summary of the energy consumption results for these two cases, and

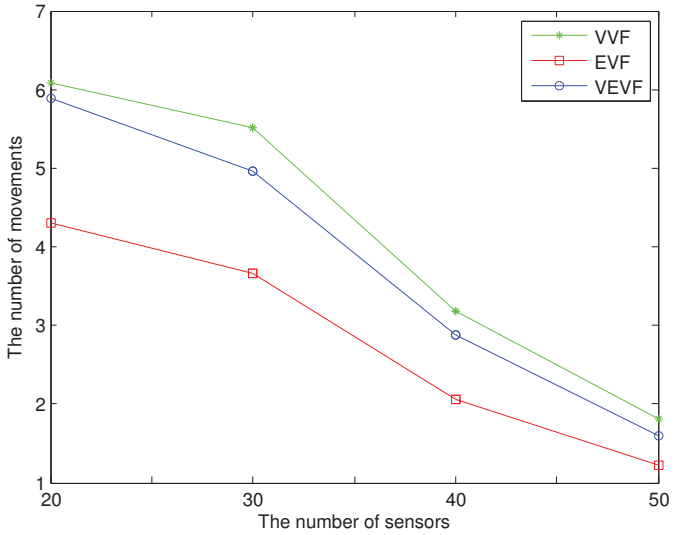


Figure 7.9: The number of movements required for different number of sensors in Example 1, using the proposed algorithms.

demonstrate that in both scenarios the EVF algorithm is more energy-efficient than the other two algorithms.

Table 7.1: The energy consumption in Joule for different number of sensors using the proposed algorithms in the first case of Example 1.

	$n = 20$	$n = 30$	$n = 40$	$n = 50$
VVF	106.8229 J	80.3771 J	48.5951 J	29.5852 J
EVF	91.4222 J	69.6011 J	43.1367 J	28.1033 J
VEVF	111.2314 J	79.8485 J	48.6748 J	29.3468 J

Example 2: Consider 20 identical sensors with the sensing range of 6m and the communication range of 20m, which are randomly deployed in a 50m by 50m field. This example aims to compare the performance of the VEVF algorithm with some existing techniques, namely VEC, VOR, Minimax [64], Maxmin-vertex, Minmax-edge, Maxmin-edge, and VEDGE [98]. The coverage factor of the sensors in each round of different algorithms is depicted in Fig. 7.10, where it can be observed that the VEVF algorithm outperforms the other strategies as far as sensor coverage is concerned. In addition, the complexity of finding the new destination of each sensor

Table 7.2: The energy consumption in Joule for different number of sensors using the proposed algorithms in the second case of Example 1.

	$n = 20$	$n = 30$	$n = 40$	$n = 50$
VVF	257.8297 J	216.9892 J	127.5152 J	74.5747 J
EVF	198.1787 J	160.3506 J	94.1152 J	58.3443 J
VEVF	257.4014 J	202.7688 J	120.1662 J	68.8397 J

in the VEVF strategy is of order $O(m)$, while this complexity in the Minimax, Maxmin-vertex, Maxmin-edge, VEDGE and Minmax-edge techniques is of order $O(m^4)$, where m is the number of vertices of the Voronoi polygon.

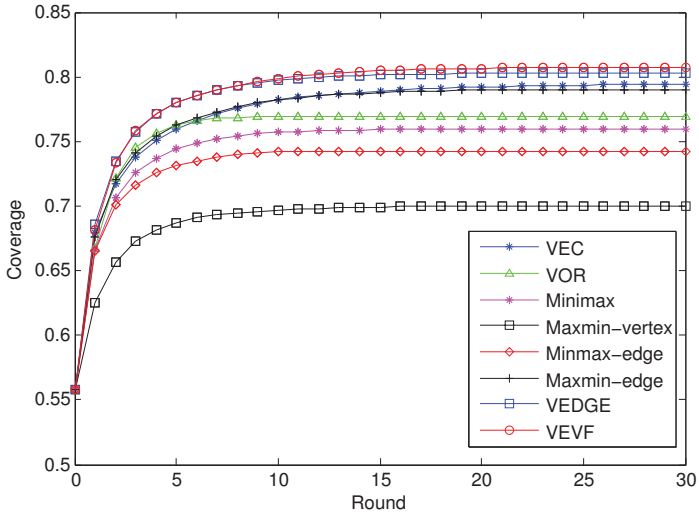


Figure 7.10: Coverage factor per round for 20 sensors in Example 2.

Example 3: In this example, 27 sensors are randomly deployed in a 50m by 50m field: 15 sensors with a sensing radius of 6m, 6 with a sensing radius of 5m, 3 with a sensing radius of 7m, and 3 with a sensing radius of 9m. Let the communication radius of each sensor be 10/3 times its sensing radius. The CPVF algorithm is now compared with six other techniques reported in the literature, namely WVB, FPB [99], Minmax-vertex, Maxmin-vertex [101], Minmax-curve and Maxmin-curve [100]. The coverage factor in each round is depicted in Fig. 7.11 for different strategies. This figure clearly shows that the coverage factor obtained by

using the CPVF algorithm is better than that obtained by using any other algorithm cited above. As in the previous example, the complexity of finding the new destination of each sensor in the CPVF strategy is of order $O(m)$, while this complexity in the Maxmin-vertex, Minmax-vertex, Maxmin-curve and Minmax-curve algorithms is of order $O(m^4)$.

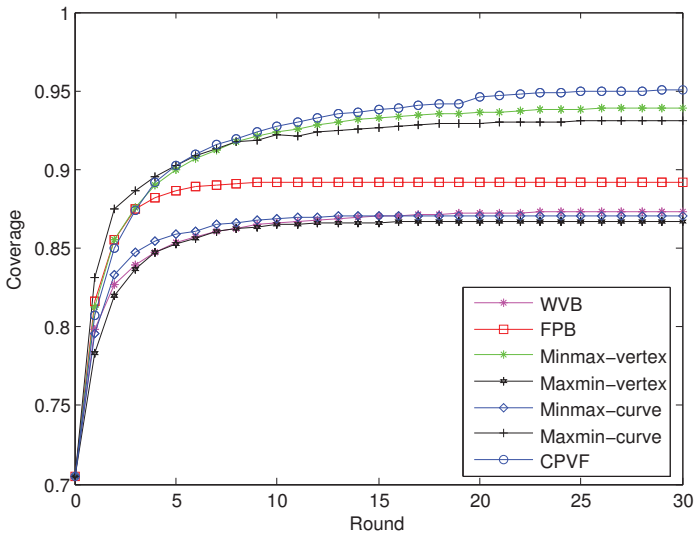


Figure 7.11: Coverage factor in each round of different algorithms in Example 3.

Chapter 8

Self-deployment Algorithms for Coverage Improvement in a Network of Nonidentical Mobile Sensors with Limited Communication Ranges

Efficient deployment algorithms are proposed in this chapter to increase coverage in a network of nonidentical mobile sensors with limited communication radii. The proposed algorithms calculate and update the positions of the sensors iteratively in such a way that the overall network coverage increases. The notion of multiplicatively-weighted Voronoi (MW-Voronoi) diagram is extended to the case of sensors with limited communication capability to introduce the limited communication MW-Voronoi (LCMW-Voronoi) diagram. This diagram is used to discover coverage holes in the network. The sensors move in such a way that the coverage holes in the network are eliminated as much as possible. Simulations demonstrate the efficacy

of the proposed distributed deployment schemes.

The outline of this chapter is as follows. The problem description is presented in Section 8.1. Then in Section 8.2 the notion of LCMW-Voronoi diagram is defined and characterized mathematically, which is used in Section 8.3 to present the deployment algorithms as the main result of the chapter. Simulations are provided in Section 8.4 to show the efficacy of the proposed algorithms.

8.1 Problem Statement

Consider a group of n mobile sensors, randomly distributed in a sensing field. The communication capabilities of the sensors is assumed to be limited and not necessarily the same for all sensors. It is also assumed that the sensors are not necessarily identical in terms of sensing capabilities. Let the sensing range and communication range of the i-th sensor be circles of radius r_{si} and r_{ci}, respectively, centered at the position of that sensor. It is desired to move the sensors and place them in proper locations using a distributed deployment strategy such that the covered area is improved as much as possible. In other words, the objective is to increase the coverage area using limited information exchange between sensors.

Several algorithms have been provided in the literature for improving coverage in a network of mobile sensors. These algorithms often use Voronoi diagram (for the case of identical sensors) or MW-Voronoi diagram (for the case of nonidentical sensors) for partitioning the target field. In practical applications, the communication range of sensors is bounded (and not necessarily the same for all sensors). This is a limiting factor, potentially preventing the sensors from communicating with their neighbors, and can cause a sensor to generate a Voronoi region whose boundaries are different from the exact ones. As a result, such Voronoi regions do not necessarily partition the field, in the sense that some of them overlap with

each other. Such MW-Voronoi regions (or Voronoi polygons) can have a negative impact on the ability to detect coverage holes. On the other hand, the overlap of the Voronoi regions can lead to sensor collisions. As an example, consider two sensors with the sensing range of 6m and 7m deployed in a 50m $\times$ 50m field (see Fig.8.1). It is assumed that the communication ranges of the sensors are 20m and 25m, respectively, and they move according to the Minmax-vertex algorithm [101] to increase network coverage. Since these sensors cannot communicate with each other, they fail to partition the field correctly and each one considers the entire field as its MW-Voronoi region. Then each sensor moves to the Minmax-vertex centroid of the field (point M). As a result, not only does the network coverage decrease, but it also collides with the other sensor. The limited communication multiplicatively weighted Voronoi (LCMW-Voronoi) diagram described in the next section is used to remedy this shortcoming.

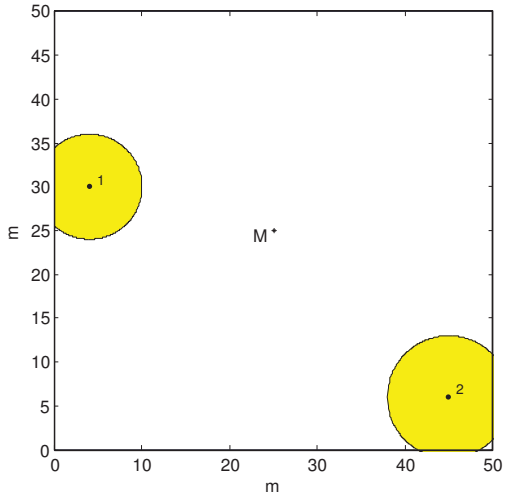


Figure 8.1: An example of two sensors with insufficient communication power which fail to derive the correct MW-Voronoi regions and eventually collide.

8.2 Limited Communication Multiplicatively Weighted Voronoi Diagram

Let $\mathbf{S}$ be a set of n distinct nodes representing n mobile sensors (p_1, r_{c1}, r_{s1}),(p_2, r_{c2}, r_{s2}), ...,(p_n, r_{cn}, r_{sn}) distributed in the 2D field Q, where $r_{si} > 0$ is the sensing radius of the i-th sensor, p_i is its position and $r_{ci} > 0$ is its communication range, for any $i \in \mathbf{n} := \{1, 2, \ldots, n\}$. Let also $\mathcal{G}_i$ be the set of all sensors whose communication ranges cover the i-th sensor, and hence can send the required information about their positions and sensing radii to the i-th sensor, i.e. $(p_j, r_{cj}, r_{sj}) \in \mathcal{G}_i$ if and only if $\|p_j - p_i\| \leq r_{cj}$. Denote by $\mathbf{Indx}(\mathcal{G}_i)$ the indices of the sensors in $\mathcal{G}_i$, and let the minimum communication range of the sensors be denoted by r_{min} (i.e., $r_{min} = min_{j\in\mathbf{n}}\{r_{cj}\}$). The point $q \in Q$ is called a "distant" point if $d(q, p_i) > \frac{r_{min}}{2}$ for all $i \in \mathbf{n}$, and is called a "close" point otherwise. The set of all distant points is called "distant" region and will hereafter be denoted by $\Psi_{\mathbf{S}}$.

Definition 8.1. *The weighted distance of a point q from a node (p_i, r_{ci}, r_{si}), $i \in \mathbf{n}$, is defined as:*

$$d_w(q, p_i) = \frac{d(q, p_i)}{r_{si}}$$

where $d(q, p_i)$ denotes the Euclidean distance between the point q and the node p_i.

The *Limited Communication Multiplicatively Weighted Voronoi* (LCMW-Voronoi) diagram of $\mathbf{S}$ is the set of the regions $\Pi(\mathbf{S}) = \{\Pi_1, \Pi_2, \ldots, \Pi_n\}$, where each region is characterized as:

$$\Pi_i = \left\{ q \in Q \mid d(q, p_i) \leq \frac{r_{min}}{2},\ d_w(q, p_i) < d_w(q, p_j), \forall j \in \mathbf{Indx}(\mathcal{G}_i) \right\} \tag{8.1}$$

where $d_w(q, p_i)$ is the weighted distance of the point q from node (p_i, r_{ci}, r_{si}).

Assumption 8.1. *It is assumed that the minimum communication radius, r_{min}, is known by each sensor a priori.*

To construct the i-th LCMW-Voronoi region, first the Apollonian circles $\Omega_{p_i p_j, \frac{r_{si}}{r_{sj}}}$ are found for all $j \in \mathbf{Indx}(\mathcal{G}_i)$. The intersection of the smallest region (created by the above circles) containing the i-th sensor and a circle with radius $\frac{r_{min}}{2}$ centered at p_i is, in fact, the i-th LCMW-Voronoi region. An example of the LCMW-Voronoi region is shown in Fig. 8.2. In this figure, sensor located at p_1 receives the information of the four sensors located at $p_2, \ldots, p_5$, and construct its region (shaded area). An example of the LCMW-Voronoi diagram for a group of 9 sensors is sketched in Fig. 8.3. In this figure, the distant region is denoted by brown color.

Note that in the case of equal sensing radii (i.e. $r_{si} = r_{sj}, \forall i, j \in \mathbf{n}$), the Apollonian circles are in fact the perpendicular bisectors of the segments connecting different pairs of sensors' positions. In this case, the obtained diagram is referred to as the *Limited Communication Voronoi* (LCG-Voronoi) diagram.

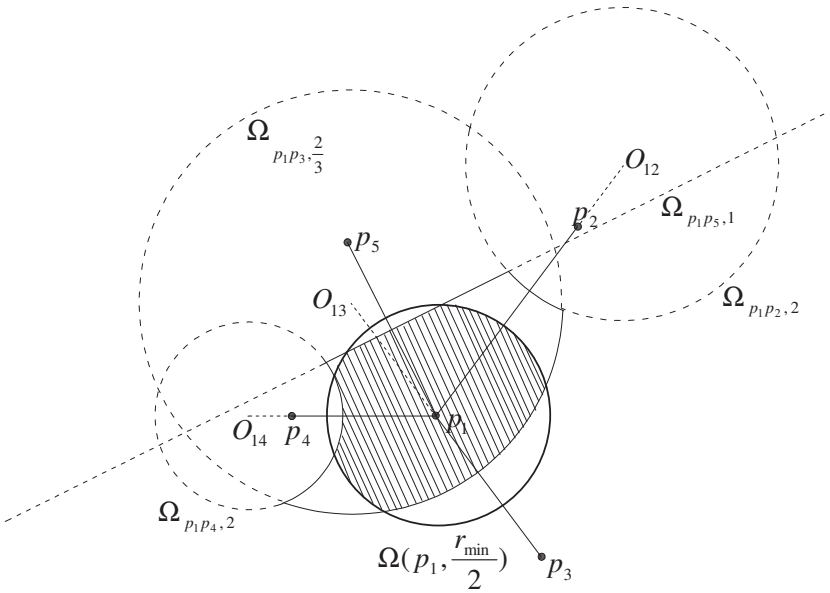


Figure 8.2: The LCMW-Voronoi region for a sensor p_1 with four neighboring sensors $p_2, \ldots, p_5$.

Theorem 8.1. *Let* $\mathbf{S}$ *be a set of* n *mobile sensors distributed in* $Q \subset \mathbb{R}^2$*, and* $\Pi_1, \Pi_2, \ldots, \Pi_n$ *be the regions of the corresponding LCMW-Voronoi diagram. Then,* $\Pi_i \cap \Pi_j = \emptyset$.

Proof. Suppose there are two regions Π_i and Π_j such that $\Pi_i \cap \Pi_j \neq \emptyset$. This means that:

$$\exists q \in Q : q \in \Pi_i, \quad q \in \Pi_j \tag{8.2}$$

It can be concluded from (8.2) and (8.1) that:

$$d(q, p_i) \leq \frac{r_{min}}{2}, \quad d(q, p_j) \leq \frac{r_{min}}{2} \tag{8.3}$$

Using (8.3) and the triangle inequality, one arrives at:

$$d(p_i, p_j) \leq d(q, p_i) + d(q, p_j) \leq r_{min} \tag{8.4}$$

From (8.4) and on noting that $r_{min} \leq min\{r_{ci}, r_{cj}\}$, one can conclude that:

$$i \in \mathbf{Indx}(\mathcal{G}_j), \quad j \in \mathbf{Indx}(\mathcal{G}_i) \tag{8.5}$$

Since $q \in \Pi_i$ and $j \in \mathbf{Indx}(\mathcal{G}_i)$, hence (8.1) yields:

$$d_w(q, p_i) < d_w(q, p_j) \tag{8.6}$$

On the other hand, since $q \in \Pi_j$ and $i \in \mathbf{Indx}(\mathcal{G}_j)$, it is deduced from (8.1) that:

$$d_w(q, p_j) < d_w(q, p_i) \tag{8.7}$$

which contradicts the inequality (8.6), and hence invalidates the initial assumption $\Pi_i \cap \Pi_j \neq \emptyset$. ■

Unlike the multiplicatively weighted Voronoi (MW-Voronoi) diagram, the LCMW-Voronoi diagram does not partition the field. In fact, due to the limited communication capability of sensors, it is possible that some of the points in Q do not belong to any region. The LCMW-Voronoi diagram is the main tool used for developing the sensor deployment strategies in this chapter.

Remark 8.1. *From the characterization of the LCMW-Voronoi regions, it can be shown that if a sensor cannot detect a point in its corresponding region, no other sensor can detect it either.*

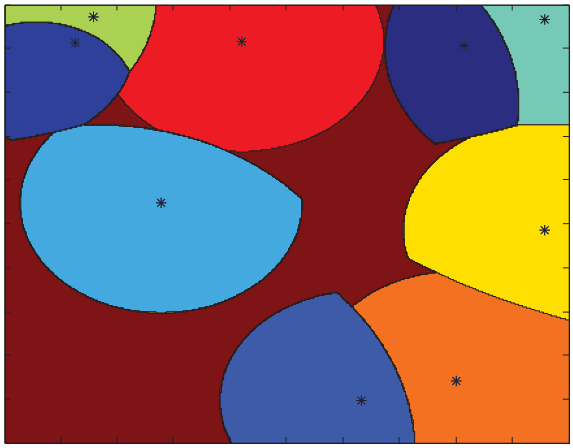


Figure 8.3: An example of the LCMW-Voronoi diagram for a group of 9 sensors with different sensing and communication ranges.

Assumption 8.2. *Since the communication range of a mobile sensor is typically much larger than its sensing range [130], in this chapter it is assumed that the sensing range of the sensors is less than or equal to* $\frac{r_{min}}{2}$.

8.3 Deployment Protocols

In this section, two different deployment protocols are developed for a network of nonidentical sensors with limited communication capabilities in an obstacle-free environment. The proposed techniques are iterative, where in each iteration every sensor S_i, $i \in \mathbf{n}$, first broadcasts its sensing radius r_{si} and position P_i to other sensors in its communication range, and subsequently constructs its own LCMW-Voronoi region based on the similar information it receives from other sensors. Then, it identifies coverage holes in its region, based on which it finds a candidate point as its new location using one of the proposed algorithms. Once the new location $\acute{P}_i$ is calculated, the coverage area w.r.t. this point (i.e. $\beta_{\Pi_i}^{\acute{P}_i}$) is evaluated and compared to the current coverage area (i.e. $\beta_{\Pi_i}^{P_i}$). The sensor moves to the new location only if the resultant coverage area is greater than the present value, i.e. $\beta_{\Pi_i}^{\acute{P}_i} > \beta_{\Pi_i}^{P_i}$; otherwise, it remains in its current location in this iteration. A proper threshold is

considered for coverage increase in each iteration to terminate the algorithm if no sensor can improve its coverage area by this threshold.

The following theorem shows that using the proposed sensor deployment algorithms network coverage increases.

Theorem 8.2. *Let the position of the i-th sensor and its LCMW-Voronoi region be denoted by P_i and Π_i, respectively. Let also the distant region be denoted by $\Psi_{\mathbf{P}}$. Assume the i-th sensor moves to the new location $\acute{P}_i$ with the corresponding LCMW-Voronoi region $\acute{\Pi}_i$, such that $\acute{P}_i \neq P_i$ for all $i \in \mathbf{k}$, where $\mathbf{k}$ is a non-empty subset of $\mathbf{n}$. If the i-th coverage area w.r.t. $\acute{P}_i$ in the previously constructed LCMW-Voronoi region Π_i is greater than the previous i-th local coverage area (i.e., $\beta_{\Pi_i}^{\acute{P}_i} > \beta_{\Pi_i}^{P_i}$) for all $i \in \mathbf{k}$, then the total coverage in the network increases.*

Proof. Define $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ and $\acute{\mathbf{P}} = \{\acute{P}_1, \acute{P}_2, \ldots, \acute{P}_n\}$, and denote the total uncovered area (coverage hole) of the field when the sensors are located at the points in $\mathbf{P}$ and $\acute{\mathbf{P}}$ by θ and $\acute{\theta}$, respectively. Since it is assumed that the sensing range of every sensor is less than or equal to $\frac{r_{min}}{2}$, no point of $\Psi_{\mathbf{P}}$ can be covered by the sensors located in $\mathbf{P}$, and consequently one can deduce from the characterization of the LCMW-Voronoi diagram that:

$$\theta = \Psi_{\mathbf{P}} + \sum_{i=1}^{n} \theta_{\Pi_i}^{P_i} \tag{8.8}$$

It is straightforward to show that by increasing the coverage area in $\Pi_i,\ i \in \mathbf{k}$, the corresponding coverage hole will become smaller. Since it is assumed that the i-th coverage area w.r.t. $\acute{P}_i$ is greater than the i-th local coverage area for any $i \in \mathbf{k}$, one can conclude that:

$$\theta_{\Pi_i}^{\acute{P}_i} < \theta_{\Pi_i}^{P_i}, \quad \forall i \in \mathbf{k} \tag{8.9}$$

On the other hand, it is possible that some of the points in $\theta_{\Pi_i}^{\acute{P}_i}$ and $\Psi_{\mathbf{P}}$ are also covered by other mobile sensors when they move to the points in $\acute{\mathbf{P}}$. Hence:

$$\acute{\theta} \leq \Psi_{\mathbf{P}} + \sum_{i=1}^{n} \theta_{\Pi_i}^{\acute{P}_i} \tag{8.10}$$

From the last two relations and on noting that for any $i \in \mathbf{n}\backslash\mathbf{k}$ by definition $\theta_{\Pi_i}^{\acute{P}_i} = \theta_{\Pi_i}^{P_i}$, one arrives at the following inequality:

$$\acute{\theta} < \Psi_{\mathbf{P}} + \sum_{i=1}^{n} \theta_{\Pi_i}^{P_i} \tag{8.11}$$

Now, (8.8) and (8.11) yield:

$$\acute{\theta} < \theta \tag{8.12}$$

This means that the total coverage area increases using the proposed deployment algorithms. ■

The procedure introduced in this section so far will be used in the next two subsections to develop two sensor deployment algorithms.

8.3.1 Limited Communication Farthest Point (LCFP) Strategy

In this algorithm, when a sensor S_i detects a coverage hole in its LCMW-Voronoi region, it finds the farthest point $X_{i,far}$ in that region, and moves toward it until this point is covered. Note that the farthest point in the i-th LCMW-Voronoi region is located on the boundary of that region. It is straightforward to verify from the characterization of the LCMW-Voronoi regions that some boundary curves of the i-th region might be located on a circle of radius $\frac{r_{min}}{2}$ centered at the position of the sensor, and consequently all points on these boundary curves have the maximum distance from S_i. In such cases, the midpoint of these curves are considered as the farthest point. Also, when there are more than one farthest point, that one that leads to a larger increase in the coverage area within the LCMW-Voronoi region would be selected as the destination point for the corresponding sensor. Fig. 8.4 shows an example of the LCMW-Voronoi region constructed by the sensor S_i. The segments AE and BC are located between S_i and two of its neighbors with the same sensing radius r_i. The arcs AB and CD are constructed by the Apollonian circles

associated with S_i and two of its neighboring sensors. The neighbor corresponding to AB has a smaller sensing radius compared to S_i, while the one corresponding to CD has a larger radius than r_{si}, and finally arc DE is formed by a circle with radius $\frac{r_{min}}{2}$ centered at the position of S_i. As the figure illustrates, the LCFP algorithm finds the farthest point from S_i (i.e. $X_{i,far}$) as a candidate for the sensor to move in that direction. Since the coverage area of the sensor within its LCMW-Voronoi region will increase by moving to P_i' (which is between the current position of the sensor and the farthest point, and the sensor can cover the farthest point from there), the sensor moves there.

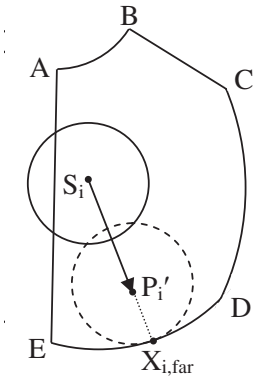
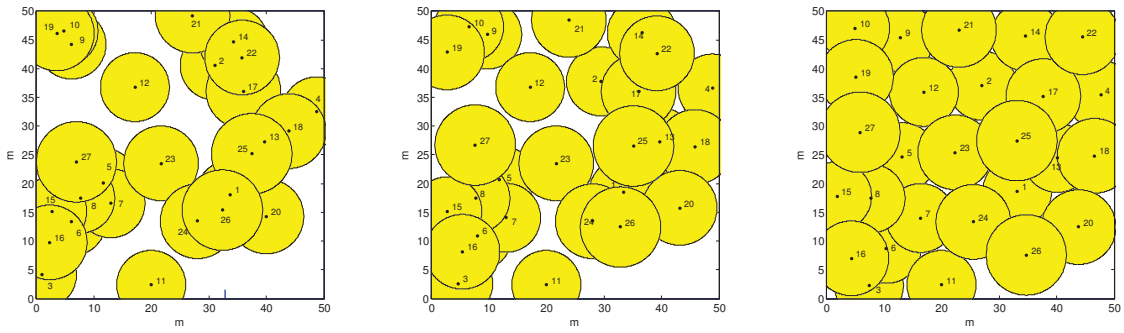


Figure 8.4: An LCMW-Voronoi region and a candidate point obtained by using the LCFP method.

Fig. 8.5 shows an operational example of the LCFP Algorithm. In this example, 27 sensors are randomly deployed in a 50m $\times$ 50m flat space: 15 with a sensing radius of 6m, 9 with a sensing radius of 6.5m, and 3 with a sensing radius of 7m. Moreover, the communication range of each sensor is assumed to be 10/3 times its sensing range. In this figure, three snapshots are provided, and in each one the coverage area of the sensors (filled circles) is depicted. The initial coverage is 66.37%, but after the first round it increases to 79.82%, and the final coverage is 94.53%.



(a) (b) (c)

Figure 8.5: Snapshots of the sensor locations using the LCFP algorithm. (a) Initial coverage; (b) coverage after the first round, and (c) final coverage.

8.3.2 Limited Communication Minmax Point (LCMP) Strategy

Although simulations confirm the efficiency of the LCFP algorithm, there are certain network settings and node configurations for which it is not as effective. Fig. 8.6 shows an example of such a case. One can easily observe that the candidate point obtained by the LCFP algorithm for the next location in this setting is not a good position for the sensor because if it moves there the coverage area w.r.t. this candidate point decreases. Thus, under the LCFP algorithm the sensor remain in its current location while its coverage area within the corresponding LCMW-Voronoi region could increase if it moved in a different direction.

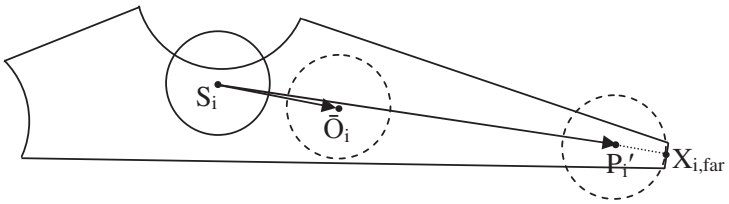


Figure 8.6: An example of an LCMW-Voronoi region for which the LCFP method performs poorly because of a narrow area, while the LCMP technique provides a better candidate location for the sensor.

The limited communication minmax point (LCMP) strategy is introduced in

the sequel to address this shortcoming of the LCFP algorithm. The main idea behind the LCMP strategy is that in an optimal sensor configuration, no sensor should be too far from any point in its corresponding LCMW-Voronoi region. The LCMP strategy finds the location whose distance from the farthest point of the region is minimum and considers it as the candidate location for the sensor in the next step. This point is called the *LCMP centroid*, and is denoted by $\bar{O}_i$ for the i-th region, $i \in \mathbf{n}$. It is clear that the candidate point $\bar{O}_i$ in Fig. 8.6 yields better coverage compared to the one obtained by using the LCFP technique.

Theorem 8.3. *The LCFP and LCMP algorithms are convergent.*

Proof. The proof is similar to that of Theorem 5.4, and is omitted here. ■

The following theorem provides an upper-bound on the number of rounds required to run the algorithm, as a function of ϵ.

Theorem 8.4. *Consider a set* $\mathbf{S}$ *of* n *mobile sensors randomly deployed in a 2D field. Using any of the proposed algorithms with the coverage improvement threshold* ϵ*, the number of required rounds to run the algorithm is less than or equal to* $\frac{A_{total}}{\epsilon}$*, where* A_{total} *is the overall area of the field.*

Proof. Let the number of rounds required to run the algorithm in order to meet the termination condition be denoted by ζ_f. Let also the total uncovered area of the field in the k-th round be represented by $\theta(k)$, and note that $\beta(k) = A_{total} - \theta(k)$. Denote the position of the sensors in the k-th round by $\mathbf{P}(k) = \{P_1(k), P_2(k), \ldots, P_n(k)\}$, and let the LCMW-Voronoi region of the i-th sensor be represented by $\Pi_i(k)$, $i \in \mathbf{n}$. Denote also the distant region in the k-th round by $\Psi_{\mathbf{P}}(k)$. From the properties of the LCMW-Voronoi diagram, one can conclude that:

$$\theta(k) = \Psi_{\mathbf{P}}(k) + \sum_{i=1}^{n} \theta_{\Pi_i(k)}^{P_i(k)}, \ \forall 1 \leq k \leq \zeta_f \tag{8.13}$$

Define the *moving set of the k-th round* as the largest subset of $\mathbf{S}$ that moves in the k-th round, and denote the indices of the sensors in this set by $\mathbf{Ix}(k)$. Note that at least

one sensor moves in the k-th round, and hence $\mathbf{Ix}(k) \neq \emptyset$, $\forall k \in \{1, 2, \ldots, \zeta_f\}$. By assumption, the i-th sensor, $i \in \mathbf{Ix}(k)$, moves in the k-th round if $\beta_{\Pi_i(k)}^{P_i(k+1)} \geq \beta_{\Pi_i(k)}^{P_i(k)} + \epsilon$. This means that:

$$\theta_{\Pi_i(k)}^{P_i(k+1)} \leq \theta_{\Pi_i(k)}^{P_i(k)} - \epsilon, \quad \forall i \in \mathbf{Ix}(k) \tag{8.14}$$

Note also that some of the points in $\Psi_{\mathbf{P}}(k)$ might be covered by some sensors located at $\mathbf{P}(k+1)$. In addition, some of the points in $\theta_{\Pi_i(k)}^{P_i(k+1)}$ might also be covered by another sensor located at $P_j(k+1)$, for some $j \in \mathbf{n} \backslash \{i\}$. Hence:

$$\theta(k+1) \leq \Psi_{\mathbf{P}}(k) + \sum_{i=1}^{n} \theta_{\Pi_i(k)}^{P_i(k+1)} \tag{8.15}$$

From the last two relations and on noting that for any $i \in \mathbf{n} \backslash \mathbf{Ix}(k)$ the i-th sensor does not move (which implies $\theta_{\Pi_i(k)}^{P_i(k+1)} = \theta_{\Pi_i(k)}^{P_i(k)}$), one arrives at:

$$\theta(k+1) \leq \Psi_{\mathbf{P}} + \sum_{i=1}^{n} \theta_{\Pi_i(k)}^{P_i(k)} - |\mathbf{Ix}(k)|\, \epsilon \tag{8.16}$$

It is now concluded from (8.13) and (8.16) that:

$$\theta(k+1) \leq \theta(k) - |\mathbf{Ix}(k)|\, \epsilon \leq \theta(k) - \epsilon \tag{8.17}$$

or equivalently:

$$\beta(k+1) \geq \beta(k) + |\mathbf{Ix}(k)|\, \epsilon \geq \beta(k) + \epsilon \tag{8.18}$$

which implies that using the underlying sensor relocation scheme, in each round the total covered area increases by at least ϵ. Therefore, the total amount of increased coverage from the first round to the termination round is greater than or equal to $\zeta_f \epsilon$. Since the total covered area is always less than or equal to A_{total}, hence $A_{total} \geq \zeta_f \epsilon$ or equivalently $\frac{A_{total}}{\epsilon} \geq \zeta_f$. ■

Remark 8.2. *One of the important properties of the LCMW-Voronoi diagram is that its regions are mutually disjoint (see Theorem 8.1) and also there is exactly one sensor in each region. Since in the proposed algorithms the new location for each sensor is inside the corresponding LCMW-Voronoi region and each sensor moves within its region, hence the sensors would not collide.*

In the next section, the performance of the proposed algorithms in terms of coverage area, energy consumption of the sensors, and rate of convergence are investigated.

8.4 Simulation Results

The results presented in this section are the average values obtained from 20 different random initial sensor deployments. In each simulation, the algorithm is terminated when none of the sensors' coverage in its corresponding LCMW-Voronoi region is improved by more than 0.1m^2 in the next move.

The two algorithms proposed in Section 8.3 are applied to a flat space of size $50\text{m} \times 50\text{m}$. Assume first that 36 mobile sensors are randomly placed in the field: 20 with a sensing radius of 6m, 12 with a sensing radius of 6.5m, and 4 with a sensing radius of 7m. The communication range of each sensor is assumed to be 10/3 times its sensing range; e.g., a sensor with a sensing radius of 6m has a communication radius of 20m. Define the *coverage factor* as the ratio of the covered area to the total area in the field. Fig. 8.7 shows the coverage factor in each round for both algorithms. As it can be seen in this figure, both algorithms provide good coverage but the LCMP technique performs better than the LCFP algorithm.

It is desired now to investigate the effect of the number of sensors on the performance of the algorithms. To this end, consider three more setups: n=18, 27, and 45, in addition to n=36 discussed above. Let the changes in the number of identical sensors in the new setups be proportional to the changes in the total number of sensors (e.g., for n=27 there are 15 sensors with sensing radius 6m, 9 with sensing radius 6.5m, and 3 with sensing radius 7m). The final coverage results are shown in Fig. 8.8. Note that both algorithms yield good results in all settings, and in particular the LCMP strategy performs better.

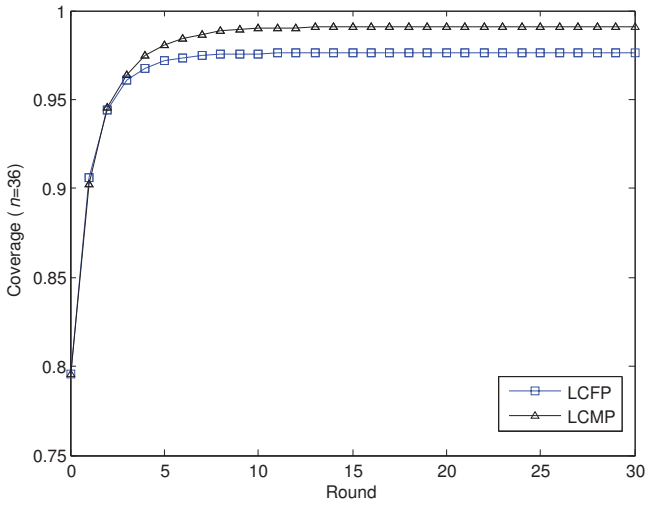


Figure 8.7: Network coverage per round for 36 sensors.

One of the important factors that must be taken into account in the performance evaluation of different deployment techniques is the time it takes for the algorithm to stop. Suppose that the deployment time of the sensors in each round of both techniques is the same. Then the number of rounds for each algorithm to reach a predetermined termination criterion is a good measure of the deployment speed. Fig. 8.9 shows that under both algorithms the number of rounds required to meet a certain termination condition increases by increasing the number of sensors from 18 to 27, and then starts to decrease by adding more sensors. This is primarily due to the fact that when the number of sensors is small, because of the relatively large size of the LCMW-Voronoi regions, there is a high chance for the sensors' coverage circles to be enclosed inside their corresponding regions. In this case, further movement of each sensor in its region would not increase the coverage level. On the other hand, when the number of sensors is relatively large, then the LCMW-Voronoi regions are small, and hence there is a high chance that the coverage circle of each sensor encloses the corresponding LCMW-Voronoi region. This in turn implies that the termination condition is satisfied in smaller number of rounds. The figure shows that the convergence rate of the LCFP algorithm is faster than that of the LCMP

strategy. Thus, the LCFP algorithm is a better candidate for field coverage as far as the deployment speed is concerned.

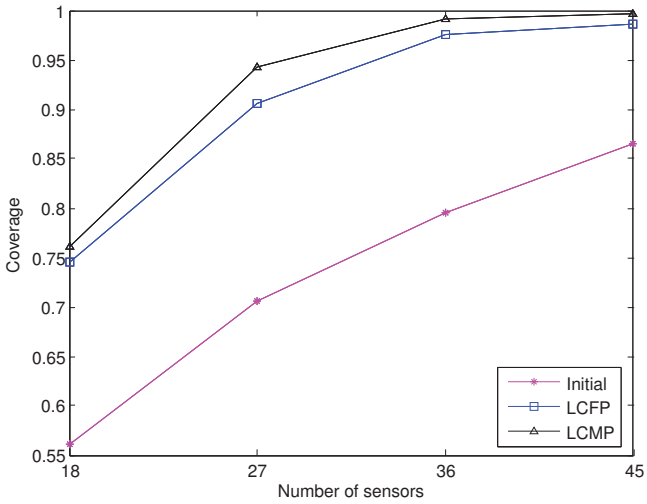


Figure 8.8: The coverage factor achieved for different number of sensors under the proposed algorithms.

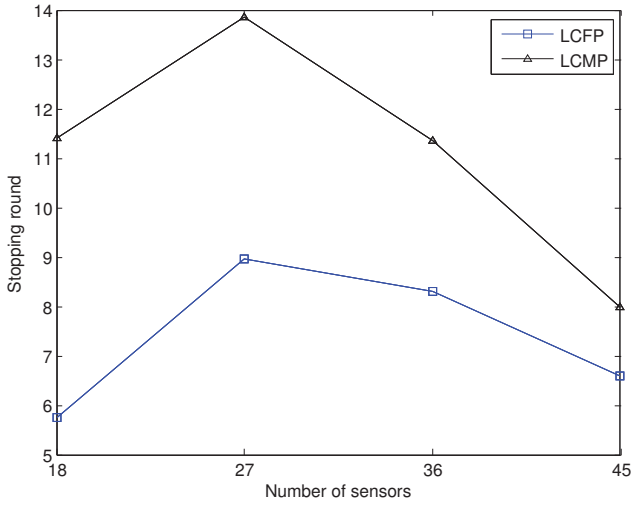


Figure 8.9: The number of rounds required to reach the termination condition for different number of sensors using the proposed algorithms.

Energy-efficiency is another important factor which needs to be taken into consideration for comparing the performance of different deployment algorithms in mobile senors networks. The energy consumption of a mobile sensor is mainly due to movement. More precisely, the traveling distance of a sensor and also the number

of times it stops (the latter is due to static friction) are the dominant sources of energy consumption. Fig. 8.10 shows the average distance traveled by each sensor for different number of sensors in the network. It can be observed from this figure that for a large number of sensors, the average traveled distance is small in general. In fact, as the number of sensors increases, the distance between each sensor and its candidate location in the corresponding LCMW-Voronoi region decreases. This in turn decreases the average traveling distance, which leads to a decrease in energy consumption. Moreover, it can be seen from Fig. 8.10 that the LCFP algorithm is more efficient than the LCMP strategy in terms of traveling distance. The number of movements versus the number of sensors is depicted in Fig. 8.11, which shows in both algorithms the number of movements increases with the number of sensors up to a certain value, and decreases after that. Again, this can be justified based on the size of sensing disk of each sensor and that of the corresponding LCMW-Voronoi region. According to Figs. 8.10 and 8.11, the LCFP algorithm outperforms the LCMP strategy as far as energy consumption is concerned.

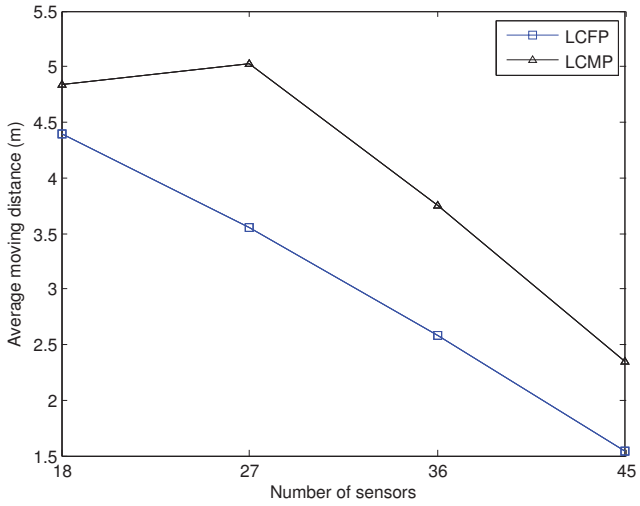


Figure 8.10: The average distance each sensor travels for different number of sensors, using the proposed algorithms.

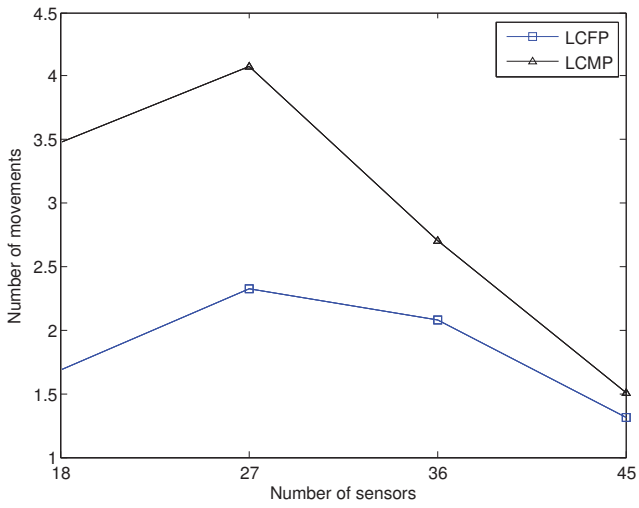


Figure 8.11: The number of movements required for different number of sensors, using the proposed algorithms.

Chapter 9

Self-deployment Algorithms for Coverage Improvement in Mobile Sensor Networks in Presence of Obstacles

In this chapter, efficient algorithms for mobile sensor deployment are proposed to improve the coverage area in target fields containing obstacles. The proposed algorithms iteratively calculate and update the position of the sensors in order to improve the overall achievable coverage by the network. The *visibility-aware multiplicatively weighted Voronoi* (VMW-Voronoi) diagram is introduced and used to discover coverage holes in networks that have sensors with different sensing capabilities. The sensors, then, reduce the size of the coverage holes in the target field. The relocation strategy also considers possible existing obstacles on the field. Simulation results are provided to demonstrate the effectiveness of the proposed distributed deployment schemes.

The rest of this chapter is organized as follows. Section 9.1 introduces visibility-aware multiplicatively weighted Voronoi diagram as an extension of the conventional Voronoi diagram. The proposed algorithms for sensor relocation are introduced in Section 9.2. Finally, simulation results that demonstrate the effectiveness of our approach are provided in Section 9.3.

9.1 Visibility-aware Multiplicatively Weighted Voronoi Diagram

Let $\mathbf{F} \subset \mathbb{R}^{\mathbf{2}}$ represent a 2D target field. Consider $\mathbf{S} = (S_1, w_1), (S_2, w_2), \ldots, (S_n, w_n)$ to be a set of n distinct weighted nodes within the field $\mathbf{F}$. $w_i > 0$ is the weighting factor associated with the node S_i, for any $i \in \mathbf{n} := \{1, 2, \ldots, n\}$. The *visible set* of an arbitrary point $Q \in \mathbf{F}$ is defined as the largest subset of $\mathbf{S}$ with a non-obstructed line of sight view from all of its elements to the point Q. Let $\mathbf{Indx}_Q$ represent the indices of the nodes in this subset. In presence of obstacles in the field and depending on the location of point Q, the set $\mathbf{Indx}_Q$ may have between 0 to n elements. In particular, Q is called an *invisible point* if $\mathbf{Indx}_Q$ is an empty set, otherwise it is called a *visible point*. The set of all invisible points in the field $\mathbf{F}$ is called the invisible region and will be denoted by $\Theta_{\mathbf{S}}$. The invisible region is highly dependent on the positions of the nodes S_i and the obstacles on the field.

Definition 9.1. *The weighted distance of a point Q from a node (S_i, w_i), $i \in \mathbf{n}$ is defined as:*

$$d_w(Q, S_i) = \frac{d(Q, S_i)}{w_i}$$

where $d(Q, S_i)$ denotes the Euclidean distance between the point Q and the node S_i in the 2D field $\mathbf{F}$.

It is desired now to partition the visible area of the field ($\mathbf{F} \backslash \Theta_{\mathbf{S}}$) into n regions

such that:

- Each region contains only one node, and
- the nearest node, in the sense of weighted distance, to any point inside a region is the node assigned to that region.

The mathematical characterization of the invisible region and each visible region is given by:

$$\Theta_{\mathbf{S}} = \left\{ Q \in \mathbb{R}^2 \mid \mathbf{Indx}_Q = \emptyset \right\} \tag{9.1}$$

$$\Pi_i = \left\{ Q \in \mathbb{R}^2 \mid i \in \mathbf{Indx}_Q,\ d_w(Q, S_i) \leq d_w(Q, S_j), \forall j \in \mathbf{Indx}_Q - \{i\} \right\} \tag{9.2}$$

The diagram obtained by partitioning the field $\mathbf{F}$ into the invisible region and the above-mentioned n regions is called the *visibility-aware multiplicatively weighted Voronoi* (VMW-Voronoi) diagram. According to (9.2), any point Q in the i-th VMW-Voronoi region Π_i has the following property:

$$\frac{d(Q, S_i)}{d(Q, S_j)} \leq \frac{w_i}{w_j}, \quad \forall i, j \in \mathbf{Indx}_Q,\ i \neq j \tag{9.3}$$

The VMW-Voronoi diagram is the main tool for developing the sensor deployment strategy in this chapter. Each sensor is characterized by a sensing area which is a circle whose size is not necessarily the same for different sensors. Consider the position of each sensor in the field as a node with a weight equal to the sensor's sensing radius, and sketch the VMW-Voronoi region for each sensor; the resultant diagram, together with the invisible region, covers the entire field $\mathbf{F}$.

From the characterization of the VMW-Voronoi regions provided in (9.2), it is straightforward to show that if a sensor cannot detect a point in its corresponding region, no other sensor can detect it either. This means that in order to find the

"so-called" coverage holes (i.e., the undetectable points in **F**), it would suffice to compare the VMW-Voronoi region of every sensor with its local coverage area. A VMW-Voronoi diagram with 3 sensors S_1, S_2 and S_3 with the sensing radii 10m, 18m, and 18m, respectively, is depicted in Fig. 9.1.

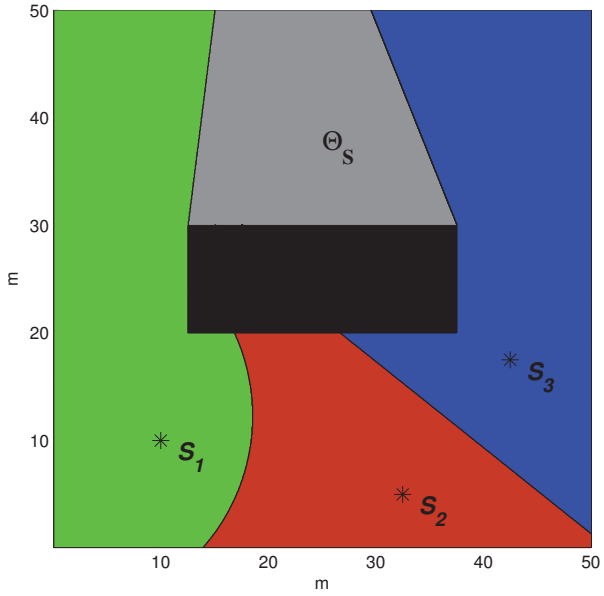


Figure 9.1: An example of the VMW-Voronoi diagram for a group of 3 non-identical sensors in a field with obstacles.

Assumption 9.1. *The communication range of the sensors is bounded (and not necessarily the same for all sensors). This is a limiting factor for sensors, potentially preventing them from communicating with their neighbors, and can result in wrong VMW-Voronoi regions around some sensors. Consequently, such a limitation can negatively affect the detection of coverage holes. Since the number of sensors in a mobile sensor network is typically large (or more precisely, there is a sufficient number of sensors per area unit) [129], [130], it is assumed that the graph representing sensors' communication topology is connected [128]. Hence, each sensor can obtain the information about the locations and sensing radii of the other sensors (and in particular its neighbors) through proper communication routes. Also it is assumed that the obstacles' locations are known by each sensor as a priori information. As*

a result, each sensor can calculate its VMW-Voronoi region accurately.

9.2 Deployment Protocols

In this section, two different deployment protocols are developed for a network of non-identical sensors in the presence of obstacles. The proposed techniques are iterative, where in each iteration every sensor S_i, $i \in \mathbf{n}$, first broadcasts its sensing radius r_i and position P_i to other sensors. Thus, every sensor is able to construct its own VMW-Voronoi region based on the information received from other sensors in the network. Then, every sensor detects coverage holes in its region. When a coverage hole is discovered, the corresponding sensor calculates its new position using one of the proposed algorithms such that the coverage hole is eliminated or at least its total area is reduced by a certain amount if the sensor moves to that position. Once the new location $\acute{P}_i$ is calculated, the coverage area w.r.t. this new location, i.e., $\beta_{\Pi_i}^{\acute{P}_i}$ is evaluated and compared to the current coverage area, i.e., $\beta_{\Pi_i}^{P_i}$. The sensor moves to the new location only if the resultant coverage area is greater than the present value, i.e. $\beta_{\Pi_i}^{\acute{P}_i} > \beta_{\Pi_i}^{P_i}$; otherwise, it does not move in this iteration. In order to have a termination criterion for the algorithms, a proper threshold ϵ is defined; if no sensor can improve its coverage area by this threshold, the algorithm is terminated.

The following theorem shows that any sensor deployment strategy which follows the scheme described in the previous paragraph is guaranteed to increase the total coverage.

Theorem 9.1. *Let the positions of the sensors in the set* $\mathbf{S}$ *be represented by* $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ *with the corresponding VMW-Voronoi regions* $\Pi_1, \Pi_2, \ldots, \Pi_n$. *Let also the invisible region be denoted by* $\Theta_{\mathbf{P}}$. *Assume the sensors move to new positions* $\acute{\mathbf{P}} = \{\acute{P}_1, \acute{P}_2, \ldots, \acute{P}_n\}$ *with the corresponding VMW-Voronoi regions* $\acute{\Pi}_1, \acute{\Pi}_2, \ldots, \acute{\Pi}_n$

such that $\acute{P}_i \neq P_i$ *for all* $i \in \mathbf{k}$*, where* $\mathbf{k}$ *is a non-empty subset of* $\mathbf{n}$*. If the* i*-th coverage area w.r.t.* $\acute{P}_i$ *in the previously constructed VMW-Voronoi region* Π_i *is greater than the previous* i*-th local coverage area (i.e.,* $\beta_{\Pi_i}^{\acute{P}_i} > \beta_{\Pi_i}^{P_i}$*) for all* $i \in \mathbf{k}$*, then the total coverage in the network increases.*

Proof. Denote the total uncovered area (coverage hole) of the field when the sensors are located in $\mathbf{P}$ and $\acute{\mathbf{P}}$ by θ and $\acute{\theta}$, respectively. It is deduced from the characterization of the VMW-Voronoi diagram that:

$$\theta = \Theta_{\mathbf{P}} + \sum_{i=1}^{n} \theta_{\Pi_i}^{P_i} \tag{9.4}$$

It is straightforward to show that by increasing the coverage area in $\Pi_i,\ i \in \mathbf{k}$, the corresponding coverage hole will be decreased. Since it is assumed that the i-th coverage area w.r.t. $\acute{P}_i$ is greater than the i-th local coverage area for any $i \in \mathbf{k}$, one can conclude that:

$$\theta_{\Pi_i}^{\acute{P}_i} < \theta_{\Pi_i}^{P_i}, \quad \forall i \in \mathbf{k} \tag{9.5}$$

On the other hand, it is possible that some of the points in $\theta_{\Pi_i}^{\acute{P}_i}$ and $\Theta_{\mathbf{P}}$ are also covered by other mobile sensors at $\acute{\mathbf{P}}$. Hence:

$$\acute{\theta} \leq \Theta_{\mathbf{P}} + \sum_{i=1}^{n} \theta_{\Pi_i}^{\acute{P}_i} \tag{9.6}$$

From the last two relations and on noting that for any $i \in \mathbf{n} \backslash \mathbf{k}$ by definition $\theta_{\Pi_i}^{\acute{P}_i} = \theta_{\Pi_i}^{P_i}$, one arrives at the following inequality:

$$\acute{\theta} < \Theta_{\mathbf{P}} + \sum_{i=1}^{n} \theta_{\Pi_i}^{P_i} \tag{9.7}$$

Now, it is concluded from (9.4) and (9.7) that:

$$\acute{\theta} < \theta \tag{9.8}$$

which means that the total coverage area increases under this deployment scheme. ■

The above-mentioned procedure will be used in the next two subsections to develop two algorithms, namely, OFP and OMP.

9.2.1 Obstructed Farthest Point (OFP) Strategy

The main idea behind this algorithm is to move every sensor to the farthest point in its VMW-Voronoi region such that any existing coverage hole is covered. If a sensor S_i detects a coverage hole in its corresponding VMW-Voronoi region, it calculates the farthest point $X_{i,far}$ in that region, and moves toward it until this point is covered. Fig. 9.2 shows a sample VMW-Voronoi region constructed by the sensor S_1. The segments g and e are generated due to the two neighboring sensors with sensing radii equal to that of S_1. The segments a and h are obstacle edges, and the segment c is formed because of the field boundary. The edge b, in fact, is constructed by the sight line of the sensor, and finally arcs d and f are formed by two neighboring sensors with larger and smaller sensing radii than that of S_1, respectively. As the figure illustrates, the OFP algorithm finds the farthest point to S_1, (i.e. $X_{1,far}$) as a candidate for the next location of the sensor. Since the coverage area of the sensor within its region will increase by moving to S_1', the sensor moves toward $X_{1,far}$ until it is covered.

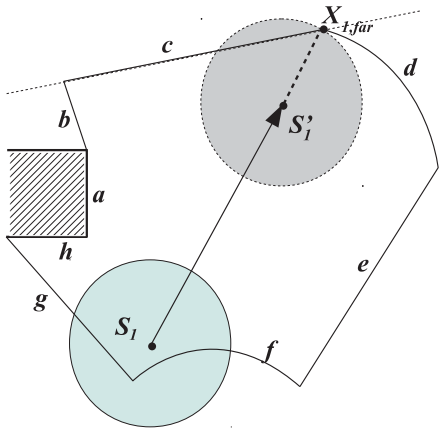
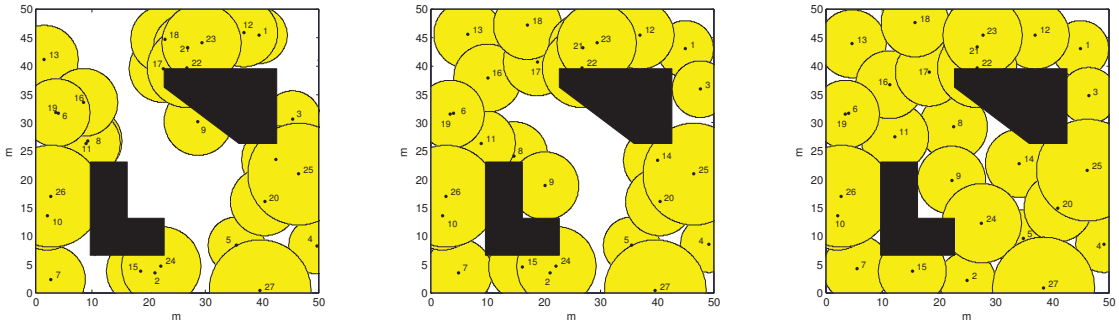


Figure 9.2: A sample VMW-Voronoi region and a candidate point calculated using the OFP method.

Fig. 9.3 shows an example of coverage improvement as a result of using the OFP Algorithm. In this example, 27 mobile sensors with varying sensing ranges

are randomly deployed in a 2D field of size 50m × 50m. 15 sensors have a sensing radius of 6m, 6 with a sensing radius of 5m, 3 with a sensing radius of 7m, and the remaining 3 with a sensing radius of 9m. The communication range of each sensor is assumed to be 10/3 times its sensing range. Three snapshots of the field coverage are shown in Fig. 9.3. The circles represent the sensing area of each sensor. As observed, the coverage is 68.44% initially, but it increases to 81.31% after the first iteration of the presented algorithm (Fig. 9.3(b)). The final coverage is 95.32% (Fig. 9.3(c)).



(a) (b) (c)

Figure 9.3: Snapshots of the execution of the movement of the sensors under the OFP algorithm. (a) Initial coverage; (b) field coverage after the first round, and (c) final coverage.

9.2.2 Obstructed Minmax Point (OMP) Strategy

Although the OFP algorithm performed well in most simulated scenarios, there exist certain network setups and node configurations, where it might not be as effective. Fig. 9.4 shows such an example. The next candidate location for the sensor under the OFP algorithm does not lead to any improvement in the coverage area of the sensor within its region. Thus, the mobile sensor remains in its previous location. However, there exit other potential positions for sensor relocation that can increase its coverage area within the corresponding region. Another configuration for which the OFP algorithm is not as effective is when the calculated candidate point lies on

a sight line connecting the sensor to an obstacle (e.g., see Fig. 9.5). These candidate points are, by definition, on the sight line; therefore, if the sensor moves to such a position, then there is a good chance that part of the sensor's sensing capability is blocked in its corresponding region.

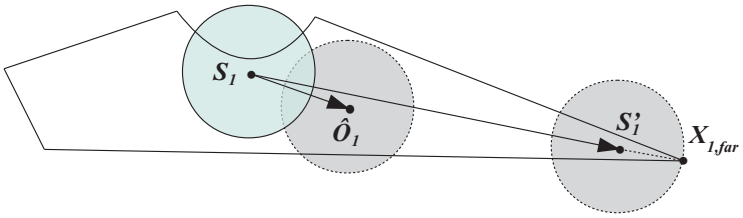


Figure 9.4: A sample VMW-Voronoi region for which the OFP method performs poorly because of a narrow angle, while the OMP strategy finds a proper candidate location.

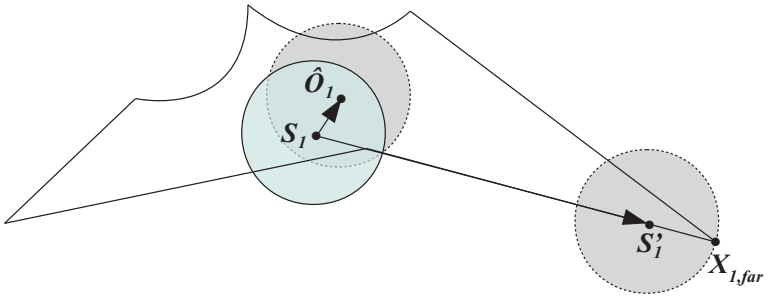
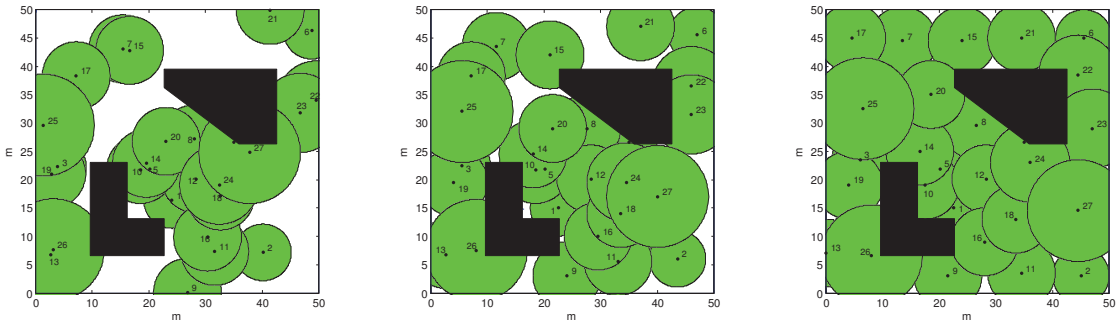


Figure 9.5: A sample VMW-Voronoi region for which the OFP method performs poorly because of the sensor's sight line, while the OMP strategy finds a proper candidate location.

As it can be concluded from the above discussion, although the OFP algorithm is effective in many cases, one may find a proper location for the sensor in the special cases described above. The obstructed minmax point (OMP) strategy is introduced in the sequel to address this shortcoming of the OFP algorithm. The main idea behind the OMP strategy is that to achieve maximum coverage, no sensor should be too far from any point in its corresponding VMW-Voronoi region. The OMP strategy finds the location whose distance from the farthest point of the region is minimum and considers it as the candidate location for the sensor in the next step.



(a) (b) (c)

Figure 9.6: Snapshots of the execution of the movement of the sensors under the OMP algorithm. (a) Initial coverage; (b) field coverage after the first round, and (c) final coverage.

This point is called the *OMP centroid*, and is denoted by $\hat{O}_i$ for the i-th region, $i \in \mathbf{n}$. It is clear that the candidate point $\hat{O}_1$ in Fig. 9.4 yield better coverage compared to the one obtained by using the OFP technique. Also, in Fig. 9.5, the OMP strategy performs more efficiently than the OFP algorithm due to the specific shape of the region which makes the farthest point lie on the sight line of the sensor.

Consider the initial setting of Fig. 9.6(a), and let the OMP strategy be employed. The results are depicted in Figs. 9.6(b) and 9.6(c), where it is shown that after the first round the coverage increases from 67.09% to 84.94%, and that the final coverage is 97.62%.

Theorem 9.2. *The OFP and OMP algorithms are convergent.*

Proof. The proof is similar to that of Theorem 5.4, and is omitted here. ■

The following theorem provides an upper-bound on the number of rounds required to run the algorithm, as a function of ϵ.

Theorem 9.3. *Consider a set of n mobile sensors $\mathbf{S}$, randomly deployed in a 2D field. Using any of the proposed algorithms with the coverage improvement threshold ϵ, the number of required rounds to run the algorithm is at most $\frac{A_{total}-A_{obs}}{\epsilon}$, where A_{total} and A_{obs} are the overall area of the field and area of the obstacles, respectively.*

Proof. Let the number of rounds required to run the algorithm in order to meet the termination condition be denoted by ζ_f. Let also the total uncovered area of the field in the k-th round be represented by $\theta(k)$, and note that $\beta(k) = A_{total} - A_{obs} - \theta(k)$. Denote the position of the sensors and their corresponding VMW-Voronoi regions in the k-th round by $\mathbf{P}(k) = \{P_1(k), P_2(k), \ldots, P_n(k)\}$ and $\Pi_1(k), \Pi_2(k), \ldots, \Pi_n(k)$, respectively. Also the invisible region in the k-th round is denoted by $\Theta_{\mathbf{P}}(k)$. From the properties of the VMW-Voronoi diagram, one can conclude that:

$$\theta(k) = \Theta_{\mathbf{P}}(k) + \sum_{i=1}^{n} \theta_{\Pi_i(k)}^{P_i(k)}, \ \forall 1 \leq k \leq \zeta_f \tag{9.9}$$

Define the *moving set of the k-th round* as the largest subset of $\mathbf{S}$ that moves in the k-th round, and denote the indices of the sensors in this set by $\mathbf{Ix}(k)$. Note that at least one sensor moves in the k-th round, i.e. $\mathbf{Ix}(k) \neq \emptyset$, $\forall 1 \leq k \leq \zeta_f$. Note also that the i-th sensor, $i \in \mathbf{Ix}(k)$, moves in the k-th round if $\beta_{\Pi_i(k)}^{P_i(k+1)} \geq \beta_{\Pi_i(k)}^{P_i(k)} + \epsilon$. This means that:

$$\theta_{\Pi_i(k)}^{P_i(k+1)} \leq \theta_{\Pi_i(k)}^{P_i(k)} - \epsilon, \quad \forall i \in \mathbf{Ix}(k) \tag{9.10}$$

Note that some of the points in $\Theta_{\mathbf{P}}(k)$ might be covered by some sensors located at $\mathbf{P}(k+1)$. In addition some of the points in $\theta_{\Pi_i(k)}^{P_i(k+1)}$ might also be covered by another sensor located at $P_j(k+1)$, for some $j \in \mathbf{n}\backslash\{i\}$. Hence:

$$\theta(k+1) \leq \Theta_{\mathbf{P}}(k) + \sum_{i=1}^{n} \theta_{\Pi_i(k)}^{P_i(k+1)} \tag{9.11}$$

From the last two relations and on noting that for any $i \in \mathbf{n}\backslash\mathbf{Ix}(k)$ the i-th sensor does not move (which implies $\theta_{\Pi_i(k)}^{P_i(k+1)} = \theta_{\Pi_i(k)}^{P_i(k)}$), one arrives at:

$$\theta(k+1) \leq \Theta_{\mathbf{P}}(k) + \sum_{i=1}^{n} \theta_{\Pi_i(k)}^{P_i(k)} - |\mathbf{Ix}(k)|\, \epsilon \tag{9.12}$$

It is now concluded from (9.9) and (9.12) that:

$$\theta(k+1) \leq \theta(k) - |\mathbf{Ix}(k)|\, \epsilon \leq \theta(k) - \epsilon \tag{9.13}$$

or equivalently:

$$\beta(k+1) \geq \beta(k) + |\mathbf{Ix}(k)|\,\epsilon \geq \beta(k) + \epsilon \tag{9.14}$$

which implies that using the underlying sensor relocation scheme, in each round the total covered area increases by at least ϵ. Therefore, the total amount of increased coverage from the first round to the termination round is greater than or equal to $\zeta_f \epsilon$. Since the total covered area is always less than or equal to $A_{total} - A_{obs}$, hence $A_{total} - A_{obs} \geq \zeta_f \epsilon$ or equivalently $\frac{A_{total} - A_{obs}}{\epsilon} \geq \zeta_f$. ■

In the next section, the performance of the proposed algorithms in terms of the coverage area, energy consumption of the sensors, rate of convergence, and computational complexity are investigated.

9.3 Simulation Results

Consider a sensing field of size 50m × 50m with two obstacles. Also consider a network of 36 mobile sensors with varying sensing radii i.e. 20 sensors with a sensing radius of 6m, 8 with a sensing radius of 5m, 4 with a sensing radius of 7m, and 4 with a sensing radius of 9m. The communication range of each sensor is assumed to be 10/3 times its sensing range; e.g., a sensor with a sensing radius of 6m has a communication radius of 20m.

Define the *coverage factor* as the ratio of the covered area to the total area in the field. The simulation results presented in this section are the average values obtained from 20 different random initial sensor deployments. Coverage factor under both algorithms is shown in Fig. 9.7. While both algorithms provide satisfactory coverage, the OMP technique exhibits a better performance in this example.

To investigate the effect of the number of sensors on the performance of the algorithms, we considered four more setups: n=9, 18, 27, and 45, in addition to n=36 discussed above. It is assumed that the changes in the number of identical

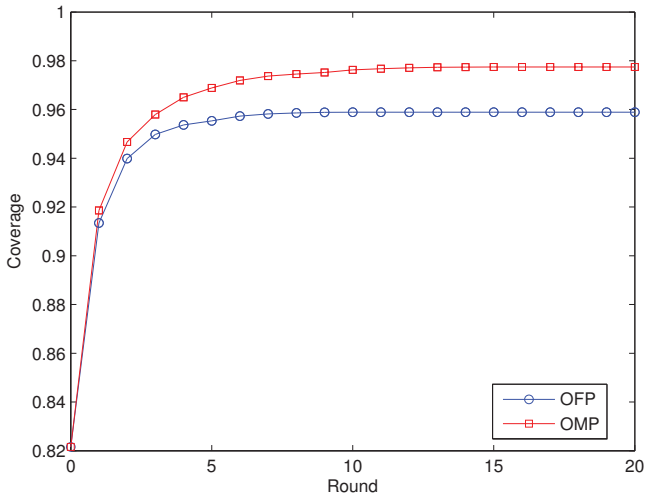


Figure 9.7: Network coverage per round for 36 sensors.

sensors in the new setups are proportional to the changes in the total number of sensors. For example, for n=27 there will be 15 sensors with sensing radius of 6m, 6 with sensing radius of 5m, 3 with sensing radius of 7m, and 3 with sensing radius of 9m. Fig. 9.8 shows the resulting final coverage versus number of sensors. Both algorithms yield satisfactory results with the OMP strategy still performing better.

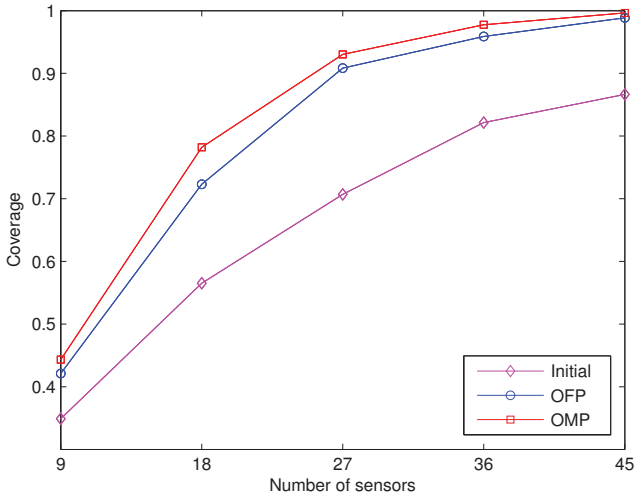


Figure 9.8: The coverage factor achieved for different number of sensors under the proposed algorithms.

An important factor in the performance evaluation of different deployment

techniques is the time it takes to reach the desired termination criteria. Assuming that both relocation strategies require the same deployment time in each round of algorithm execution, then, the number of rounds to reach a predetermined termination criteria is a good measure of the deployment speed of each algorithm.

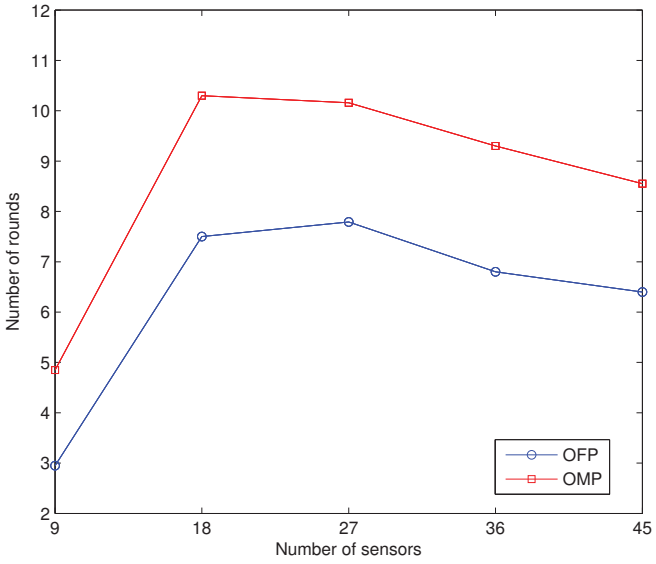


Figure 9.9: The number of rounds required to reach the termination conditions for different number of sensors using the proposed algorithms.

Fig. 9.9 shows that under both algorithms the number of rounds required to meet a certain termination condition increases with the number of sensors up to a certain point, and then decreases after that. The reason can be explained as follows. When the number of sensors is small, the sizes of their corresponding VMW-Voronoi regions are relatively larger than their coverage circles. And, it will be likely for some sensors that their entire coverage circles are enclosed within their VMW-Voronoi regions. Therefore, further relocation of each sensor in its region would not increase the coverage level. On the other hand, when the number of sensors is relatively large, the size of their corresponding VMW-Voronoi regions will be small. And, with a high likelihood, the coverage circles of most sensors enclose their VMW-Voronoi regions. This in turn implies that the termination condition is

satisfied in smaller number of rounds. Fig. 9.9 shows that the convergence rate of the OFP algorithm is faster than that of the OMP; therefore, it is a better candidate for field coverage if higher deployment speed is required.

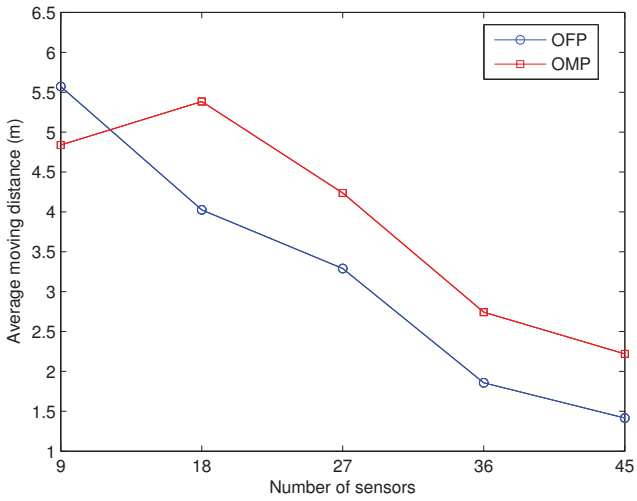


Figure 9.10: The average distance each sensor travels for different number of sensors, under the proposed algorithms.

Another important factor in performance evaluation of deployment algorithms in mobile sensor networks is energy-efficiency. Movement of a sensor, more precisely, the distance it travels and also the number of times it stops (impact of static friction) are the dominant sources of energy consumption. Fig. 9.10 shows the average distance traveled by a sensor versus number of sensors in the network. As observed, the average traveled distance for a large number of sensors is small. For large number of sensors, the distance between each sensor's position and its candidate location in its corresponding VMW-Voronoi region decreases. Therefore, the average traveling distance required by a sensor decreases. This, in turn, leads to a reduction in energy consumption. Also, it can be seen from Fig. 9.10 that the OFP algorithm is more efficient than the OMP strategy for a larger number of sensors. Fig. 9.11 shows the number of relocations versus the number of sensors in the network. In both algorithms, and up to certain value, the number of relocations increases with the

number of sensors, and then decreases after that. Again, this can be justified based on the relative sizes of the sensors coverage circles versus their VMW-Voronoi regions. Figs. 9.10 and 9.11 clearly demonstrate that the OFP algorithm outperforms the OMP strategy in energy consumption.

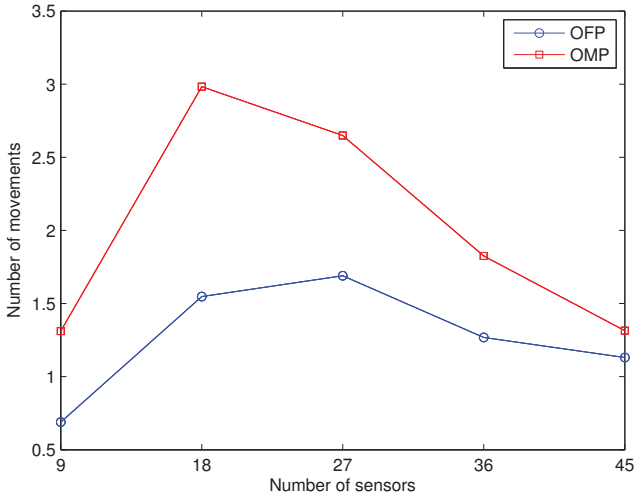


Figure 9.11: The number of movements required for different number of sensors, using the proposed algorithms.

Remark 9.1. *Note that the algorithms introduced in this chapter differ only in the way the new locations of the sensors are determined. Since the complexity of finding the new location of the i-th sensor in the OFP strategy is more than that in OMP, hence the OFP algorithm outperforms the OMP algorithm as far as the computational complexity is concerned.*

The above discussion is summarized below:

1. The OMP algorithm is more preferable as far as network coverage is concerned.
2. The OFP algorithm is more desirable when:
   - the deployment time is the main concern.
   - the energy consumption is the main concern.

- the computational complexity is concerned.

Chapter 10

Distributed Deployment Strategies for Improved Coverage in a Network of Mobile Sensors with Prioritized Sensing Field

In this chapter, efficient deployment strategies are proposed for a mobile sensor network, where the coverage priority of different points in the field is specified by a priority function. The multiplicatively weighted Voronoi (MW-Voronoi) diagram is utilized to find the coverage holes of the network for the case where the sensing ranges of different sensors are not the same. Under the proposed strategies, each sensor detects coverage holes within its MW-Voronoi region, and then moves in a proper direction to reduce their size. Since the coverage priority of the field is not uniform, the target location of each sensor is determined based on the weights of the vertices or the points inside the corresponding MW-Voronoi region. Simulations validate the theoretical results.

The plan of the rest of the chapter is as follows. The problem is formulated

in Section 10.1, and some important assumptions and definitions are also given which will be used later to develop the main results. Section 10.2 presents the main contributions of the chapter, where new deployment algorithms are introduced, and finally the proposed algorithms are compared in Section 10.3.

10.1 Problem Formulation

Consider a group of n mobile sensors, randomly distributed in the sensing field, and let the sensing radius of the i-th sensor be denoted by r_i (note that the sensing radii of the sensors can be different). The coverage priority of different points in the field is assumed to be specified by a priority function $\varphi(q)$. In other words, the coverage importance of the point q is more than that of point p if $\varphi(q) > \varphi(p)$.

It is desired to move the sensors and place them in proper positions in the field using a distributed deployment strategy such that the more important points are covered as much as possible. In other words, the objective is to increase the weighted coverage area with limited information exchange between sensors.

Definition 10.1. *The integral of the priority function over the MW-Voronoi region* Π_i *is referred to as the* i-th region weight*, and is denoted by* α_{Π_i}*, i.e.:*

$$\alpha_{\Pi_i} = \int_{\Pi_i} \varphi(q) dq$$

Also, the integral of the priority function over the entire sensing field is referred to as the total field weight*, and is denoted by* α*. It is straightforward to show that* $\alpha = \sum_{i=1}^{n} \alpha_{\Pi_i}$.

Definition 10.2. *Consider a sensor* S_i *with the sensing radius* r_i *and the corresponding MW-Voronoi region* Π_i*,* $i \in \mathbf{n}$*, and let* x *be an arbitrary point inside* Π_i*. The integral of the priority function over the* i*-th coverage area w.r.t.* x *is referred*

to as the i-th weighted coverage w.r.t. x*, and is mathematically characterized as:*

$$\beta^x_{\Pi_i} = \int_{\Pi_i \cap C(x,r_i)} \varphi(q)dq \tag{10.1}$$

where $C(x, r_i)$ *is a circle of radius* r_i *centered at* x*. Also, the* i-th weighted hole w.r.t. x *is denoted by* $\theta^x_{\Pi_i}$*, and is expressed as:*

$$\theta^x_{\Pi_i} = \alpha_{\Pi_i} - \beta^x_{\Pi_i} \tag{10.2}$$

Inside an MW-Voronoi region, the weighted covered and uncovered areas w.r.t. the location p_i *of the sensor* S_i *(i.e.* $\beta^{p_i}_{\Pi_i}$ *and* $\theta^{p_i}_{\Pi_i}$*) are called the* i*-th local weighted coverage and* i*-th local weighted hole of that sensor, respectively. Furthermore, the integral of the priority function over the covered area (non-covered area) in the field is referred to as the* total weighted coverage *(*total weighted hole*).*

10.2 Deployment Protocols

In this section, three distributed deployment algorithms are introduced for a mobile sensor network. The proposed deployment algorithms perform iteratively until a prespecified termination condition is satisfied. Each iteration in the proposed algorithms consists of four phases. In the first phase, every sensor broadcasts its location and sensing radius to other sensors, and constructs its MW-Voronoi region subsequently based on the information it receives from other sensors. Then in the second phase, each sensor uses the available information to compute its destination point in its MW-Voronoi region according to the specific deployment strategy. Once the new target location $\acute{p_i}$ is determined, the weighted coverage area w.r.t. this location (i.e. $\beta^{\acute{p_i}}_{\Pi_i}$) is obtained in the third phase. If this value is greater than the previous local weighted coverage area (i.e. $\beta^{\acute{p_i}}_{\Pi_i} > \beta^{p_i}_{\Pi_i}$), then the sensor moves to the new destination; otherwise, it remains in its current position. Finally, in the termination phase, if the weighted covered area by none of the sensors within its corresponding

MW-Voronoi region is improved by a certain amount, then the iteration stops. Note that the first, third and fourth phases described above are exactly the same. Thus, the algorithms introduced later in this section differ only in the second phase where the new location of each sensor is determined. Using an approach similar to the proof of Theorem 4.1, it can be shown that the total weighted coverage under this type of deployment scheme increases, in general.

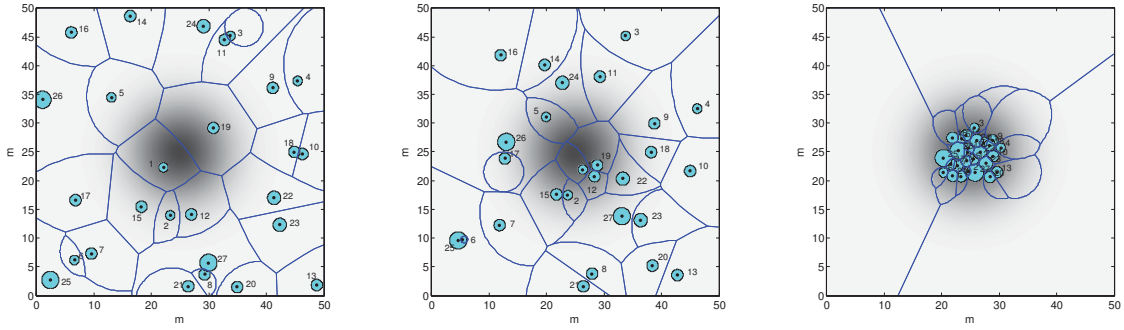
Assumption 10.1. *It is implicitly assumed that a synchronization protocol (similar to one in [137]) is implemented to guarantee that all sensors start the first phase at the same time. Furthermore, the coverage rounds are assumed to be sufficiently long, so that all four phases described above can be completed in one round.*

The proposed deployment strategies will be presented in the sequel.

10.2.1 The Maximum Weighted Vertex (MWV) Strategy

In this strategy, each sensor moves toward the vertex with maximum weight in its MW-Voronoi region. This vertex is referred to as the *heaviest vertex*, and is denoted by $V_{i,max}$ for the i-th region. According to this strategy, all sensors search for any coverage holes in their MW-Voronoi regions. Once the coverage holes are detected, each sensor identifies the heaviest vertex in its MW-Voronoi region. Then, for any $i \in \mathbf{n}$, S_i moves toward $V_{i,max}$ and continues moving until it is covered. This occurs when the distance of the i-th sensor from $V_{i,max}$ is equal to its sensing radius.

As an operational example of the MWV strategy, consider 27 sensors randomly deployed in a $50\text{m} \times 50\text{m}$ flat space: 15 with a sensing radius of 1m, 6 with a sensing radius of $\frac{5}{6}$m, 3 with a sensing radius of $\frac{7}{6}$m, and 3 with a sensing radius of 1.5m. Moreover, the communication range of each sensor is assumed to be 20m. The priority function representing the network coverage priority in this example is given by $\varphi(q) = exp(-0.4[(x_q - 25)^2 + (y_q - 25)^2])$, where x_q and y_q are the abscissa and



(a) (b) (c)

Figure 10.1: Snapshots of the execution of the MWV strategy for a sensing field, where the coverage priority of different points in it is depicted by different gray levels (the white color represents the lowest priority and the black color the highest). (a) Initial coverage; (b) coverage after the first round, and (c) final coverage.

ordinate of the point q, respectively. Each point in the field is represented by a gray level proportional to the coverage priority of that point. In Fig. 10.1, three snapshots are provided, and in each one both the sensing circles of every sensor (filled circles) and the MW-Voronoi regions are depicted. It can be observed from this figure that in the final round the sensors concentrate on the area with higher coverage priority.

While the sensor deployment strategy discussed above proves effective in many practical cases, it may not be as effective when there are small number of sensors with small sensing ranges in the field. In such cases, the MW-Voronoi regions are relatively large, and hence there is a good chance that each area with high coverage priority is mainly located in only one of the regions (see Fig. 10.2). Thus, by moving toward the heaviest vertex in a region, the high-priority area in that region might not be covered. For example, under the MWV strategy the sensor S_4 in Fig. 10.2 would move toward the vertex $V_{4,max}$, missing the high-priority area which is roughly in the opposite direction. Furthermore, in the case when the priority function varies significantly over some regions, the corresponding sensors might not move in the proper direction. In the special case, if an MW-Voronoi region has no vertices (i.e.,

it is a circle), the corresponding sensor does not move under the MWV strategy, which is another shortcoming of this strategy. To remedy the above-mentioned problems, a new deployment technique will be presented in the next subsection.

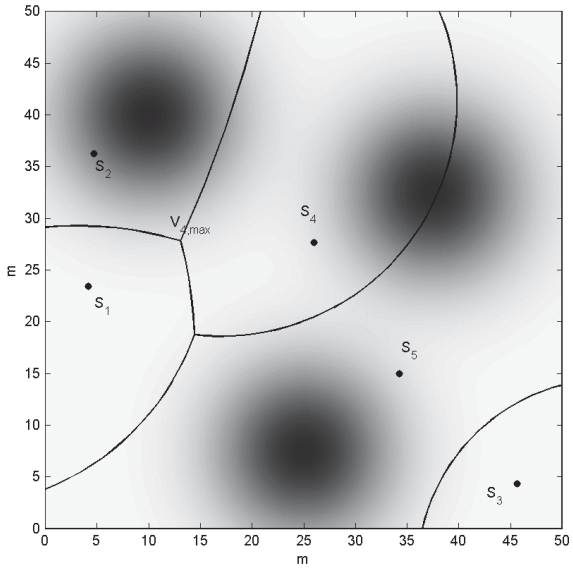
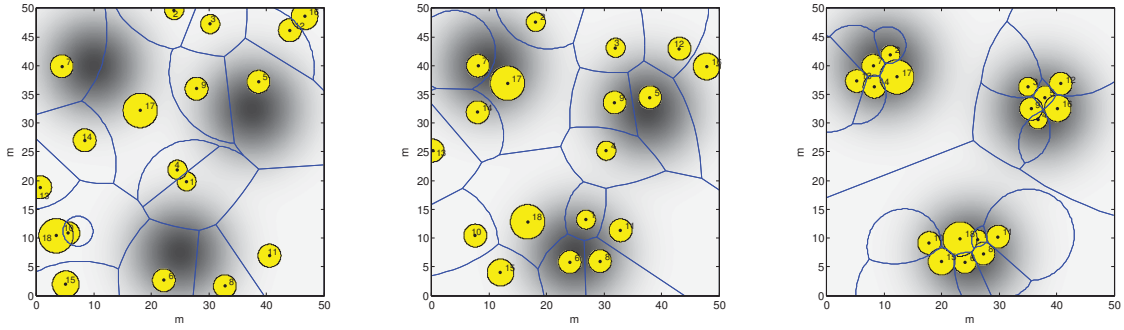


Figure 10.2: A network of 5 mobile sensors in a weighted field, where the MWV algorithm is not as effective because the sensing range of every sensor is small.

10.2.2 The Maximum Weighted Point (MWP) strategy

In this strategy, each sensor moves to a point in its MW-Voronoi region which has the maximum weight. This point will be referred to as the *heaviest point*, and is denoted by $p_{i,max}$ for the i-th region. According to this strategy, once a coverage hole is discovered in an MW-Voronoi region, the corresponding sensor finds the heaviest point in that region and moves toward it up to the position from which $p_{i,max}$ is covered.

As an example, consider the initial deployment of Fig. 10.3, and let the priority function be equal to $\varphi(q) = exp(-0.4[(x_q - 10)^2 + (y_q - 40)^2]) + exp(-0.4[(x_q - 25)^2 + (y_q - 7.5)^2]) + exp(-0.4[(x_q - 37.5)^2 + (y_q - 32.5)^2])$. Let also 18 sensors with a communication range of 20m be randomly deployed in the field: 10 with a sensing



(a) (b) (c)

Figure 10.3: Snapshots of the execution of the MWP strategy where different gray levels are used to indicate the coverage priorities, similar to Fig. 10.1. (a) Initial coverage; (b) coverage after the first round, and (c) final coverage.

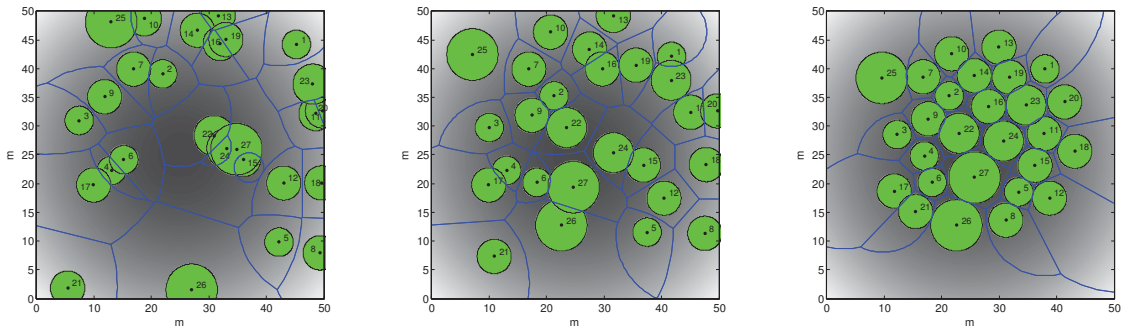
radius of 2m, 4 with a sensing radius of $\frac{5}{3}$m, 2 with a sensing radius of $\frac{7}{3}$m, and 2 with a sensing radius of 3m. As it can be observed from Fig. 10.3, after the final round the sensors are more concentrated on high-priority areas in the field.

10.2.3 The Maximum Distance Weight (MDW) Strategy

The two weight-based techniques discussed thus far are not suitable when the priority function is smooth. For instance, when the weight of all points of the field are equal (i.e., $\varphi(q)$ =constant), sensors do not move under the MWV and MWP strategies. This motivates the development of a new strategy called MDW, which operates based on both distance and weight.

For any $i \in \mathbf{n}$, the MDW strategy finds a point inside the i-th MW-Voronoi region whose distance from S_i multiplied by its weight is maximum. This point will be referred to as the *i-th MDW centroid*, and will be denoted by $p_{i,MDW}$. Once this point is obtained, S_i moves toward it and continues moving until $p_{i,MDW}$ is covered. This occurs when the distance of the i-th sensor from the point $p_{i,MDW}$ is equal to its sensing radius.

Fig. 10.4 shows an operational example of the MDW strategy. In this example, 27 sensors with the communication range of 20m are randomly deployed in a 50m



(a) (b) (c)

Figure 10.4: Snapshots of the execution of the MDW strategy where different gray levels are used to indicate the coverage priorities, similar to Fig. 10.1. (a) Initial coverage; (b) coverage after the first round, and (c) final coverage.

by 50m flat space: 15 with a sensing radius of 3m, 6 with a sensing radius of 2.5m, 3 with a sensing radius of 3.5m, and 3 with a sensing radius of 4.5m. The priority function for this example is $\varphi(q) = exp(-0.004[(x_q - 25)^2 + (y_q - 25)^2])$. It can be observed from this figure that in the final round the weighted coverage significantly increases.

Remark 10.1. *It is worth mentioning that for the case when all sensors have the same sensing capability and the weight of every point in the field is the same, the MDW strategy will be the same as the VOR strategy proposed in [64]. In other words, the MDW strategy proposed here is the generalized form of the VOR technique.*

Remark 10.2. *In order to prevent sensors from oscillatory movements, a control mechanism similar to the one in [64] can be used here. Under this mechanism, each sensor compares the newly computed direction with the previous one, and will move only if the new direction is consistent with that in the preceding round.*

Remark 10.3. *It is important to note that even in the case of a centralized deployment scheme, no optimal solution is available, in general, for the coverage problem in a non-uniform sensing field.*

10.3 Comparative Study

In this section, the three algorithms proposed in Section 10.2 are applied to a flat space of size 50m $\times$ 50m with different number of sensors. In these simulations, the algorithms terminate when none of the sensors' weighted coverage in its corresponding MW-Voronoi region would increase by more than 1% in the next move. Note that the results presented in this section for weighted coverage are all the average values obtained by using 20 random initial deployments for sensors.

Remark 10.4. *The sensing field is divided into small grid cells, and some computations are performed over the centers of the cells, which will be referred to as the grid centers in the sequel. The choice of the cell size is made based on the distribution of the priority function in different points of the field, desired precision, and processing capability of different sensors. The computational task for each algorithm consists of three phases, which are performed by each sensor individually. In the first phase, every sensor finds those grid centers which lie inside its MW-Voronoi region for all grid centers). The second phase depends on the particular strategy adopted: In the MWV method each sensor finds the heaviest vertex in its MW-Voronoi region (note that each vertex is the intersection of two Apollonian circles). In the MWP and MDW techniques, on the other hand, each sensor finds a proper point among all grid centers in its MW-Voronoi region. In the third phase, the local weighted coverage of every sensor is compared with its weighted coverage w.r.t. the newly computed destination point. To this end, the weights of the corresponding grid centers in each MW-Voronoi region are summed up and the result is multiplied by the area of each cell. It is worth mentioning that with the current state-of-the-art technology for industrial sensors (e.g. Mica2 [138] or Epic [139]), the computations described above can be efficiently carried out in a short period of time.*

Example 10.1. *In this example,* 27 *sensors with a communication range of* 20m

are randomly deployed in the flat space described above: 15 with a sensing radius of 6m, 6 with a sensing radius of 5m, 3 with a sensing radius of 7m, and 3 with a sensing radius of 9m. The priority function representing the importance of coverage of different points in the field is assumed to be $\varphi(q) = exp(-0.4[(x_q-25)^2+(y_q-25)^2])$ *for this example. Fig. 10.5 depicts the weighted coverage factor, defined as the ratio of the total weighted coverage to the total field weight, after each round of the three algorithms. The figure shows that the performances of the MDW and MWV strategies in this example are more or less the same. In fact, it can be verified that when a relatively large number of sensors with large sensing ranges are distributed in the field, all three algorithms reach a satisfactory weighted coverage. Since the computational complexity for finding the point* $V_{i,max}$ *is less than that for finding* $p_{i,max}$ *and* $p_{i,MDW}$*, the MWV algorithm is more efficient in such scenarios as far as the processing capability of the sensors is concerned.* ■

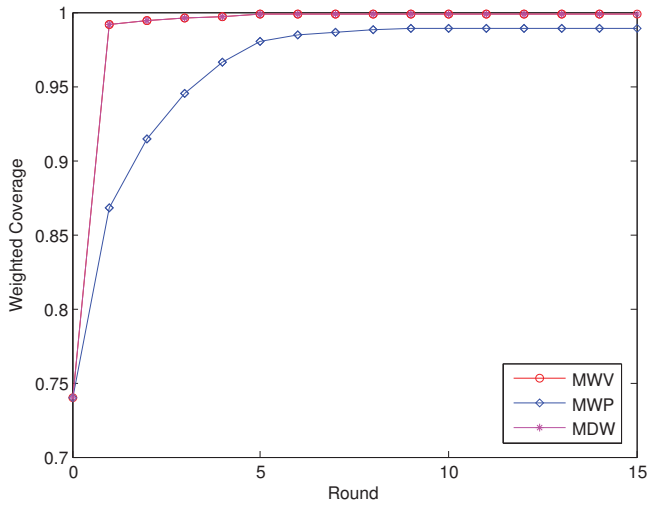


Figure 10.5: The weighted coverage per round for Example 1.

Example 10.2. *Consider* 9 *sensors with the communication range of* 20m *randomly deployed in the flat space described earlier: 5 with a sensing radius of 1m, 2 with a sensing radius of* $\frac{5}{6}m$*, 1 with a sensing radius of* $\frac{7}{6}m$*, and 1 with a sensing radius of 1.5m. Let the priority function be equal to* $\varphi(q) = exp(-k[(x_q - 10)^2 + (y_q -$

$40)^2]) + exp(-k[(x_q - 25)^2 + (y_q - 7.5)^2]) + exp(-k[(x_q - 37.5)^2 + (y_q - 32.5)^2])$, *where* $k = 0.4$. *Due to the relatively small number of sensors in this example (compared to the field size), the MW-Voronoi regions are comparatively large. Furthermore, since the priority function is sharp (concentrated in three different areas), each area with a large weight will likely lie mainly inside one MW-Voronoi region (not on its boundaries). On the other hand, because of the relatively small sensing radius of the sensors, there is a good chance that these important areas would not be covered by moving toward the vertex with maximum weight in the MWV strategy or toward the point with the maximum weighted distance from the corresponding sensor in the MDW strategy. Hence, the MWP algorithm outperforms the other two in this case (see Fig. 10.6).* ■

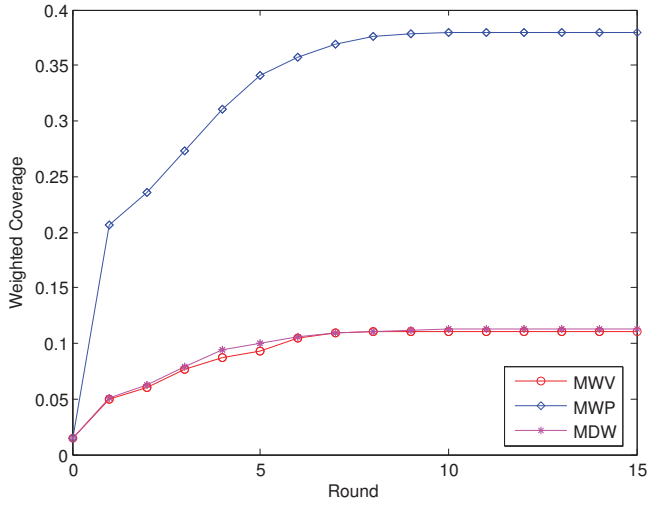


Figure 10.6: The weighted coverage per round for Example 2.

Example 10.3. *Consider the sensor network described in Example 10.1, and let the priority function here be of the same form as in the previous example, but with* $k = 0.004$. *This priority function is relatively smooth, and confirmed by Fig. 10.7, the MWV and MWP strategies are not as effective as the MDW strategy for this case. In addition, since the sensing ranges of the sensors are relatively large, it is more likely that the sensors will have overlapped sensing areas, if they move to the*

heaviest points or vertices, without taking the traveling distance into consideration. In general, when the priority function is not sharp and the sensing ranges of the sensors are relatively large, the MDW strategy outperforms the other two. ■

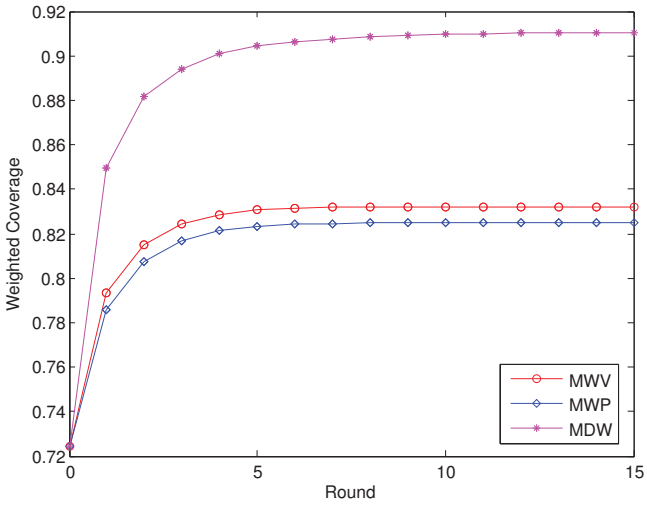


Figure 10.7: The weighted coverage per round for Example 3.

Example 10.4. *The performance of the proposed algorithms is investigated for two different setups in this example. The first setup is the same as the one in Example 10.2, and the second setup is the same as that in Example 10.1. In both scenarios, it is assumed that initially the sensors are distributed randomly in the field. The priority function is of the same form as in Example 10.2 in both setups. Fig. 10.8 depicts the final weighted coverage for different values of k, in the first setup. As it can be observed from this figure, when there are a small number of sensors with small sensing ranges in the field, the MWP strategy results in a better weighted coverage compared to the other two algorithms, and this superiority is considerable when the priority function is sharp. Fig. 10.9 shows the final weighted coverage for different values of k in the second setup. As it can be seen from this figure, when a large number of sensors with high sensing capabilities are distributed in the field, the MDW strategy outperforms the other two. As mentioned before, this superiority is significant when the priority function is smooth.* ■

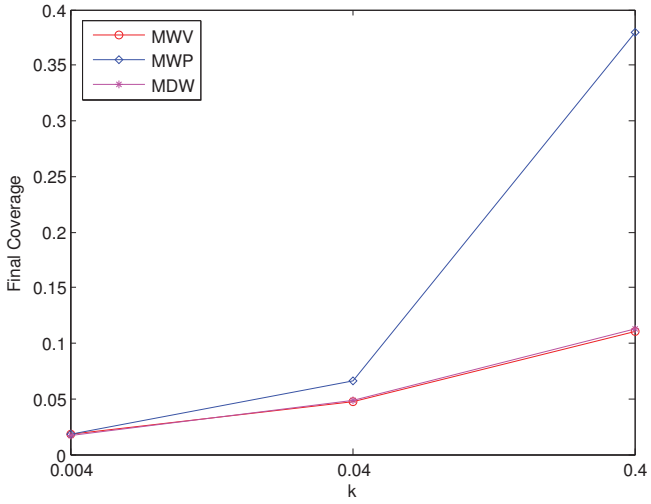


Figure 10.8: The final weighted coverage for different values of k, in the first scenario of Example 10.4.

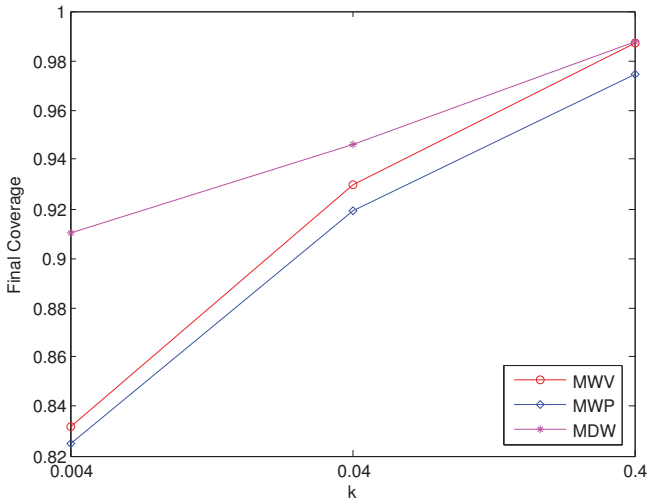
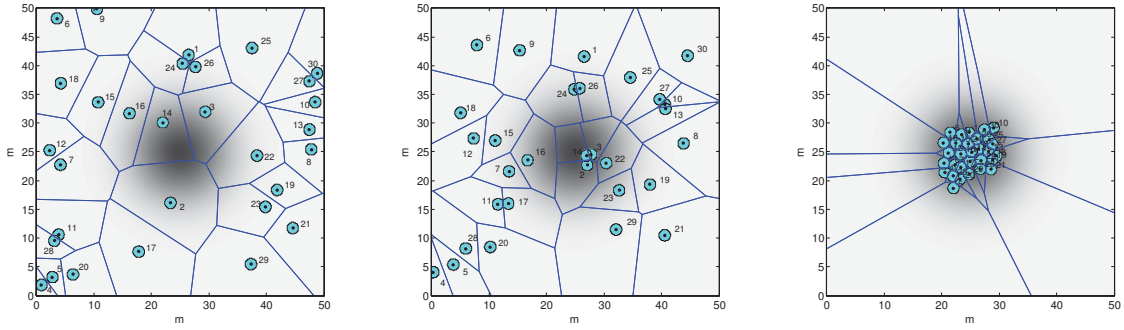


Figure 10.9: The final weighted coverage for different values of k, in the second scenario of Example 10.4.



(a) (b) (c)

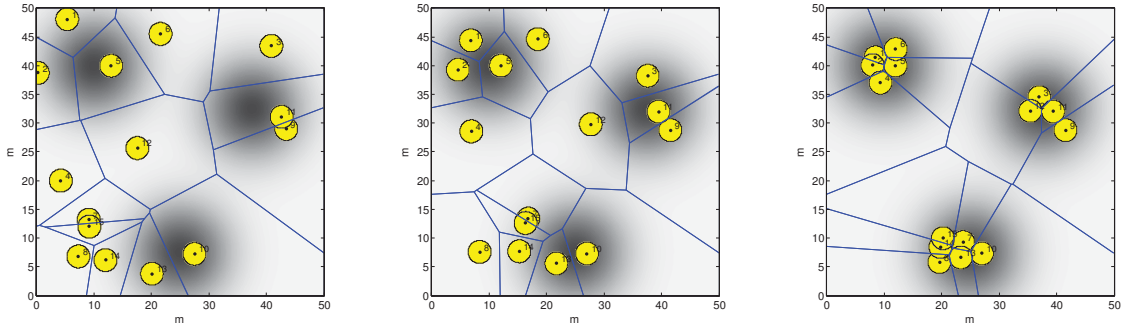
Figure 10.10: Snapshots of the execution of the MWV strategy for the first scenario of Example 10.5 (the coverage priority is indicated by different gray levels). (a) Initial coverage; (b) coverage after the first round, and (c) final coverage.

Example 10.5. *Three operational scenarios are considered here for the special case of identical sensors. The communication range of the sensors in all three scenarios is 20m, and initially they are assumed to be placed randomly in the field described earlier. In the first scenario, 30 sensors with a sensing range of 1m are considered, and the priority function is the same as that in Example 10.1. Three snapshots in this case are provided in Fig. 10.10, and in each one the sensing circles of every sensor (filled circles) as well as the Voronoi polygons are depicted. It can be observed from this figure that in the final round the sensors are more concentrated in the area with higher coverage priority.*

In the second scenario, 15 sensors with a sensing range of 2m are deployed in the field, with the same priority function as in Example 10.2. Three snapshots are provided in Fig. 10.11, similar to Fig. 10.10, which show the good performance of the MWP strategy for this scenario.

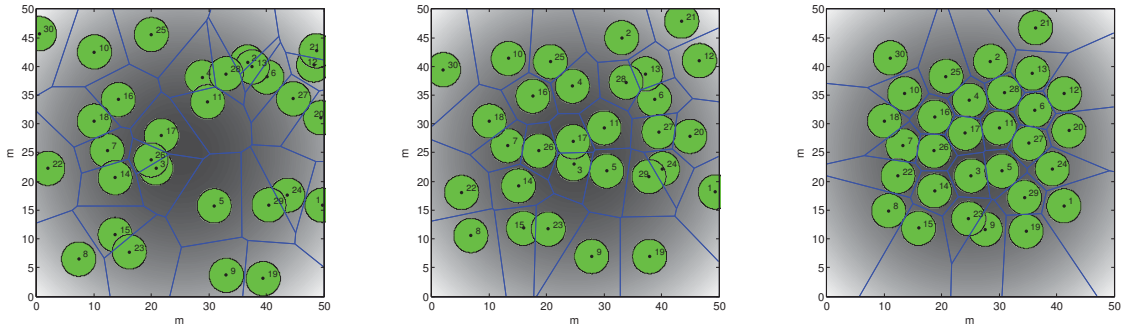
In the third scenario, 30 sensors with a sensing range of 3m are considered, and the priority function is $\varphi(q) = exp(-0.004[(x_q - 25)^2 + (y_q - 25)^2])$ *as shown in Fig. 10.12. It can be observed from this figure that in the final round the total weighted coverage significantly increases.*

It is to be noted that in all three scenarios presented above the resultant Voronoi



(a) (b) (c)

Figure 10.11: Snapshots of the execution of the MWP strategy for the second scenario of Example 10.5 (the coverage priority is indicated by different gray levels). (a) Initial coverage; (b) coverage after the first round, and (c) final coverage.



(a) (b) (c)

Figure 10.12: Snapshots of the execution of the MDW strategy for the third scenario of Example 10.5 (the coverage priority is indicated by different gray levels). (a) Initial coverage; (b) coverage after the first round, and (c) final coverage.

regions are polygons as the sensing radii of all sensors are the same. ■

Example 10.6. *The performance of the proposed algorithms is now compared with the Minmax-point algorithm [99] which is an effective coverage strategy for non-identical sensors with uniform sensing priority. In this example,* 27 *sensors with a communication range of* 20m *are randomly deployed in the flat space described earlier: 15 with a sensing radius of 3m, 6 with a sensing radius of 2.5m, 3 with a sensing radius of 3.5m, and 3 with a sensing radius of 4.5m. Let the priority function be equal to* $\varphi(q) = exp(-k[(x_q - 25)^2 + (y_q - 25)^2]$*; the smaller* k *is the closer*

the sensing priority is to being uniform. In the special case, when $k = 0$ the sensing priority throughout the field is uniform. Fig. 10.13 shows the final weighted coverage for four different values of k: $k = 0$, 0.001, 0.01, 0.1. As it can be seen from this figure, the proposed algorithms outperform the Minmax-point algorithm when the target field is non-uniform (i.e., $k \neq 0$) and this superiority is more significant for a more non-uniform priority function. Note that when the target field is uniform, the sensors do not move under the MWV and MWP algorithms, and in this case the Minmax-point algorithm is more effective than these two. However, even in this case the performance of the MDW algorithm is better than that of the Minmax-point algorithm. ■

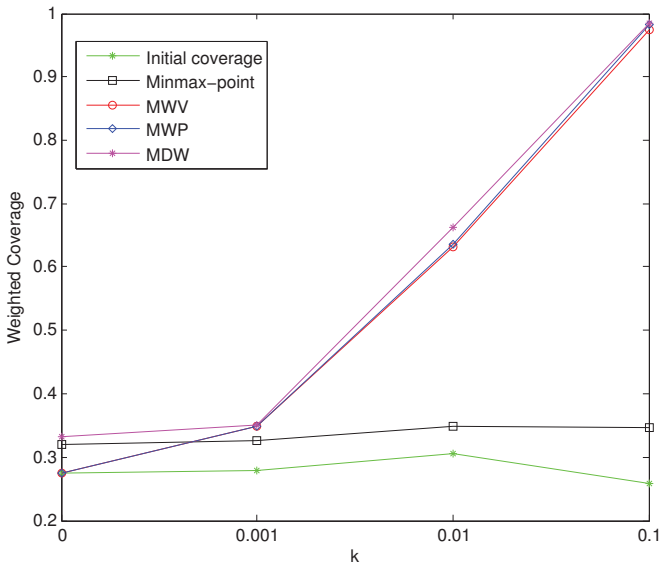


Figure 10.13: The weighted coverage for different values of k in Example 10.6.

Remark 10.5. *The overall performance of a coverage strategy highly depends on the specific application and network configuration in terms of the number of sensors, priority function, sensing range of the sensors and computational power of the mobile agents. In order to select the proper coverage strategy (which is done by the operator in the beginning), a number of issues should be taken into account. For example, as far as computational complexity is concerned, the MWV deployment*

strategy outperforms the other two techniques. On the other hand, when the priority function is more or less the same over the entire field the MDW strategy is more effective. If the priority function is highly non-uniform (e.g., it is very much focused in certain areas), then the proper choice of algorithm depends on the number of sensors and their sensing radii. Particularly, for a small number of sensors with small sensing radii and a highly non-uniform priority function, the MWP strategy outperforms the other two techniques.

Chapter 11

Distributed Deployment Algorithms for Efficient Coverage in a Network of Static and Mobile Sensors

This chapter proposes efficient schemes to increase sensing coverage in a network composed of both mobile and static sensors. The proposed deployment techniques properly assign a virtual weight to every point in the sensing field, based on the information received from the other sensors regarding their sensing radii, and the location of the static ones. The multiplicatively weighted Voronoi (MW-Voronoi) diagram is used to discover the coverage holes corresponding to different mobile sensors with different sensing ranges. According to the proposed strategies, the mobile sensors move out of the area covered by static sensors, to a point from where it can cover the coverage holes of the static sensors. As a result, under the proposed strategies coverage holes in the network are reduced. Simulation results are provided to demonstrate the effectiveness of the strategies developed in this chapter.

The plan of the rest of the chapter is as follows. The problem is defined in Section 11.1, where some important notations and assumptions are also presented. The proposed deployment algorithms are introduced in Section 11.2, as the main contribution of the chapter. Simulations are given in Section 11.3, which demonstrate the efficacy of the proposed deployment strategies.

11.1 Problem Formulation

Consider a group of n mobile and m static sensors randomly distributed in a field, and assume that the sensors have different sensing ranges, which are circles centered at the position of the sensors. It is desired that the mobile sensors change their location in a proper distributed manner such that the total covered area (by both mobile and static sensors) increases.

Represent each mobile sensor in the field as a node and sketch the corresponding MW-Voronoi regions for all mobile sensors to cover the entire sensing field. Recall from the characterization of the MW-Voronoi diagram that the nearest sensor to any point inside a MW-Voronoi region (in the sense of weighted distance) is the one inside it. Hence, if a mobile sensor cannot detect a certain point inside its corresponding region, that point cannot be detected by any other mobile sensor in the field either. Hence, in order to identify the coverage holes (i.e. the uncovered points in the field), it suffices that each mobile sensor checks its own MW-Voronoi region to find the points it cannot cover.

Notation 11.1. *In the remainder of this chapter, $\mathcal{V}$ denotes the MW-Voronoi diagram constructed based on the position and sensing radii of the mobile sensors only.*

Definition 11.1. *Consider a mobile sensor S_i with the sensing radius r_i and the corresponding MW-Voronoi region Π_i in $\mathcal{V}$, $i \in \mathbf{n}$. Let Q be an arbitrary point inside Π_i. The intersection of the region Π_i and a circle of radius r_i centered at Q*

is referred to as the i-th coverage area w.r.t. Q. *Note that this area can be covered by any mobile or static sensor. Part of the* i*-th coverage area w.r.t.* Q *which is not covered by any static sensor is referred to as the* i-th dynamic coverage area w.r.t. Q, *and is denoted by* $\lambda^{Q}_{\Pi_i}$. *The* i*-th dynamic coverage area w.r.t. the location* P_i *of the sensor* S_i *is called the dynamic local coverage area of that sensor. Also, the total covered area is denoted by* ψ, *and the part of* ψ *which is not covered by any static sensor will be referred to as the total dynamic coverage area. Let this area be denoted by* λ.

Definition 11.2. *Consider an arbitrary point* Q *inside the MW-Voronoi region* Π_i, $i \in \mathbf{n}$. *The region inside* Π_i *which is not covered by any static sensor and lies outside the* i*-th coverage area w.r.t.* Q *referred to as the* i-th coverage hole w.r.t. Q, *and is denoted by* $\theta^{Q}_{\Pi_i}$. *The* i*-th coverage hole w.r.t. the location* P_i *of the sensor* S_i *is called the local coverage hole of that sensor. Also, the union of all local coverage holes in the sensing field is referred to as the* total coverage hole, *and is denoted by* θ. *From the properties of the MW-Voronoi diagram it is straightforward to verify that* $\theta = \sum_{i=1}^{n} \theta^{P_i}_{\Pi_i}$.

11.2 Deployment Protocols

In this section, two efficient deployment strategies are presented for a distributed sensor network. First, every static sensor broadcasts its sensing radius and location to mobile sensors, and then each mobile sensor assigns a proper weight $\varphi(q)$ to every point in the field based on the received information. For a point q, the weight $\varphi(q)$ is a positive constant if and only if this point cannot be covered by any static sensor in the field. Otherwise, it is a negative amount whose absolute value depends on: (i) the number of static sensors that can cover q, and (ii) the distance between q and such static sensors. More precisely:

$$\varphi(q) = \begin{cases} -\sum_{i \in \mathbf{k}_q} f(q, \acute{r}_i, \acute{S}_i) & \text{if } q \text{ is covered by some static sensors,} \\ C & \text{otherwise} \end{cases}$$

where C is a positive constant, $\acute{S}_i$ and $\acute{r}_i$ are the position and radius of the i-th static sensor, respectively, and $\mathbf{k}_q$ is the set of all static sensors that cover the point q. Furthermore, $f(q, \acute{r}_i, \acute{S}_i)$ is an appropriate decreasing function of $d(q, \acute{S}_i)$ over $[0, \acute{r}_i]$ (e.g., a candidate example, $f = \acute{r}_i - d(q, \acute{S}_i)$). The following definition will prove useful in the presentation of the proposed algorithms.

Definition 11.3. *Consider a mobile sensor S_i with the sensing radius r_i and the corresponding MW-Voronoi region Π_i, $i \in \mathbf{n}$, and let X be an arbitrary point inside Π_i. The integral of the weight function $\varphi(.)$ over the intersection of the region Π_i and a circle of radius r_i centered at X, denoted by $C(X, r_i)$, is referred to as the* i-th weighted coverage w.r.t. X. *The mathematical characterization of the i-th weighted coverage w.r.t. X is as follows:*

$$\beta_{\Pi_i}^{X} = \int_{\Pi_i \cap C(X, r_i)} \varphi(q) dq \tag{11.2}$$

The weighted coverage w.r.t. the location P_i of the mobile sensor S_i is called the local weighted coverage of that sensor.

Once the weights are assigned to all points in the field, the proposed algorithms are performed iteratively. At each iteration, every mobile sensor first broadcasts its location and sensing radius to other mobile sensors, and then constructs its MW-Voronoi region based on the similar information it receives from other mobile sensors. Then, every mobile sensor finds its destination point in its MW-Voronoi region according to the deployment strategy of each algorithm (introduced later). Once the new target location $\acute{P}_i$ is calculated, both the weighted coverage and dynamic coverage area w.r.t. this location, i.e. $\beta_{\Pi_i}^{\acute{P}_i}$ and $\lambda_{\Pi_i}^{\acute{P}_i}$, are obtained. If this

weighted coverage is greater than the previous local weighted coverage and also dynamic coverage area is increased, i.e. $\beta_{\Pi_i}^{\acute{P}_i} > \beta_{\Pi_i}^{P_i}$ and $\lambda_{\Pi_i}^{\acute{P}_i} > \lambda_{\Pi_i}^{P_i}$, then the mobile sensor moves to the new destination; otherwise, it remains in its current position. Finally, when none of the sensors' weighted coverage or dynamic coverage area in its corresponding MW-Voronoi region would be increased by a certain level, there is no need to continue the iterations. In order to terminate the algorithm in finite time, a proper coverage improvement threshold ϵ is defined such that if the increase in the dynamic coverage area by none of the mobile sensors within its corresponding MW-Voronoi region exceeds ϵ in an iteration, then the algorithm terminates. Note that the algorithms introduced in this chapter are different only in the techniques used to find the destination point for each sensor. The following theorem is similar to Theorem 4.1, and shows that the total coverage increases under the proposed algorithms.

Theorem 11.1. *Let a set of m static sensors and n mobile sensors be randomly placed in a field in the 2D plane. Let the positions of the n mobile sensors be denoted by* $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ *with the corresponding MW-Voronoi regions* $\mathbf{\Pi} = \{\Pi_1, \Pi_2, \ldots, \Pi_n\}$. *Assume the sensors move to new positions* $\acute{\mathbf{P}} = \{\acute{P}_1, \acute{P}_2, \ldots, \acute{P}_n\}$ *with the corresponding MW-Voronoi regions* $\acute{\mathbf{\Pi}} = \left\{\acute{\Pi}_1, \acute{\Pi}_2, \ldots, \acute{\Pi}_n\right\}$ *such that* $\acute{P}_i \neq P_i$ *for all* $i \in \mathbf{K}$, *where* $\mathbf{K}$ *is a non-empty subset of* $\mathbf{n}$. *If the i-th dynamic coverage area w.r.t.* $\acute{P}_i$ *in the previously constructed MW-Voronoi region* Π_i *is greater than the i-th dynamic local coverage area in* Π_i *(i.e.* $\lambda_{\Pi_i}^{\acute{P}_i} > \lambda_{\Pi_i}^{P_i}$*) for all* $i \in \mathbf{K}$, *then the total coverage area in the network increases.*

Proof. The proof is similar to that of Theorem 4.1, and is omitted here. ■

Remark 11.1. *Note that since the increase of the weighted coverage area is a requirement in the above deployment protocol, thus mobile sensors tend to move out of the areas covered by static sensors (which have negative weights), and cover the*

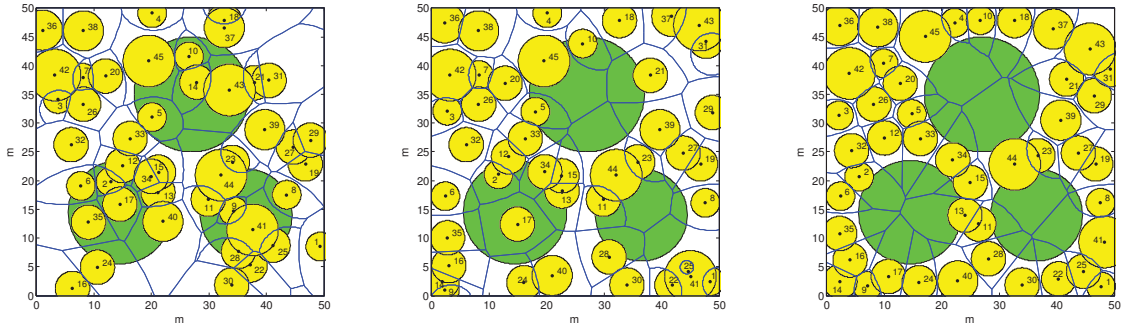
points that are not covered by static sensors (which have positive weights). Furthermore, according to Theorem 11.1, since the dynamic coverage area of the moved sensors increases under the above protocol, this guarantees that the total coverage area increases before all sensors stop moving.

The details of the proposed strategies will be presented in the next two subsections.

11.2.1 Farthest Weighted Vertex (FWV) Strategy

In this strategy, if all vertices of the i-th region have negative weight (i.e., all vertices can be covered by at least one static sensor), then S_i moves toward the vertex with minimum absolute value, up to the point from which that vertex is covered. If, on the other hand, there are one or more vertices with positive weights, then S_i moves toward the farthest one, denoted by $V_{i,fwv}$. Again, it continues moving up to the point from which it can cover that vertex. If the i-th region does not have any vertices, then S_i does not move and remains in its current position.

Fig. 11.1 shows an operational example of the FWV Algorithm. In this example, 45 mobile sensors are randomly placed in a 50m $\times$ 50m flat space: 25 with a sensing radius of 3m, 10 with a sensing radius of 2.5m, 5 with a sensing radius of 3.5m, and 5 with a sensing radius of 4.5m. There are also 3 static sensors with the sensing range of 8m, 9m and 10m. The communication range of the mobile and static sensors are assumed to be 20m and 40m, respectively. In this figure, three snapshots are provided, and in each one the sensing areas of both mobile sensors (yellow filled circles) and static sensors (green filled circles) are depicted. The MW-Voronoi diagram $\mathcal{V}$ is also depicted in the figure. The initial coverage in this setup is 58.29% (first snapshot), but after the first round it increases to 68.57% (second snapshot), and finally it reaches 80.22% (third snapshot). It can be observed from Fig. 11.1(c) that in the final round the mobile sensors are located out of the area



(a) (b) (c)

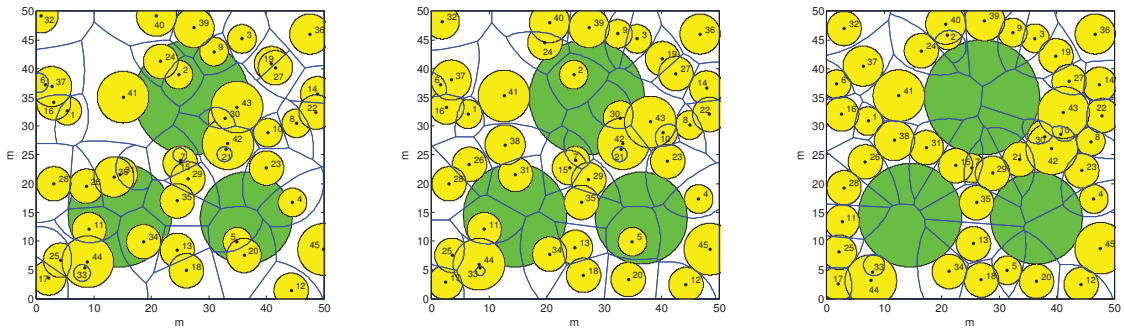
Figure 11.1: Snapshots of the execution of the FWV strategy for a network of 45 nonidentical sensors with random initial distribution. (a) Initial coverage; (b) coverage after the first round, and (c) final coverage.

covered by static sensors, and that the points they cover are not fully covered by static sensors.

11.2.2 Max-area Strategy

The Max-area is a MW-Voronoi-based coverage optimization approach which aims to locally maximize the weighted coverage of each sensor inside its own region [112]. Given an MW-Voronoi region and a disk-shaped sensing pattern of a sensor, Max-area strategy finds a point inside the region which if the sensor moves there, then the intersection of the weighted area of the region and the sensing disk is maximized. In the special case, if the radius of the sensing disk is sufficiently large, then the solution to this problem is the center of the smallest enclosing circle of the region. In addition to the small sensing radius, if the field is uniformly weighted, then the optimum point is the center of largest inscribed ball inside the region, which is known as the *Chebychev center of the region*.

In general, finding the optimum point inside the MW-Voronoi region is not straightforward, and an iterative nonlinear optimization approach may be used to find it. Such an algorithm considers the intersection area noted above as an objective function, and uses the gradient of this objective function to determine the moving



(a) (b) (c)

Figure 11.2: Snapshots of the execution of the Max-area strategy for a network of 45 nonidentical sensors with random initial distribution. (a) Initial coverage; (b) coverage after the first round, and (c) final coverage.

direction for the sensor (the objective function is guaranteed to increase if the sensor moves in that direction). In this optimization problem, the set of constraints is characterized by the boundaries of the region, and the gradient is computed iteratively to assess the proximity of the optimum point.

Consider the initial setting of Fig. 11.2(a), and let the Max-area strategy be employed. The results in this case are depicted in Figs. 11.2(b) and 11.2(c), where it is shown that after the first round the coverage increases from 59.47% to 69.51%, and that the final coverage is 79.33%. It can be observed from this figure that in the final round the mobile sensors are almost out of the area covered by the static sensors, and that at least part of the area each one covers is not reachable by any static sensor.

Theorem 11.2. *The proposed algorithms (FWV and Max-area) are convergent.*

Proof. The proof is similar to that of Theorem 5.4, and is omitted here. ■

Theorem 11.3. *Consider a set of m static and n mobile sensors randomly deployed in a 2D field. Using any of the proposed algorithms with the dynamic coverage improvement threshold ϵ, the number of required rounds for the termination of the algorithm is upper bounded by $\frac{A_{total}-A_{static}}{\epsilon}$, where A_{static} is the area covered by static*

sensors.

Proof. The proof is similar to that of Theorem 5.5, and is omitted here. ■

11.3 Simulation Results

Example 1: The two algorithms proposed in the previous section are applied to a flat space of size 50m × 50m. It is assumed that there are 3 static sensors with the sensing radii of 8m, 9m and 10m in the field. Assume also that a number of mobile sensors are randomly placed in the field. The communication range of the mobile and static sensors are assumed to be 20m and 40m, respectively. In each simulation, the algorithm terminates when none of the mobile sensors' dynamic coverage area in its corresponding MW-Voronoi region increases by more than 0.1m^2 or none of the sensors' weighted coverage increases if it makes another move. The results presented in this example for field coverage are all the average values obtained by using 20 random initial locations for the sensors.

Assume first there are 27 sensors: 15 with a sensing radius of 3m, 6 with a sensing radius of 2.5m, 3 with a sensing radius of 3.5m, and 3 with a sensing radius of 4.5m. The coverage factor (defined as the ratio of the covered area to the overall area) of the sensor network in each round is depicted in Fig. 11.3 for the two algorithms proposed in this chapter. As it can be seen from this figure, although the FWV strategy outperforms the Max-area strategy in the first few rounds, their final coverage is approximately the same.

It is desired now to compare the performance of the two algorithms in terms of the number of mobile sensors n. To this end, consider three more setups: n=18, 36 and 45, in addition to the previous setup. Let changes in the number of identical mobile sensors in the new setups be proportional to the changes in the total number of mobile sensors (e.g., for the case of n=18 there will be 10 mobile sensors with

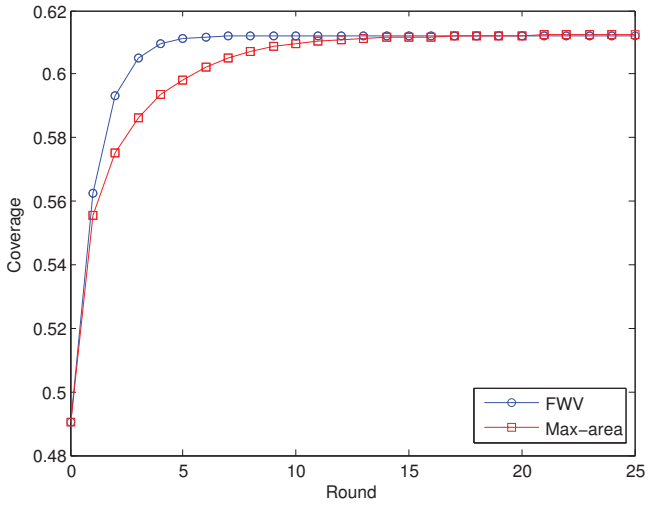


Figure 11.3: Network coverage per round for 27 mobile sensors.

a sensing radius of 3m, 4 with a sensing radius of 2.5m, 2 with a sensing radius of 3.5m, and 2 with a sensing radius of 4.5m). In Fig. 11.4, the final coverage of the algorithms is depicted for different number of sensors. It can be observed from this figure that the final coverage of both algorithms are approximately the same for various setups.

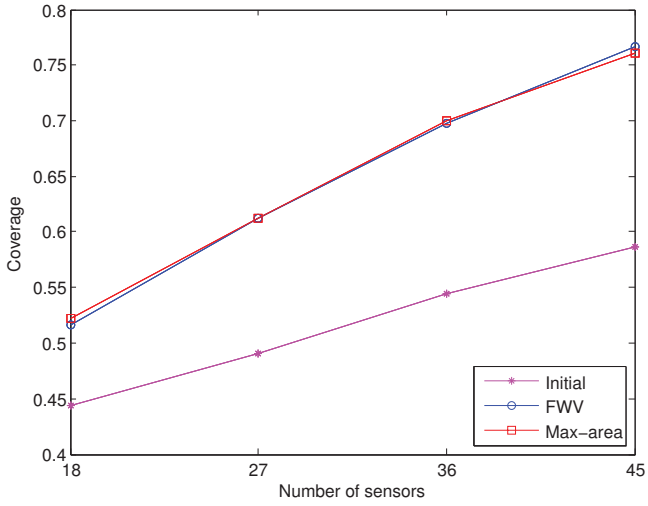


Figure 11.4: Network coverage for different number of sensors using the proposed algorithms.

Another important means of assessing the performance of sensor deployment

algorithms is the time it takes to reach the desired coverage level. This time depends on the number of rounds it takes for the sensors to provide a prescribed coverage level, as well as the sensor deployment time in each round. Thus, to compare the proposed methods in terms of deployment speed in reaching the desired coverage level, the stopping round and also the time duration of each round should be taken into consideration. As it can be seen from Fig 11.5, the number of rounds (required to meet a certain termination condition) is larger in the Max-area strategy than that in the FWV strategy. In addition, the sensor deployment time in each round for the Max-area strategy is larger than that for the FWV strategy. Therefore, the FWV algorithm is a good candidate for field coverage as far as the deployment time is concerned.

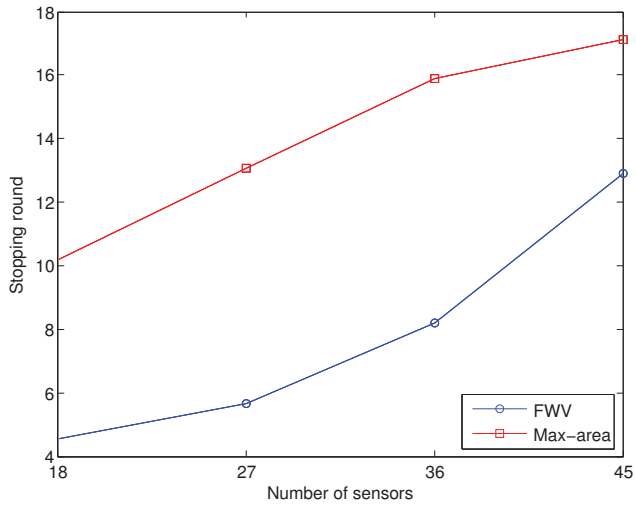


Figure 11.5: The number of rounds required to reach the termination conditions for different number of sensors using the proposed algorithms.

Another important factor in the performance evaluation of different algorithms is the energy consumption of the sensors, which is directly related to the moving distance of the sensors. It can be observed from Fig. 11.6 that the average moving distance of the Max-area strategy is smaller than that in the FWV strategy considerably. Hence, the Max-area algorithm is a better candidate for field coverage as far as the sensors' energy consumption is concerned.

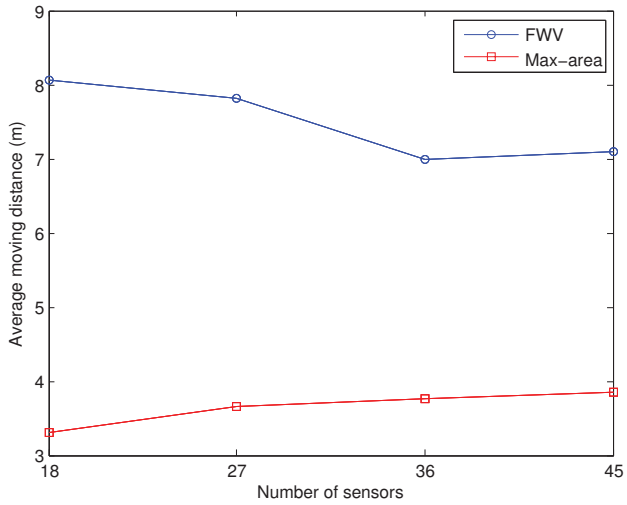
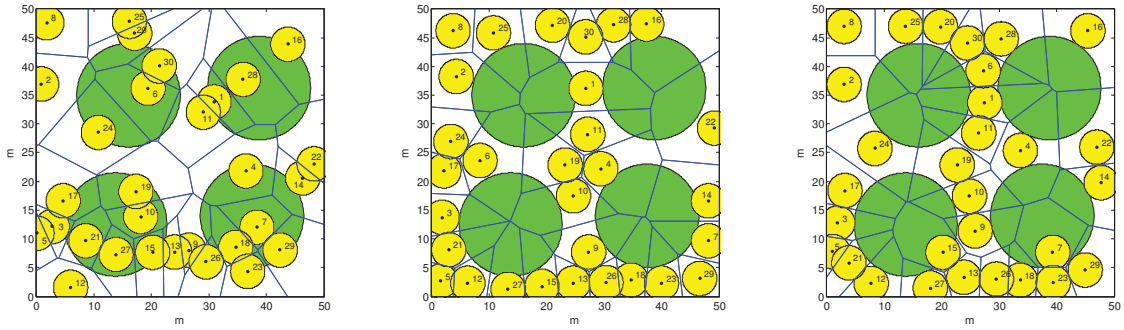


Figure 11.6: The average distance each mobile sensor travels for different number of sensors, using the proposed algorithms.

Example 2: In this example, 30 mobile sensors with the sensing range of 3m each are randomly placed in a 50m $\times$ 50m flat space. There are also 4 static sensors, each with a sensing range of 9m. The communication range of mobile and static sensors are assumed to be 20m and 40m, respectively. Fig. 11.7 shows an operational example of the FWV and Max-area strategies. Three snapshots are provided, and in each one sensing areas of the sensors (filled circles) as well as the Voronoi polygons are depicted. Since the sensing radii of all mobile sensors are the same, the regions are polygons, as in the conventional Voronoi diagram. The initial coverage in this setup is 54.63%, and the final coverage under the FWV and Max-area strategies is 71.35% and 71.44%, respectively. Note that the maximum possible coverage for this example is $\frac{30(9\pi)+4(81\pi)}{50^2} \times 100 = 74.64\%$. It can be observed from Figs. 11.7(b) and 11.7(c) that in the final round of both strategies at least part of the area covered by each mobile sensor is not covered by any static sensor, as noted before.



(a) (b) (c)

Figure 11.7: Snapshots of the execution of the proposed strategies. (a) Initial coverage; (b) final coverage under the FWV strategy, and (c) final coverage under the Max-area strategy.

Chapter 12

Conclusions

12.1 Summary

The results developed in this dissertation can be summarized as follows.

An algorithm is proposed in Chapter 2 to solve a constrained optimization concerning a mobile sensor network. The cost function takes into account the power consumption of the entire network, in order to accomplish the target monitoring objective efficiently. A strategy is also provided to maximize the durability of the sensors by monitoring the residual energy of every sensor, and then adjusting their parameters and relocating them accordingly. The proposed relocation scheme in this case ensures a uniform consumption of the remaining energy of each sensor, such that all sensors run out of energy at the same time. The algorithm guarantees end-to-end connectivity from the target to the fixed access point, which is crucial in order to monitor a moving target. Simulation results illustrate the efficacy of the proposed techniques.

A novel energy-efficient tracking technique is proposed in Chapter 3 for wireless mobile sensor networks. The field is first divided into a grid, and is then mapped into a graph. Proper weights are subsequently assigned to the edges of the graph to

model the energy consumption due to sensing, communication and movement, as the main sources of energy expenditure in this type of network. The problem of finding a proper route and selecting the corresponding sensor locations for an energy-efficient tracking is translated to the well-known shortest path problem. This is carried out by partitioning the field into Voronoi polygons and investigating different scenarios in terms of network configuration. Simulations demonstrate the effectiveness of the proposed tracking strategy.

Novel sensor deployment strategies are proposed in Chapter 4 for efficient field coverage in a mobile sensor network. Based on these strategies, each sensor moves iteratively in a direction that the coverage holes in the corresponding Voronoi polygon are reduced. The proposed strategies tend to place the sensors in the plane in such a way that undesirable network configurations are avoided. To this end, the Maxmin-vertex strategy selects each sensor's target location as a point inside the corresponding Voronoi polygon whose distance from the nearest Voronoi vertex is maximized. The Minmax-edge strategy, on the other hand, selects this target location as a point inside the corresponding Voronoi polygon whose distance from the farthest Voronoi edge is minimized. The Maxmin-edge strategy selects the target location as a point inside the corresponding Voronoi polygon whose distance from the nearest Voronoi edge is maximized. Finally, the VEDGE strategy is a combination of the Minimax and Maxmin-edge algorithms. Two target points are calculated for each sensor based on these two methods, and the one which provides better coverage is selected as the target location for that sensor. In all of these techniques, each sensor moves to the new location only if its coverage increases. Simulations demonstrate the advantages of the proposed techniques compared with other known methods.

Efficient sensor deployment algorithms are presented in Chapter 5 for increasing sensing coverage in a network of mobile sensors with different sensing radii. The

multiplicatively weighted Voronoi (MW-Voronoi) diagram is used to partition the field. Three strategies, namely weighted vector boundary (WVB), Minmax-curve and Maxmin-curve are subsequently developed to find appropriate locations for the sensors to increase sensing coverage. These locations are determined such that they are neither too close to each other and the boundaries of their MW-Voronoi regions nor too far from them. Using the proposed algorithms, the sensors move iteratively such that the coverage hole in the field decreases monotonically. Simulations with different number of sensors are provided for comparison.

Chapter 6 presents efficient sensor deployment algorithms to improve coverage in mobile sensor networks. It is assumed that the sensing radii of different sensors are not the same. A multiplicatively weighted Voronoi (MW-Voronoi) diagram is then employed to develop three distributed deployment algorithms accordingly. Using these algorithms, the sensors move iteratively to reduce coverage holes in the sensing field. The algorithms proposed here take the general characteristics of an ideal sensor configuration into account (e.g., each sensor should not be too far or too close to any of the vertices of its corresponding MW-Voronoi region). Simulation results are pretested to compare the performance of the proposed approaches for different number of sensors.

Three distributed deployment algorithms are proposed in Chapter 7 to increase coverage in a mobile sensor network. The sensing field is first partitioned using the Voronoi diagram, and the deployment algorithms are developed based on the configuration of Voronoi polygons. The algorithms are iterative, where in each iteration the next candidate position of any sensor is obtained based on the distance of the sensor from the edges and vertices of its polygon. Different virtual forces are defined which are applied to the sensor from the vertices and boundaries of the polygon. Each sensor tends to move to a new location under the vector sum of these virtual forces, but it only moves to the new location if its coverage increases.

The results are extended to the case of sensors with nonidentical sensing ranges using the notion of the multiplicatively-weighted Voronoi (MW-Voronoi) diagram. Simulations confirm the efficacy of the proposed algorithms in increasing the network coverage.

In Chapter 8, two efficient distributed sensor relocation techniques are proposed to increase sensing coverage in a mobile sensor network. The general case of a network of nonidentical sensors is considered, where the sensing radii of nodes are different. Each sensor has a limited communication range, which prevents them from collecting the required information for constructing MW-Voronoi regions. The notion of limited communication MW-Voronoi (LCMW-Voronoi) diagram was introduced and two efficient strategies, namely the LCFP and LCMP algorithms, were developed to relocate the sensors in such a way that network coverage increases. The LCFP strategy finds a candidate point based on the farthest point from the sensor in the corresponding LCMW-Voronoi region, while the candidate point in the LCMP strategy is obtained such that the distance of the sensor from the farthest point in the region is minimized. Simulations demonstrate the efficiency of the proposed techniques.

In Chapter 9, two efficient distributed sensor relocation techniques are proposed to increase field coverage of mobile sensor networks. The algorithms are applicable to networks having non-identical mobile sensors and target coverage fields with obstacles. To account for the existence of obstacles, an extension of MW-Voronoi diagram, namely visibility-aware MW-Voronoi (VMW-Voronoi) diagram has been introduced as a tool to allow enhancement in sensor's coverage area. The iterative implementation of the algorithms provides gradual maximization of the overall network coverage. Simulation results confirm the effectiveness of the proposed techniques for different number of sensors.

Efficient sensor deployment algorithms are presented in Chapter 10 to increase

coverage in a mobile sensor network with a prescribed priority assignment for different points in the sensing field. It is assumed that the sensors are not identical in terms of sensing capabilities. The multiplicatively weighted Voronoi (MW-Voronoi) diagram is then employed to develop three distributed deployment strategies. According to the proposed algorithms, each sensor moves iteratively in such a way that the prioritized uncovered area in its MW-Voronoi region is reduced. All proposed algorithms consider the relative priority of the points inside each region (or on its vertices). One of these strategies also takes the distances of each sensor and the points inside its MW-Voronoi region into account. Simulations are presented to compare the performance of the coverage algorithms developed in this chapter.

Two sensor deployment strategies are introduced in Chapter 11 to increase the sensing coverage in a network of mobile and static sensors. The problem is addressed in the most general case, where the sensing radii of different sensors are not the same. A multiplicatively weighted Voronoi (MW-Voronoi) diagram is then employed to develop two distributed deployment algorithms. According to the proposed algorithms, each mobile sensor assigns a proper weight to every point in the field, based on the information it receives from static sensors. The mobile sensors then move iteratively to proper locations out of the covered area of static sensors, in such a way that coverage holes of the network are reduced. Simulations are presented to compare the performance of the proposed approaches for different number of sensors in the network. It is shown that the Max-area strategy outperforms the other method as far as the energy consumption is concerned. On the other hand, the FWV strategy is more efficient in terms of convergence rate.

12.2 Suggestions for Future Work

In what follows, some of the possible extensions of the results of this dissertation as well as some relevant problems for future study are presented.

- The algorithms proposed in Chapters 2 and 3 do not consider the power consumption model of sensor batteries. One can develop a variation of these algorithms by considering an appropriate sensor battery model for more power-efficient deployment strategies, which would be of more practical interest.

- In all of the proposed algorithms given in Chapters 4-11, it is assumed that the sensing area of each sensor is uniform and circular. Developing proper strategies to maximize the coverage of the network for non-circular, non-uniform and probabilistic sensing patterns is another possible extension.

- The results of Chapters 4-11 are developed for ideal communication links without taking transmission delays and link failures into account. One possible future work is to consider such practical problems and investigate their impact on the performance of the deployment algorithms. The results can then be used to develop more reliable sensor deployment techniques for a practical environment.

- The algorithms developed in this dissertation assume the sensors move in a 2D plane. As a natural extension of this work, it would be interesting to study the problem when sensors move in 3D space.

- The objective of the strategies developed in Chapters 4-11 is to obtain maximum field coverage by properly deploying sensor nodes. Modifying the proposed algorithms to maximize the lifetime of the network while increasing the sensing coverage would be another important direction for future work.

Bibliography

[1] W. Liang and Y. Liu, "Online data gathering for maximizing network lifetime in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 1, pp. 2–11, 2007.

[2] K. H. Low, W. K. Leow, and M. H. Ang, "Autonomic mobile sensor network with self-coordinated task allocation and execution," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 36, no. 3, pp. 315–327, 2006.

[3] I. Hussein and D. Stipanovic, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 642–657, 2007.

[4] V. Gupta, T. Chung, B. Hassibi, and R. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.

[5] S. Martinez, "Distributed interpolation schemes for field estimation by mobile sensor networks," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 491–500, 2010.

[6] S. Sudarsan, V. Subramanian, K. Yoshigoe, S. Ramaswamy, R. Seker, and R. B. Lenin, "Impact of duty cycle variation on WSNs," in *Proceedings of*

3rd International Conference on New Technologies, Mobility and Security (NTMS), 2009, pp. 1–5.

[7] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[8] R. Tharmarasa, T. Kirubarajan, and M. L. Hernandez, "Large-scale optimal sensor array management for multitarget tracking," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 5, pp. 803–814, 2007.

[9] Z. Liang, W. A. Chaovalitwongse, A. D. Rodriguez, D. E. Jeffcoat, D. A. Grundel, and J. K. O'Neal, "Optimization of spatiotemporal clustering for target tracking from multisensor data," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 2, pp. 176–188, 2010.

[10] G. Hoffmann and C. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 32–47, 2010.

[11] Y. Wang and C.-H. Wu, "Robot-assisted sensor network deployment and data collection."

[12] L. Miao, H. Qi, and F. Wang, "Self-deployable mobile sensor networks for on-demand surveillance," in *Proceedings of International Society for Optical Engineering*, vol. 5778, no. PART I, 2005, pp. 496–507.

[13] G. J. A. Van Dijk and M. G. Maris, "Wireless sensor network for mobile surveillance systems," in *Proceedings of International Society for Optical Engineering*, vol. 5611, 2004, pp. 185–191.

[14] W.-T. Chen, P.-Y. Chen, W.-S. Lee, and C.-F. Huang, "Design and implementation of a real time video surveillance system with wireless sensor networks," in *Proceedings of IEEE Vehicular Technology Conference*, 2008, pp. 218–222.

[15] S. Susca, F. Bullo, and S. Martinez, "Monitoring environmental boundaries with a robotic sensor network," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 288–296, 2008.

[16] H.-C. Lin, Y.-C. Kan, and Y.-M. Hong, "The comprehensive gateway model for diverse environmental monitoring upon wireless sensor network," *IEEE Sensors Journal*, vol. 11, no. 5, pp. 1293–1303, 2011.

[17] D. Dardari, A. Conti, C. Buratti, and R. Verdone, "Mathematical evaluation of environmental monitoring estimation error through energy-efficient wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 7, pp. 790–802, 2007.

[18] X. Wang, J. Ma, S. Wang, and D. Bi, "Distributed energy optimization for target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 73–86, 2010.

[19] W.-P. Chen, J. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 258–271, 2004.

[20] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proceedings of 6th International Symposium on Information Processing in Sensor Networks*, 2007, pp. 254–263.

[21] J. Meyer, R. Bischoff, G. Feltrin, and M. Motavalli, "Wireless sensor networks for long-term structural health monitoring," *Smart Structures and Systems*, vol. 6, no. 3, pp. 263–275, 2010.

[22] Y. Zou and K. Chakrabarty, "Distributed mobility management for target tracking in mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 872–887, 2007.

[23] C.-Y. Lin, W.-C. Peng, and Y.-C. Tseng, "Efficient in-network moving object tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, pp. 1044–1056, 2006.

[24] S. J. Geoghegan, G. McCorkle, C. Robinson, G. Fundyler, S. Ramaswamy, M. Tudoreanu, R. Seker, J. Brown, and M. Itmi, "A multi-agent system architecture for cooperative maritime networks," in *Proceedings of IEEE International Systems Conference*, 2009, pp. 7–12.

[25] S. J. Geoghegan, G. McCorkle, C. Robinson, G. Fundyler, S. Ramaswamy, J. Brown, and M. Itmi, "A P2P, agent-based system of systems architecture for cooperative maritime networks," in *Proceedings of 6th IEEE Consumer Communications and Networking Conference (CCNC)*, 2009, pp. 1–2.

[26] M. Johansson and L. Xiao, "Cross-layer optimization of wireless networks using nonlinear column generation," *IEEE Transactions on Wireless Communications*, vol. 5, no. 2, pp. 435–445, 2006.

[27] R. Madan, S. Cui, S. Lall, and A. Goldsmith, "Cross-layer design for lifetime maximization in interference-limited wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3142–3152, 2006.

[28] S. Shenker, "Fundamental design issues for the future Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1176–1188, 1995.

[29] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.

[30] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 104–116, 2005.

[31] X. Lin, N. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1452–1463, 2006.

[32] S. Martinez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[33] C.-F. Wang and S.-C. Huang, "Efficient deployment algorithms for prolonging network lifetime and ensuring coverage in wireless sensor networks," in *Proceedings of 8th International Conference on Intelligent Systems Design and Applications*, 2008, pp. 196–201.

[34] C.-E. Weng, J.-K. Lain, J.-M. Zhang, and J.-H. Wen, "Ueeda: uniform and energy-efficient deployment algorithm for wireless sensor networks," *International Journal of Communication Systems*, vol. 21, no. 5, pp. 453–467, 2008.

[35] L. Arienzo, "An information-theoretic approach for energy-efficient collaborative tracking in wireless sensor networks," *Eurasip Journal on Wireless Communications and Networking*, 2010.

[36] M. Mansouri, H. Snoussi, and C. Richard, "Robust target tracking with quantized proximity sensors," in *Proceedings of 5th IEEE International Symposium on Wireless Pervasive Computing*, 2010, pp. 278–282.

[37] Y. Jin, L. Wang, J.-Y. Jo, Y. Kim, M. Yang, and Y. Jiang, "EECCR: An energy-efficient m-coverage and n-connectivity routing algorithm under border effects in heterogeneous sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1429–1442, 2009.

[38] J. Cortes, S. Martinez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM. Control, Optimization and Calculus of Variations*, vol. 11, pp. 691–719, 2005.

[39] A. Boukerche and X. Fei, "A Voronoi approach for coverage protocols in wireless sensor networks," in *Proceedings of IEEE Global Communications Conference*, 2007, pp. 5190–5194.

[40] A. Kwok and S. Martinez, "Unicycle coverage control via hybrid modeling," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 528–532, 2010.

[41] A. Konstantinidis, K. Yang, and Q. Zhang, "An evolutionary algorithm to a multi-objective deployment and power assignment problem in wireless sensor networks," in *Proceedings of IEEE Global Communications Conference*, 2008, pp. 475–480.

[42] J. Cortes and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2006.

[43] T. M. Cavalier, W. A. Conner, E. del Castillo, and S. I. Brown, "A heuristic algorithm for minimax sensor location in the plane," *European Journal of Operational Research*, vol. 183, no. 1, pp. 42–55, 2007.

[44] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.

[45] M. Pavone, E. Frazzoli, and F. Bullo, "Distributed policies for equitable partitioning: Theory and applications," in *Proceedings of 47th IEEE Conference on Decision and Control*, 2008, pp. 4191–4197.

[46] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *Proceedings of 47th IEEE Conference on Decision and Control*, 2008, pp. 3947–3952.

[47] R. Graham and J. Cortes, "Asymptotic optimality of multicenter voronoi configurations for random field estimation," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 153–158, 2009.

[48] W. Li and C. G. Cassandras, "Distributed cooperative coverage control of sensor networks," in *Proceedings of the IEEE Conference on Decision and Control, and the European Control Conference*, vol. 2005, 2005, pp. 2542–2547.

[49] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *European Journal of Control*, vol. 11, no. 4-5, pp. 436–463, 2005.

[50] M. Zhong and C. G. Cassandras, "Distributed coverage control and data collection with mobile sensor networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2445–2455, 2011.

[51] M. S. Stankovic, K. H. Johansson, and D. M. Stipanovic, "Distributed seeking of nash equilibria in mobile sensor networks," in *Proceedings of the IEEE Conference on Decision and Control*, 2010, pp. 5598–5603.

[52] M. Zhong and C. G. Cassandras, "Asynchronous distributed optimization with event-driven communication," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2735–2750, 2010.

[53] C. Zhang, Y. Zhang, and Y. Fang, "A coverage inference protocol for wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 850–864, 2010.

[54] W. Choi, G. Ghidini, and S. K. Das, "A novel framework for energy-efficient data gathering with random coverage in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 8, no. 4, 2012.

[55] S. He, J. Chen, and Y. Sun, "Coverage and connectivity in duty-cycled wireless sensor networks for event monitoring," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 475–482, 2012.

[56] Y. Xiao, H. Chen, K. Wu, B. Sun, Y. Zhang, X. Sun, and C. Liu, "Coverage and detection of a randomized scheduling algorithm in wireless sensor networks," *IEEE Transactions on Computers*, vol. 59, no. 4, pp. 507–521, 2010.

[57] R. MacHado, W. Zhang, G. Wang, and S. Tekinay, "Coverage properties of clustered wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 7, no. 2, 2010.

[58] S. He, J. Chen, X. Li, X. Shen, and Y. Sun, "Leveraging prediction to improve the coverage of wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, pp. 701–712, 2012.

[59] X. Bai, Z. Yun, D. Xuan, T. H. Lai, and W. Jia, "Optimal patterns for four-connectivity and full coverage in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 3, pp. 435–448, 2010.

[60] C. Liu and G. Cao, "Spatial-temporal coverage optimization in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 4, pp. 465–478, 2011.

[61] C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," vol. 91, no. 8, 2003, pp. 1247–1256.

[62] X. Wang, X. Lin, Q. Wang, and W. Luan, "Mobility increases the connectivity of wireless networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 440–454, 2013.

[63] M. Grossglauser and D. N. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, 2002.

[64] G. Wang, G. Cao, and T. F. L. Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

[65] G. Wang, G. Cao, P. Berman, and T. F. L. Porta, "A bidding protocol for deploying mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 563–576, 2007.

[66] M. Ma and Y. Yang, "Adaptive triangular deployment algorithm for unattended mobile sensor networks," *IEEE Transactions on Computers*, vol. 56, no. 7, pp. 946–958, 2007.

[67] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE Transactions on Mobile Computing*, vol. 7, no. 2, pp. 262–274, 2008.

[68] C.-Y. Chang, C.-T. Chang, Y.-C. Chen, and H.-R. Chang, "Obstacle-resistant deployment algorithms for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 6, pp. 2925–2941, 2009.

[69] Z. Shen, Y. Chang, H. Jiang, Y. Wang, and Z. Yan, "A generic framework for optimal mobile sensor redeployment," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 8, pp. 4043–4057, 2010.

[70] S. Yang, M. Li, and J. Wu, "Scan-based movement-assisted sensor deployment methods in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 8, pp. 1108–1121, 2007.

[71] Z. Jing and Z. Jian-Chao, "A virtual centripetal force-based coverage- enhancing algorithm for wireless multimedia sensor networks," *IEEE Sensors Journal*, vol. 10, no. 8, pp. 1328–1334, 2010.

[72] W. Wang, V. Srinivasan, and K.-C. Chua, "Coverage in hybrid mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 11, pp. 1374–1387, 2008.

[73] S. Chellappan, W. Gu, X. Bai, D. Xuan, B. Ma, and K. Zhang, "Deploying wireless sensor networks under limited mobility constraints," *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, pp. 1142–1157, 2007.

[74] S. Chellappan, X. Bai, B. Ma, D. Xuan, and C. Xu, "Mobility limited flip-based sensor networks deployment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 2, pp. 199–211, 2007.

[75] X. Wang, S. Han, Y. Wu, and X. Wang, "Coverage and energy consumption control in mobile heterogeneous wireless sensor networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 975–988, 2013.

[76] C.-H. Wu and Y.-C. Chung, "Heterogeneous wireless sensor network deployment and topology control based on irregular sensor model," in *Proceedings of 2nd International Conference in Grid and Pervasive Computing*, 2007, pp. 78–88.

[77] Y. Stergiopoulos and A. Tzes, "Convex Voronoi-inspired space partitioning for heterogeneous networks: A coverage-oriented approach," *IET Control Theory and Applications*, vol. 4, no. 12, pp. 2802–2812, 2010.

[78] X. Xu and S. Sahni, "Approximation algorithms for sensor deployment," *IEEE Transactions on Computers*, vol. 56, no. 12, pp. 1681–1695, 2007.

[79] N. Bartolini, T. Calamoneri, T. La Porta, C. Petrioli, and S. Silvestri, "Sensor activation and radius adaptation (SARA) in heterogeneous sensor networks," *ACM Transactions on Sensor Networks*, vol. 8, no. 3, 2012.

[80] H. M. Ammari and S. K. Das, "Scheduling protocols for homogeneous and heterogeneous k-covered wireless sensor networks," *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 79–97, 2011.

[81] Q. Wang, K. Xu, G. Takahara, and H. Hassanein, "Device placement for heterogeneous wireless sensor networks: Minimum cost with lifetime constraints," *IEEE Transactions on Wireless Communications*, vol. 6, no. 7, pp. 2444–2453, 2007.

[82] Y. Lin, J. Zhang, H. S.-H. Chung, W. H. Ip, Y. Li, and Y.-H. Shi, "An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 3, pp. 408–420, 2012.

[83] A. M. Khedr and W. Osamy, "Minimum connected cover of a query region in heterogeneous wireless sensor networks," *Information Sciences*, vol. 223, pp. 153–163, 2013.

[84] R. Machado, N. Ansari, G. Wang, and S. Tekinay, "Adaptive density control in heterogeneous wireless sensor networks with and without power management," *IET Communications*, vol. 4, no. 7, pp. 758–767, 2010.

[85] M. Noori and M. Ardakani, "Design of heterogeneous sensor networks with lifetime and coverage considerations," *IEEE Wireless Communications Letters*, vol. 1, no. 3, pp. 193–196, 2012.

[86] A. M. Khedr and W. Osamy, "Minimum perimeter coverage of query regions in a heterogeneous wireless sensor network," *Information Sciences*, vol. 181, no. 15, pp. 3130–3142, 2011.

[87] S. Kumar, T. H. Lai, M. E. Posner, and P. Sinha, "Maximizing the lifetime of a barrier of wireless sensors," *IEEE Transactions on Mobile Computing*, vol. 9, no. 8, pp. 1161–1172, 2010.

[88] J. Jia, J. Chen, G. Chang, Y. Wen, and J. Song, "Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius," *Computers and Mathematics with Applications*, vol. 57, no. 11-12, pp. 1767–1775, 2009.

[89] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stoimenovic, "Localized sensor area coverage with low communication overhead," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 661–671, 2008.

[90] E. Deza and M. M. Deza, *Encyclopedia of Distances*. Springer, 2009.

[91] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, 2000.

[92] H. Tan, Y. Wang, X. Hao, Q.-S. Hua, and F. Lau, "Arbitrary obstacles constrained full coverage in wireless sensor networks," in *Proceedings of 5th International Conference of Wireless Algorithms, Systems and Applications*, 2010, pp. 1–10.

[93] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, "Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks," in *Proceedings of 1st International Conference on Wireless Internet*, 2005, pp. 114–121.

[94] H. Mahboubi, A. Momeni, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "An efficient target monitoring scheme with controlled node mobility for

sensor networks," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 6, pp. 1522–1532, 2012.

[95] ——, "Optimal target tracking strategy with controlled node mobility in mobile sensor networks," in *Proceedings of American Control Conference*, 2010, pp. 2921–2928.

[96] H. Mahboubi, W. Masoudimansour, A. G. Aghdam, and K. Sayrafian-Pour, "Cost-efficient routing with controlled node mobility in sensor networks," in *Proceedings of IEEE Multiconference on Systems and Control*, 2011, pp. 1238–1243.

[97] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Distributed deployment algorithms for improved coverage in mobile sensor networks," in *Proceedings of IEEE Multiconference on Systems and Control*, 2011, pp. 1244–1249.

[98] ——, "Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 163–174, 2014.

[99] ——, "Self-deployment algorithms for coverage problem in a network of mobile sensors with unidentical sensing range," in *Proceedings of IEEE Global Communications Conference*, 2010, pp. 1–6.

[100] H. Mahboubi, K. Moezzi, A. G. Aghdam, and K. Sayrafian-Pour, "Self-deployment algorithms for field coverage in a network of nonidentical mobile sensors," in *Proceedings of IEEE International Conference on Communications*, 2011, pp. 1–6.

[101] ——, "Self-deployment algorithms for field coverage in a network of nonidentical mobile sensors: Vertex-based approach," in *Proceedings of American Control Conference*, 2011, pp. 3227–3232.

[102] H. Mahboubi and A. G. Aghdam, "Distributed deployment strategies to increase coverage in a network of wireless mobile sensors," in *Proceedings of American Control Conference*, 2013, pp. 5907–5912.

[103] ——, "Self-deployment algorithms for coverage improvement in a network of nonidentical mobile sensors with limited communication ranges," in *Proceedings of American Control Conference*, 2013, pp. 6898–6903.

[104] H. Mahboubi, J. Habibi, A. G. Aghdam, and K. Sayrafian-Pour, "Cooperative self-deployment strategies in a mobile sensor network with prioritized coverage plan," in *Proceedings of IEEE Global Communications Conference*, 2011, pp. 1–6.

[105] ——, "Distributed deployment strategies for improved coverage in a network of mobile sensors with prioritized sensing field," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 451–461, 2013.

[106] ——, "Distributed coverage optimization in a network of static and mobile sensors," in *Proceedings of American Control Conference*, 2013, pp. 6877–6881.

[107] H. Mahboubi, W. Masoudimansour, A. G. Aghdam, and K. Sayrafian-Pour, "Maximum life span strategy for target tracking in mobile sensor networks," in *Proceedings of American Control Conference*, 2012, pp. 5096–5101.

[108] H. Mahboubi, F. Sharifi, A. G. Aghdam, and Y. M. Zhang, "Distributed coordination of multi-agent systems for coverage problem in presence of obstacles," in *Proceedings of American Control Conference*, 2012, pp. 5252–5257.

[109] H. Mahboubi, F. Sharifi, A. G. Aghdam, and Y. Zhang, "Distributed coordination of a network of nonidentical agents with limited communication capabilities in the presence of fixed obstacles," in *Proceedings of American Control Conference*, 2013, pp. 5418–5423.

[110] H. Mahboubi and A. G. Aghdam, "An energy-efficient strategy to improve coverage in a network of wireless mobile sensors with nonidentical sensing ranges," in *Proceedings of IEEE Vehicular Technology Conference*, 2013, pp. 1–5.

[111] W. Masoudimansour, H. Mahboubi, A. G. Aghdam, and K. Sayrafian-Pour, "Maximum lifetime strategy for target monitoring in a mobile sensor network with obstacles," in *Proceedings of 51th IEEE Conference on Decision and Control*, 2012, pp. 1404–1410.

[112] J. Habibi, H. Mahboubi, and A. G. Aghdam, "A nonlinear optimization approach to coverage problem in mobile sensor networks," in *Proceedings of 50th IEEE Conference on Decision and Control*, 2011, pp. 7255–7261.

[113] ——, "Distributed coverage optimization in a network of mobile agents subject to measurement error," in *Proceedings of American Control Conference*, 2012, pp. 4510–4515.

[114] M. Tousi, A. Ajorlou, H. Mahboubi, and A. G. Aghdam, "Decentralized pole-placement using generalized sampled-data hold functions," in *Proceedings of 51th IEEE Conference on Decision and Control*, 2012, pp. 6921–6925.

[115] V. Marbukh, K. Sayrafian-pour, H. Mahboubi, A. Momeni, and A. G. Aghdam, "Cooperative sensor relocation in a mobile sensor network by distributed subgradient algorithm," in *Proceedings of International Conference on Sensor Technologies and Applications*, 2011, pp. 91–96.

[116] ——, "Towards evolutionary-pricing framework for mobile sensor network self-organization," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2010, pp. 4394–4401.

[117] F. Sharifi, Y. M. Zhang, H. Mahboubi, and A. G. Aghdam, "Coverage control in multi-agent systems subject to communication delays," in *Proceedings of IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications*, 2012, pp. 267–274.

[118] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. S. G. Lee, "Deployment strategy for mobile robots with energy and timing constraints," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2005, pp. 2816–2821.

[119] G. J. Pottie and W. J. Kaiser, *Wireless Integrated Network Sensors*. ACM New York, NY, USA, 2000.

[120] D. K. Goldenberg, J. Lin, A. S. Morse, B. E. Rosen, and Y. R. Yang, "Towards mobility as a network control primitive," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing*, 2004, pp. 163–174.

[121] P. Coelho and U. Nunes, "Path-following control of mobile robots in presence of uncertainties," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 252–261, 2005.

[122] D. S. Mitrinović, J. E. Pecarić, and A. M. Fink, *Classical and New Inequalities in Analysis*. Kluwer Academic Publishers, Dordrecht, 1993.

[123] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," *SIAM review*, vol. 41, pp. 637–676, 1999.

[124] R. Klein, *Concrete and Abstract Voronoi Diagrams*. Springer, 1989.

[125] D. E. Koditschek, *Robot Planning and Control via Potential Functions.* MIT Press Cambridge, MA, USA, 1989.

[126] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of IEEE INFOCOM. 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2003, pp. 1734–1743.

[127] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of 6th Annual International Conference on Mobile Computing and Networking*, 2000, pp. 56–67.

[128] A. Kwok and S. Martinez, "A distributed deterministic annealing algorithm for limited-range sensor coverage," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 4, pp. 792–804, 2011.

[129] Y.-C. Wang and Y.-C. Tseng, "Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1280–1294, 2008.

[130] S. Yoon, O. Soysal, M. Demirbas, and C. Qiao, "Coordinated locomotion and monitoring using autonomous mobile sensor nodes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 10, pp. 1742–1756, 2011.

[131] S. P. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[132] A. Howard, M. J. Matarić, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots*, vol. 13, no. 2, pp. 113–126, 2002.

[133] J. Luo, D. Wang, and Q. Zhang, "On the double mobility problem for water surface coverage with mobile sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 146–159, 2012.

[134] X. Li, H. Frey, N. Santoro, and I. Stojmenovic, "Strictly localized sensor self-deployment for optimal focused coverage," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1520–1533, 2011.

[135] M. Rahimi, H. Shah, G. S. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, 2003, pp. 19–24.

[136] A. V. Akopyan and A. A. Zaslavsky, *Geometry of Conics*. American Mathematical Society, 2007.

[137] C. Rentel and T. Kunz, "A mutual network synchronization method for wireless ad hoc and sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 633–646, 2008.

[138] P. Ballal and F. Lewis, "Introduction to Crossbow Mica2 Sensors," Course Note, University of Texas at Arlington, 2007.

[139] P. Dutta, J. Taneja, J. Jeong, X. Jiang, and D. Culler, "A building block approach to sensornet systems," in *Proceedings of 6th ACM Conference on Embedded Network Sensor Systems*, 2008, pp. 267–280.