

國立臺灣大學資訊管理研究所

博士論文

指導教授： 林永松 博士

支援定位服務之感測器配置演算法



Sensor Deployment Algorithms for Target
Positioning Services

研究生： 邱佩玲 撰

中華民國九十六年七月

Sensor Deployment Algorithms for Target Positioning Services

by Pei-Ling Chiu

A dissertation submitted to
Graduate Institute of Information Management
of National Taiwan University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

July 2007

論文摘要

支援定位服務之感測器配置演算法

邱佩玲

中華民國九十六年七月

指導教授：林永松 博士

國立台灣大學資訊管理研究所

近年來，無論在實務或是學術領域，無線感測器網路(Wireless Sensor Networks, WSNs)的應用與技術發展都是極受關注的議題。目前已有許多重要的 WSNs 應用被熱切探討，如：環境監測、標的物定位、物體追蹤及健康照護等。預期未來在軍事與民用上，WSNs 都會有更廣泛的應用出現。

感測器網路的設計有兩個典型的議題：一是如何建構一個能滿足應用所需之服務品質的感測器網路。另一則是如何延長感測網路生命期。從應用層的觀點來看，改善服務品質需求，必須考慮感測器網路對於應用的支援，如環境監測、標的物定位或追蹤等的的能力。再者，感測器電力有限，通常很難再充電。因此，如何延長 WSNs 的生命期也是配置感測器網路所關注的議題。

此論文朝向提供具環境監測與標的物定位服務的應用方向，來探討感測器配置(sensor deployment)議題，由不同觀點進行一系列的深入研究。茲將每一研究主題之內涵與成果簡述如下：

- 考慮在資源限制下，配置一個兼具環境監測與標的物定位功能的同質性感測器網路。此研究的實驗結果顯示，所提演算法不但可得到高品質的解，且具有效力(effectiveness)、擴展性(scalability)與強固性(robustness)。接著探討在給定相同資源的情況下，藉由調整每個感測器的偵測半徑，是否會改善感測器網路的定位服務品質。於是，我們修改求解的演算法。實驗結果顯示，感測器位置已知之下，

調整偵測半徑確實可以很有效的改善 WSNs 的定位服務品質。

- 第二個研究方向是以支援差異化服務品質(differentiated QoS)為焦點。此研究處理感測區內對服務品質及服務優先權的要求並非一致的問題。當資源充足時，以滿足各服務區對於服務品質的要求為目標。或是依據各服務區對服務品質要求的優先權，最小化因資源不足所導致之服務品質降級。實驗結果顯示，當資源稀少時，差異化服務品質的配置機制，較一致化服務品質的配置機制，能獲得更好的整體服務品質等級。而且，當感測區擺設感測器的位置有限制時，採用可調的偵測半徑的感測器比固定半徑的感測器，能獲得更好的服務品質等級。
- 第三個研究方向是以電能效率(energy efficiency)為焦點。配置具有 k 個相互獨立覆蓋(cover)的感測器網路，每個覆蓋輪流支援監測服務，並共同運作以提供定位功能。實驗結果顯示，此機制可以有效地延長感測器網路生命期。每一層覆蓋平均所需感測器密度隨著偵測半徑與覆蓋數增加而降低，甚至可以只有複製配置策略(duplicated placement strategy)的 9%。此外，在一層覆蓋所需的感測器配置成本限制下，改用此電能效率的配置演算法可以使得生命期至少延長 3 倍。

對於上述研究，我們都先將問題描述為數學最佳化模型。提供監測服務的感測網路配置問題，可以說是類比於典型的 set cover 或 set- k cover 問題。這些都是 NP-Complete 的問題。而我們考慮的是兼具環境監測與標的物定位服務的感測網路配置問題，所以是比 set cover 或 set- k cover 更為複雜的問題。我們採用拉格蘭日鬆弛法與模擬退火等方法發展許多的演算法，來解決這一系列最佳化問題。

關鍵詞：感測器配置、標的物定位、完全覆蓋、完全辨識、服務品質、電能效率、拉格蘭日鬆弛法、模擬退火法、數學規劃、網路最佳化

Dissertation Abstract

Sensor Deployment Algorithms for Target Positioning Services

Pei-Ling Chiu

July 2007

Advisor: Prof. Frank Yeong-Sung Lin

Graduate Institute of Information Management

National Taiwan University

From either practical or theoretical viewpoint, wireless sensor network (WSN) techniques are new and important research issues. Numbers of interesting applications for WSNs have been investigated, e.g., surveillance, target positioning, tracing, and health care. Sensor networks have been forecasted to apply variously, both the civilian and military domains.

There are two important issues usually are concerned on WSNs design. One is to construct a qualified WSN for applications to guarantee desired quality of service (QoS). The other challenge is to prolong the network lifetime. From application perspective, in order to improve the QoS supporting by WSNs, the ability of environment surveillance, target positioning, or target tracking have to be controlled carefully. Moreover, it is difficult to replace or recharge the battery for numerous sensors in the most scenarios. Hence, how to prolong the lifetime of WSNs also becomes a key challenge.

In this dissertation, we focus on the sensor deployment problem to support environment surveillance and target positioning services from various perspectives. Subsequently, we present each topic briefly as follows:

- We address the homogeneous sensor placement problem for environment surveillance and target positioning subject to the deployment budget. The experimental results reveal that the proposed algorithm can't only efficiently

obtain a high-quality solution but also is effective, scalable, and robust. Afterward, to investigate whether adjusting the sensing radius for individual sensor will improve QoS for positioning services. The experimental results indicate, when locations are given, using sensors with adjustable sensing radius can actually get better QoS for positioning services.

- The next topic focuses on supporting differentiated QoS for WSNs. The work deals with the differentiated QoS requirements in WSNs. The goal of the problem is either satisfied QoS requirement of all regions of interest (ROIs) or minimized the QoS degradation for ROIs based on its' level of priority. The experimental results show when the given resource is scarce, the sensor deployment approach with differentiated QoS requirements can obtain better QoS solution than that with the uniform QoS requirement. Furthermore, for a sensor field with placement limitations, using sensors with adjustable radius can obtain higher level of QoS than adopting fixed radius sensors.
- The third topic focuses on the energy efficiency issue. We deploy K independent sets of sensors (K covers) to monitoring a sensor field in turn and locating targets together. The experimental results indicate that the strategy is very effective for energy conservation. The average sensor density of one cover is effectively reduced when radius and amount of covers increase; even the deployment cost can be reduced to 9% of that using the duplicate sensor deployment approach. Furthermore, using the same deployment density for a single-cover sensor network, we can deploy an energy-efficient sensor network such that the lifetime extends up to 3 times at least.

All of the problems are formulated as mathematical optimization models. The sensor placement problem for surveillance is analog to classic set-cover/set- K -cover problem, which is NP-complete problem. Our problems consider both environment surveillance and target positioning; it is therefore more difficult than the set-cover/set- K -cover problem. Based on Lagrangean relaxation method and simulated annealing method, we develop many heuristics

to solve these optimization problems.

Keywords: Sensor deployment, target positioning, quality of positioning service, Energy-Efficiency, Lagrangean Relaxation, Simulated Annealing, Mathematical Modeling, Network Optimization.



Table of Contents

| | |
|---|-----------|
| CHAPTER 1 INTRODUCTION | 1 |
| 1.1 OVERVIEW | 1 |
| 1.2 RESEARCH SCOPE | 5 |
| CHAPTER 2 RESEARCH BACKGROUND | 11 |
| 2.1 RELATED WORK | 11 |
| 2.1.1 Coverage | 11 |
| 2.1.2 QoS in WSNs | 22 |
| 2.1.3 Energy Efficiency | 23 |
| 2.1.4 Target Location | 24 |
| 2.1.5 Sensor Deployment | 35 |
| 2.2 LAGRANGEAN RELAXATION METHOD | 46 |
| 2.3 SIMULATED ANNEALING METHOD | 51 |
| CHAPTER 3 SENSOR PLACEMENT ALGORITHMS FOR ACHIEVING COMPLETE COVERAGE AND DISCRIMINATION | 55 |
| 3.1 OVERVIEW | 55 |
| 3.2 PROBLEM DESCRIPTION | 56 |
| 3.3 SENSOR PLACEMENT PROBLEM FOR ACHIEVING COMPLETE COVERAGE AND DISCRIMINATION | 58 |
| 3.3.1 Mathematical Model | 58 |
| 3.3.2 Algorithm | 59 |
| 3.3.3 Computational Results | 61 |
| 3.3.4 Concluding Remarks | 66 |
| 3.4 CONSIDER ADJUSTABLE DETECTION RADIUS ON RANDOM SENSOR PLACEMENT PROBLEM | 67 |
| 3.4.1 Simulated Annealing Based Algorithm | 67 |
| 3.4.2 Computational Results | 69 |
| 3.5 CONCLUDING REMARKS | 75 |
| CHAPTER 4 SENSOR PLACEMENT PROBLEM FOR DIFFERENTIATED QUALITY OF POSITIONING AND SURVEILLANCE SERVICES | 77 |
| 4.1 OVERVIEW | 77 |
| 4.2 PROBLEM DESCRIPTION | 80 |
| 4.2.1 The Framework | 80 |
| 4.2.2 An Example | 82 |
| 4.3 MATHEMATICAL MODEL | 84 |

| | |
|--|------------|
| 4.4 LAGRANGEAN RELAXATION APPROACH | 88 |
| 4.4.1 <i>Equivalent Model</i> | 88 |
| 4.4.2 <i>Transformation</i> | 90 |
| 4.4.3 <i>Relaxation</i> | 92 |
| 4.4.4 <i>Getting Primal Feasible Solutions</i> | 100 |
| 4.4.5 <i>Computational Results</i> | 102 |
| 4.5 SIMULATED ANNEALING APPROACH..... | 113 |
| 4.5.1 <i>Algorithm SA_1</i> | 113 |
| 4.5.2 <i>Algorithm SA_2</i> | 115 |
| 4.5.3 <i>Computational Results</i> | 115 |
| 4.6 PERFORMANCE COMPARISONS | 120 |
| 4.6.1 <i>Performance Evaluations</i> | 120 |
| 4.6.2 <i>Convergence Properties Analysis</i> | 121 |
| 4.6.3 <i>Compare with Other Approaches</i> | 126 |
| 4.6.4 <i>Results Analysis</i> | 129 |
| 4.7 CONCLUDING REMARKS..... | 130 |
| CHAPTER 5 ENERGY-EFFICIENT SENSOR NETWORKS DESIGN | 133 |
| 5.1 OVERVIEW..... | 133 |
| 5.2 PROBLEM DESCRIPTION..... | 135 |
| 5.3 MATHEMATICAL MODEL..... | 142 |
| 5.4 LAGRANGEAN RELAXATION APPROACH..... | 144 |
| 5.4.1 <i>Relaxation</i> | 144 |
| 5.4.2 <i>Getting Primal Feasible Solutions</i> | 147 |
| 5.4.3 <i>Computational Results</i> | 148 |
| 5.5 SIMULATED ANNEALING APPROACH..... | 155 |
| 5.5.1 <i>Algorithm</i> | 155 |
| 5.5.2 <i>Computational Results</i> | 157 |
| 5.6 CONCLUDING REMARKS..... | 167 |
| CHAPTER 6 CONCLUSION AND FUTURE WORK..... | 169 |
| 6.1 SUMMARY | 169 |
| 6.2 FUTURE WORK | 172 |
| REFERENCES..... | 173 |
| PUBLICATIONS..... | 181 |
| APPENDIX A: SET-COVER AND SET K-COVER PROBLEMS..... | 182 |
| APPENDIX B: THEOREM A.1..... | 184 |
| APPENDIX C:..... | 194 |
| APPENDIX D: LIST OF NOTATIONS | 197 |

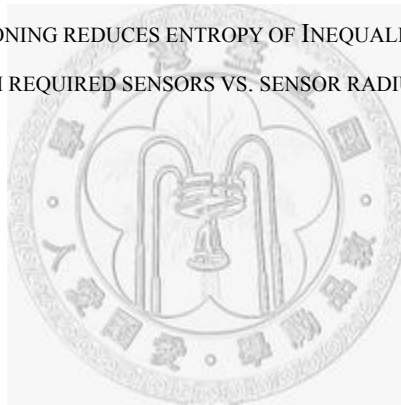
List of Figures

| | |
|---|----|
| FIGURE 1.1: THE CLASSIFICATION OF SENSOR DEPLOYMENT RESEARCH..... | 5 |
| FIGURE 1.2: A COMPLETE COVERED AND DISCRIMINATED SENSOR FIELD. | 8 |
| FIGURE 2.1: BINARY (0/1) DETECTION MODEL. | 12 |
| FIGURE 2.2: PROBABILISTIC DETECTION MODEL. | 13 |
| FIGURE 2.3: PROBABILISTIC DETECTION MODEL. | 14 |
| FIGURE 2.4: CHANGE IN DETECTION PROBABILITY WITH DISTANCE..... | 16 |
| FIGURE 2.5: PROBABILISTIC SENSING MODEL..... | 17 |
| FIGURE 2.6: COMPARISON PHYSICAL AND INFORMATION COVERAGE..... | 19 |
| FIGURE 2.7: (A) AREA COVERAGE AND (B) POINT COVERAGE..... | 20 |
| FIGURE 2.8: THE CLASSIFICATION OF LOCATION SENSING TECHNIQUES. | 25 |
| FIGURE 2.9: TRIANGULATION LOCATION-SENSING TECHNIQUE..... | 25 |
| FIGURE 2.10: AN EXAMPLE OF 2D ANGULATION TECHNIQUE..... | 27 |
| FIGURE 2.11: SENSOR PLACEMENT BASED ON THE CODING THEORY. | 31 |
| FIGURE 2.12: THE LOCATION SYSTEM PROPOSED BY RAY ET AL. | 32 |
| FIGURE 2.13: PERFORMANCE OF VARIOUS HEURISTICS FOR $ V =128$ VERTICES GRAPHS. | 33 |
| FIGURE 2.14: GRID-BASED SENSOR FIELD AND POWER CODE. | 34 |
| FIGURE 2.15: SENSOR PLACEMENTS FOR 4 BY 4 SENSOR FIELD. (A) $M=1$, (B) $M=2$, (C) $M=3$ | 39 |
| FIGURE 2.16: FOUR COMMON REGULAR PATTERNS OF DEPLOYMENT. (A) HEXAGON. (B) SQUARE. (C) RHOMBUS. (D) TRIANGULAR LATTICE..... | 40 |
| FIGURE 2.17: STRIP-BASED DEPLOYMENT PATTERN TO ACHIEVE COVERAGE AND 2-CONNECTIVITY. | 41 |
| FIGURE 2.18: STRIP-BASED DEPLOYMENT THAT IS OPTIMAL FOR ACHIEVING COVERAGE WITH 1-CONNECTIVITY. | 42 |
| FIGURE 2.19: THE CONCEPT OF THE DUAL PROBLEM. | 48 |
| FIGURE 2.20: THE OVERALL PROCEDURE OF THE LR APPROACH. | 49 |
| FIGURE 1.2: A COMPLETE COVERED AND DISCRIMINATED SENSOR FIELD. | 8 |
| FIGURE 3.1: A COMPLETE COVERED AND DISCRIMINATED SENSOR FIELD. | 56 |
| FIGURE 3.2: ERROR DISTANCE VS. SENSOR DENSITY. (10x10, $R=1$)..... | 63 |
| FIGURE 3.3: ERROR DISTANCE VS. SENSOR DENSITY. (30x30, $R=1$)..... | 64 |
| FIGURE 3.4: ERROR DISTANCE VS. SENSOR DENSITY. (10x10, $R=2$)..... | 65 |
| FIGURE 3.5: ERROR DISTANCE VS. SENSOR DENSITY. (30x30, $R=2$)..... | 65 |
| FIGURE 3.6: AVERAGE ERROR DISTANCE OF THE RANDOM APPROACH. | 70 |

| | |
|---|-----|
| FIGURE 3.7: MINIMUM ERROR DISTANCE OF THE RANDOM APPROACH. | 70 |
| FIGURE 3.8: PROBABILITY TO ACHIEVE COMPLETE COVERAGE. (RANDOM APPROACH) | 71 |
| FIGURE 3.9: PROBABILITY TO ACHIEVE COMPLETE DISCRIMINATION. (RANDOM APPROACH) | 71 |
| FIGURE 3.10: AVERAGE ERROR DISTANCE. | 72 |
| FIGURE 3.11: MINIMUM ERROR DISTANCE..... | 73 |
| FIGURE 3.12: PROBABILITY TO ACHIEVE COMPLETE DISCRIMINATION..... | 73 |
| FIGURE 4.1: A MAP OF MUSEUM. (A SENSOR FIELD WITH 150 SERVICE POINTS) | 80 |
| FIGURE 4.2: RELATIONSHIP BETWEEN V_{IK} , V_{JK} , AND T_{LIK} | 90 |
| FIGURE 4.3: THE MINIMUM REQUIRED NUMBER OF SENSOR VS. VARIOUS SENSING RADIUS. (FIXED RADIUS)..... | 103 |
| FIGURE 4.4: THE BEST-FOUND OBJECTIVE VALUES FOR VARIOUS SET OF SENSING RADIUS. (UNIFORM QoS)..... | 104 |
| FIGURE 4.5: THE BEST-FOUND OBJECTIVE VALUES FOR VARIOUS SET OF SENSING RADIUS. (DIFFERENTIATED QoS) | 104 |
| FIGURE 4.6: PERFORMANCE COMPARISON BETWEEN THE UNIFORM (U) AND DIFFERENTIATED (D) QoS SERVICES. (ADJUSTABLE RADIUS, $R=\{3, 4, 5, 6, 7\}$) | 105 |
| FIGURE 4.7: PERFORMANCE COMPARISON BETWEEN THE UNIFORM (U) AND DIFFERENTIATED (D) QoS SERVICES. (FIXED RADIUS, $R=5$)..... | 106 |
| FIGURE 4.8: THE TOPOGRAPHY WITH PLACEMENT LIMITATIONS. | 107 |
| FIGURE 4.9: THE MINIMUM REQUIRED NUMBER OF SENSOR VS. VARIOUS SENSING RADIUS. (FIXED RADIUS, PLACEMENT LIMITED) | 107 |
| FIGURE 4.10: THE BEST-FOUND OBJECTIVE VALUES FOR VARIOUS SET OF SENSING RADIUS. (UNIFORM QoS, PLACEMENT LIMITED) | 108 |
| FIGURE 4.11: THE BEST-FOUND OBJECTIVE VALUES FOR VARIOUS SET OF SENSING RADIUS. (DIFFERENTIATED QoS, PLACEMENT LIMITED)..... | 108 |
| FIGURE 4.12: PERFORMANCE COMPARISON BETWEEN THE UNIFORM (U) AND DIFFERENTIATED (D) QoS SERVICES. (ADJUSTABLE RADIUS, PLACEMENT LIMITED) | 109 |
| FIGURE 4.13: PERFORMANCE COMPARISON BETWEEN THE UNIFORM (U) AND DIFFERENTIATED (D) QoS SERVICES. (FIXED RADIUS, $R=6$, PLACEMENT LIMITED)..... | 110 |
| FIGURE 4.14: THE MINIMUM REQUIRES SENSOR DENSITY UNDER VARIOUS SENSING RADIUS AND SENSING AREA. (LR, FIXED RADIUS)..... | 111 |
| FIGURE 4.15: THE COMPUTATION TIME. (LR)..... | 111 |
| FIGURE 4.16: PERFORMANCE COMPARISON BETWEEN THE UNIFORM (U) AND DIFFERENTIATED (D) QoS SERVICES. (SA_2, $R=\{3, 4, 5, 6, 7\}$) | 117 |
| FIGURE 4.17: PERFORMANCE COMPARISON BETWEEN ALGORITHMS SA_1 AND SA_2. (DIFFERENTIATED QoS SERVICE, $R=\{3, 4, 5, 6, 7\}$) | 117 |

| | |
|---|-----|
| FIGURE 4.18: THE MINIMUM REQUIRES SENSOR DENSITY UNDER VARIOUS SENSING RADIUS AND SENSING AREA. (SA_2, FIXED RADIUS)..... | 118 |
| FIGURE 4.19: THE SOLUTION TIME (SECONDS) OF SA_2 IN VARIOUS AREAS. | 119 |
| FIGURE 4.20: COMPARISON OF SOLUTION QUALITY BETWEEN SA_1, SA_2, AND LR ALGORITHMS. (AREA: 10X15, DIFFERENTIATED QOS, R={3, 4, 5, 6, 7}) | 120 |
| FIGURE 4.21: CONVERGENCE TREND OF LR ALGORITHM. (AREA: 10 BY 15, DIFFERENTIATED QOS, R= {3, 4, 5, 6, 7}, NUMBER OF SENSOR IS 25, I.C.=IMPROVEMENT COUNTER)..... | 122 |
| FIGURE 4.22: CONVERGENCE TREND OF SA_2 FOR DIFFERENT COOLING RATIO..... | 124 |
| FIGURE 4.23: CONVERGENCE TREND OF SA_2 FOR DIFFERENT R_0 . ($N=150$)..... | 124 |
| FIGURE 4.24: CONVERGENCE TREND OF SA_2 FOR DIFFERENT T_0 (OR p_0)...... | 125 |
| FIGURE 4.25: THE SENSOR DEPLOYMENT IN A 13X13 SENSOR FIELD. (A) BY CIQ APPROACH (79 SENSORS). (B) BY LR APPROACH (68 SENSORS)..... | 126 |
| FIGURE 4.26: PERFORMANCE OF THE ID-CODE ALGORITHM IN 150 GRID POINTS SENSOR FIELD. | 127 |
| FIGURE 4.27: PERFORMANCE COMPARISONS BETWEEN ID_CODE_BEST, SA_2, AND LR ALGORITHMS UNDER VARIOUS AREAS OF SENSOR FIELDS..... | 128 |
| FIGURE 5.1: A COMPLETE COVERAGE/DISCRIMINATION SENSOR FIELD..... | 135 |
| FIGURE 5.2: A GRID-BASED SENSOR FIELD WITH 3 COVERS: (A) OVERALL PLACEMENT. (B) COVER 1. (C) COVER 2. (D) COVER 3..... | 136 |
| FIGURE 5.3: THE STATE DIAGRAM OF THE SENSOR NETWORK..... | 137 |
| FIGURE 5.4: SENSOR AND ITS COVERAGE. (THE DISTANCE BETWEEN ANY TWO ADJACENT GRID POINTS IS USED AS ONE LENGTH UNIT.) | 140 |
| FIGURE 5.5: PROPORTION OF THE LIFETIME EXTENDING TIMES TO AVERAGE SENSOR DENSITY PER COVER. (LR, R=1~4)..... | 151 |
| FIGURE 5.6: PROPORTION OF THE LIFETIME EXTENDING TIMES TO AVERAGE SENSOR DENSITY PER COVER. (LR, R=5~7)..... | 151 |
| FIGURE 5.7: VARIANT OF THE SENSOR RADIUS AND THE CORRESPONDING DENSITY REQUIREMENT. (LR, K=1) | 153 |
| FIGURE 5.8: THE SOLUTION TIME FOR 10X10 SENSOR AREA. (R=4)..... | 153 |
| FIGURE 5.9: THE SOLUTION TIME FOR VARIOUS SENSOR AREAS. (LR, R=1)..... | 154 |
| FIGURE 5.10: AVERAGE DEPLOYMENT DENSITY FOR 10X10 SENSOR FIELD. (SA, R=1 ~ 4)..... | 160 |
| FIGURE 5.11: AVERAGE DEPLOYMENT DENSITY FOR 10X10 SENSOR FIELD. (SA, R=5 ~ 7)..... | 161 |
| FIGURE 5.12: THE REQUIREMENT SENSOR DENSITY COMPARES BETWEEN THE SA ALGORITHM AND CPLEX. (R=4) | 161 |
| FIGURE 5.13: THE REQUIREMENT SENSOR DENSITY COMPARES BETWEEN THE SA ALGORITHM AND CPLEX. (R=5)..... | 162 |

| | |
|--|-----|
| FIGURE 5.14: THE REQUIREMENT SENSOR DENSITY COMPARES BETWEEN THE SA ALGORITHM AND CPLEX. (R=6) | 162 |
| FIGURE 5.15: THE REQUIREMENT SENSOR DENSITY COMPARES BETWEEN THE SA ALGORITHM AND CPLEX. (R=7) | 163 |
| FIGURE 5.16: THE REQUIRED SENSOR DENSITY FOR VARIOUS SENSOR FIELDS. (K=1, R=1) | 164 |
| FIGURE 5.17: THE REQUIRED SENSOR DENSITY FOR VARIOUS SENSOR FIELDS. (K=2, R=1) | 165 |
| FIGURE 5.18: THE REQUIRED SENSOR DENSITY FOR VARIOUS SENSOR FIELDS. (K=3, R=1) | 165 |
| FIGURE 5.19: THE SOLUTION TIME FOR VARIOUS SENSOR FIELDS (K=1)..... | 166 |
| FIGURE A.1: RELATIONSHIP BETWEEN ENTROPY AND MUTUAL INFORMATION..... | 185 |
| FIGURE A.2: RANDOM VARIABLES X_k (SENSOR k) AND Y (GRID POINT IDENTIFICATION)..... | 187 |
| FIGURE A.3: AN EXAMPLE OF IDEAL IDENTIFYING CODE..... | 187 |
| FIGURE A.4: MUTUAL INFORMATION AND ENTROPY IN EQUATION (1)..... | 188 |
| FIGURE A.5: A LOT OF GRID POINTS CANNOT BE UNIQUELY IDENTIFIED..... | 189 |
| FIGURE A.6: MUTUAL INFORMATION AND ENTROPY FOR EQUATION (2)..... | 189 |
| FIGURE A.7: MUTUAL INFORMATION AND ENTROPY OF INEQUALITY (4)..... | 190 |
| FIGURE A.8: THE CONDITIONING REDUCES ENTROPY OF INEQUALITY (6). | 191 |
| FIGURE A.9: THE MINIMUM REQUIRED SENSORS VS. SENSOR RADIUS IN A 10X15 SENSOR FIELD. 193 | |



List of Tables

| | |
|--|-----|
| TABLE 1.1: SCOPE AND PROBLEM DEFINITION OF THIS DISSERTATION..... | 7 |
| TABLE 2.1: QoS PARAMETERS AND FUNCTION LAYERS. | 22 |
| TABLE 2.2: CLASSIFICATION OF LITERATURES FOR THE SENSOR DEPLOYMENT PROBLEM. | 44 |
| TABLE 2.3: APPLICATIONS OF LAGRANGIAN RELAXATION. | 50 |
| TABLE 3.1: COMPARISON BETWEEN EXHAUSTIVE SEARCH AND THE SA ALGORITHM..... | 62 |
| TABLE 4.1: THE QoS SUPPORTED FOR ROIS IN THE MUSEUM EXAMPLE. | 83 |
| TABLE 4.2: TRUTH TABLE FOR VARIABLES V_{IK} , V_{JK} , AND T_{IJK} | 90 |
| TABLE 4.3: THE LEVELS OF QoS AND THEIR RANGES OF $Z_{IP4.3}$ IN THE EXPERIMENT FOR DIFFERENTIATED QoS..... | 105 |
| TABLE 4.4: THE MIN. REQUIRED SENSORS (S) FOR LEVEL 4 QoS IN SA_2. | 116 |
| TABLE 4.5: THE MINIMAL IMPROVEMENT COUNTERS FOR CONVERGENCE IN 150 GRID POINTS SENSOR FIELD. | 123 |
| TABLE 5.1: THE WORKING MODES ON THREE STATES FOR THE SENSOR NODE. THE RADIO IS DOMINANT POWER CONSUMER. | 138 |
| TABLE 5.2: THE THEORETIC UPPER BOUND OF THE NUMBER OF COVERS IN 10X10 SENSOR FIELD. | 141 |
| TABLE 5.3: COMPARISON OF U_r BETWEEN THE THEORETICAL AND THE BEST FOUND VALUES. | 149 |
| TABLE 5.4: SELECTED SENSOR DENSITIES OBTAINED IN THE EXPERIMENT. | 150 |
| TABLE 5.5: PERFORMANCE COMPARISON BETWEEN THE DUPLICATE DEPLOYMENT AND THE PROPOSED SENSOR PLACEMENT APPROACH. | 152 |
| TABLE 5.6: THE EXECUTION TIME OF EACH SET EXPERIMENTS. | 152 |
| TABLE 5.7: THE SA PSEUDO CODE FOR SENSOR PLACEMENT. | 156 |
| TABLE 5.8: COMPARISON OF U_r BETWEEN THE THEORETICAL AND THE BEST FOUND VALUES. | 159 |
| TABLE 5.9: SELECTED SENSOR DENSITIES OBTAINED IN EXPERIMENT. | 159 |
| TABLE 5.10: PERFORMANCE COMPARISON BETWEEN THE DUPLICATE DEPLOYMENT AND THE SA-BASED SENSOR PLACEMENT ALGORITHM. | 160 |
| TABLE 5.11: THE SOLUTION TIME OF THE SA ALGORITHM. | 163 |
| TABLE A.1: LISTS OF NOTATIONS USED IN APPENDIX B. | 184 |
| TABLE A.2: THE MINIMUM REQUIRED SENSORS FOR VARIOUS RADII. | 193 |
| TABLE A.3: THE MINIMUM SENSOR DENSITY (%) FOR VARIOUS SIZES OF SENSOR FIELD. (THE WIDTH OF SENSOR FIELD IS 15.)..... | 195 |

CHAPTER 1 INTRODUCTION

1.1 Overview

The evolutions in sensor technology and wireless communication have led to the development of wireless sensor networks (WSNs) [QIC01]. WSNs will affect phenomena sensing and monitoring in upcoming years. Many applications of WSNs, such as surveillance [DC03] [DCI02], target positioning [CIQ02] [CL04] [LC05] [RST04] [RUP03] [ZC03a] [ZC03b], environment monitoring, health care, and animal tracking, have been studied [CHS04].

A WSN comprises of a large number of tiny sensor nodes, which are low-cost and low power. Numerous sensors are ad hoc deployed in the interested area. These sensors collect physical information, process it and forward the local information to the sink nodes. Hence, the back-ends can obtain global views according to the information provided by the sensors [ASC02a] [ASC02b]. The system responds for making appropriate decisions based on the received information. Obviously, the quality of the information dominates the final decision for the WSNs. Correct information can be obtained by constructing a WSN with a high quality of service (QoS) through careful planning in the sensor deployment phase. For instance, numbers of sensors are deployed carefully to fully cover the whole sensor field such that any target, which is at any position in the sensor field, can be detected by at least one sensor.

Then the sensor field is completely covered by the sensor network. Therefore, the QoS of a WSN is one of the most important issues in sensor deployment problem.

The quality of service of a WSN is influenced by the power efficiency, deployment cost, deployment methods, and other physical limitations. A sensor network can be deployed in two ways: random or controlled placement [MKP01]. When the environment is unknown, dangerous, or inhospitable, sensors cannot be deployed manually. In such environments, the placement may be relied on aircrafts, cannons, rocket, or missile. By this way, massive sensor nodes will be randomly placed on anywhere of sensor field. On the contrary, if the terrain of sensor field is predetermined, we can adopt the controlled approach that deploys sensors by carefully planning to meet a desired quality of service requirement. For instance, we can construct a WSN by carefully planning in parking lots, schools, shopping malls, art galleries, or everywhere we interested, to perform surveillance, target positioning, or target tracking [LP06]. Obviously, to achieve the same quality of service requirement, the random placement approach wastes more resources than the controlled placement approach.

Recently, the emerging the mobile computing applications and trend of user-centric services enable the requirements of location-based services, personalized services, and user-aware services increase rapidly, for instance, mobile learning and emergency rescue services [CCK06] [DCT05] [KKL06] [LC06] [LXP05]. No doubt, localization capability is therefore one of the most important technique for supporting these applications. In various localization techniques, sensor-network-based positioning system can provide the indoor localization services a simple and feasible solution. Hence, besides the surveillance, this dissertation also focuses on target positioning services.

One of important applications for sensor network is *target location*, i.e. *target positioning*, which refers to decide the position of target by cooperation of sensors in a sensor network [CIQ01] [CIQ02]. Hence, the sensors must be deployed carefully. Besides, the sensor network coverage has to cover the whole

sensor field, if the coverage areas of multiple sensors overlap, they may all report a target in their respective zones. Based on the reports, location of the target can be determined. If target in a zone (i.e. region) can be detected by a unique set of sensors, the zone is denoted by distinguishable zone. The diameter of a distinguishable zone dominates the accuracy of the target location. Some papers use service points as reference points to replace zone/region for positioning service. When sensor network determine a target being at a certain service point, it means the target might occur on the region including the service point.

Sensor localization is another import research issue in WSNs. The sensor localization process refers to find the positions of all sensors in a sensor field based on partial information, namely the exact locations of only a few sensors (called anchor sensors) and the mutual distances between pairs of sensors that are within radio distance [CKN07] [WX07]. Sensor localization problem differs from target positioning problem; the latter is concerned in this dissertation.

The degree of coverage is one of the fundamental metrics that used to quantify the QoS for a WSN which supports surveillance services [SS05] [SSS03] [WXZ03]. The sensor deployment problem where subject to coverage, has been transformed to the classical set-cover/set- k -cover problem (Appendix A) [GJ79] [GT02] [SP01], which is NP-complete, has been studied intensely for last decades [MP03]. Both surveillance and target positioning ability are adopted as the QoS parameters of the sensor placement problem in this dissertation. Hence, the proposed problems are more difficult then the classical set-cover/set k -cover problem.

On the other hand, due to cost and environment concerns, the battery of sensor is not always rechargeable — particularly when the sensor network operates in inhospitable or hostile fields. Once the sensors energy exhaust, the sensors fail to perform their jobs, it will result in degradation of quality of service on the sensor network. Therefore, how to design an energy-efficient sensor network is really a major challenge.

While the sensor deployment process subjects to some resource constraints (such as, the deployment budget as well as coverage requirement) and to achieve some specific goals (for instance, minimizing the deployment cost, maximizing the quality of service, or maximizing the lifetime) the sensor deployment can be dealt as an optimization problem. In this dissertation, we formulate these sensor deployment problems as mathematical optimization models. As well as many effective algorithms based on Lagrangean relaxation and simulated annealing meta-heuristics are proposed.

The dissertation significantly contributes to the target positioning and the wireless sensor deployment research domain. First, we introduced the positioning ability as the QoS parameter in WSNs from application perspective, as well as proposed the error distance to measure the positioning ability. Next, a generic framework for sensor deployment problem has been proposed to support the differentiated quality of positioning service in WSNs. Third, an energy-efficient sensor deployment scheme for target positioning has been proposed. Fourth, the mathematical optimization models have been proposed to define each problem strictly. Last, these combinatorial optimization problems are hard to solve. We have successfully developed many Lagrangean-relaxation-based and simulated-annealing-based algorithms to cope with these hard problems and presented detailed performance evaluations. Based on the research results, more relative problems can be solved by the proposed methods with minor modifications.

1.2 Research Scope

Recently, a lot of papers focused on the sensor deployment problems taking account of various constraints and goals, and several solutions are proposed. To clarify the research field, we attempt to categorize the researches according to different dimensions.

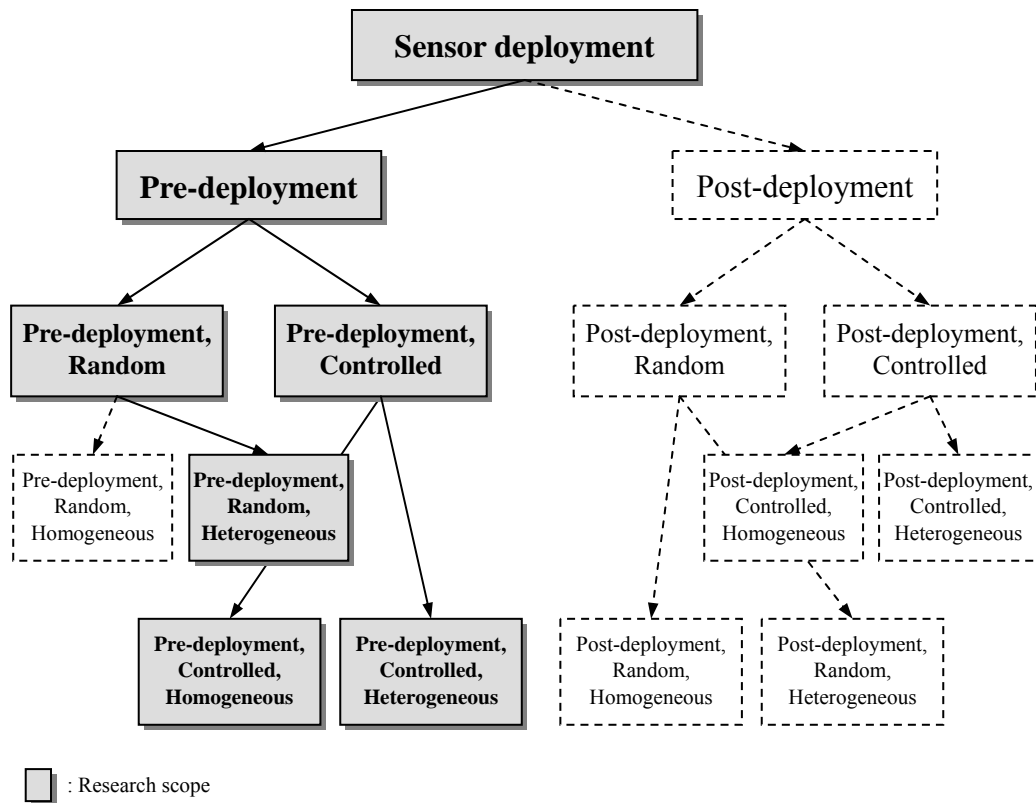


Figure 1.1: The classification of sensor deployment research.

In this section, we discuss the sensor placement problems from three different dimensions, illustrated in Figure 1.1. First, the problem can be

categorized due to the *placement method*, which includes random and controlled deployments. While the environment is unknown, the random deployment approach is adopted. Conversely, if the terrain of sensor field is predetermined, WSNs can be constructed by a deterministic way. Hence, the controlled deployment approach is fit for a candidate solution.

In reference [ASC02a], the authors define three phases for topology maintenances and changes in WSNs, including pre-deployment/deployment, post-deployment, and redeployment. We consider the timing for topology constructing and maintenance as the second dimension, as well as define two phases: pre-deployment and post-deployment phases. The algorithms might execute before sensors are deployed by random or controlled manner. Hence the sensor network topology is determined at the deploying time. We classify it the pre-deployment phase algorithms. For example, according given parameters, e.g., the topography of sensing field, sensing/communication abilities of sensor nodes, and energy constraints, the random deployment algorithm may compute the resource requirements and dropping locations to satisfy lifetime, coverage, and connectivity constraints before scattering sensors [AS03] [GCB06]. After sensors are deployed, topology might be changed by nature or artificial ways. As well as some sensor nodes are added or redeployed to sensor field. We call it is post-deployment. For instance, according the given positions of sensors, the algorithms determine the cover or role for each sensor to optimize the network lifetime [VGD06]. Moreover, after scattering mobile sensors, several papers investigate the moving directions, distance, and final positions of each mobile sensor to improve the coverage of sensor networks [ZC03b].

The last dimension in the research framework considers whether all sensors with the same abilities/duties. A number of papers take account of the homogeneous sensors in the deployment problem. All of the sensors have the same specification. Contrarily, numbers of papers consider the heterogeneous sensors being deployed. For example, the two-tier sensor networks are investigated frequently [XWH05]. The sensors in the lower tier account for sensing and forwarding the information to its cluster node. In the upper tier, the

all cluster nodes compose a connected network, these clusters deliver the information coming from the sensing nodes and the other clusters such that the information can be sent to the base stations.

Table 1.1: Scope and problem definition of this dissertation.

| Problem 1 | |
|-------------------------|---|
| Given parameters | Sensor field, set of service points, sensor cost, and detection range of sensor. |
| Constraints | Budget and complete coverage. |
| Objective | Complete discrimination / minimizing the maximum error distance. |
| Outcomes | Sensors' location and power vectors. |
| Approaches | Simulated annealing method and exhaustive search. |
| Problem 2 | |
| Given parameters | Sensor field, set of service points, number of sensors, location of each sensor, and set of candidate detection ranges. |
| Constraints | Complete coverage. |
| Objective | Complete discrimination / minimizing the maximum error distance. |
| Outcomes | Detection range for each sensor and power vectors. |
| Approaches | Simulated annealing method. |
| Problem 3 | |
| Given parameters | Sensor field, set of service points, deployment cost for each location, and set of candidate detection ranges. |
| Constraints | Budget, complete coverage, and service priority for each region of interest. |
| Objective | Minimizing the weighted maximum error distance / maximizing the level of QoS. |
| Outcomes | Sensors' location and detection range, and power vectors |
| Approaches | Lagrangian relaxation method and simulated annealing method |
| Problem 4 | |
| Given parameters | Sensor field, set of service points, sensor cost, detection radius of sensor, and number of covers. |
| Constraints | Complete coverage constraint for each cover and complete discrimination constraint for the whole sensor network. |
| Objective | Minimizing the deployment cost. |
| Outcomes | Sensors' location, members of each cover, and power vectors. |
| Approaches | Lagrangian relaxation method, simulated annealing method and CPLEX. |

In this dissertation, we study several sensor deployment problems (summarized in Table 1.1), which belong to both single and multiple categories. Mathematical formulations are used to model these problems. Based on the proposed mathematical models, Lagrangean relaxation and simulated annealing methods are adopted to solve the sensor deployment problems.

In this dissertation, we consider that the sensor field is divided into grid points at which sensors are carefully deployed. This approach is called as grid-based placement.

A grid-based sensor field can be represented as a collection of two- or three-dimensional *grid points* [DCI02]. A set of sensors can be deployed on the grid points to monitor the sensor field. The grid point, which requires the surveillance or positioning service, is also called *service point*. In this dissertation, we consider the detection model of a sensor to be a 0/1 coverage model. The coverage is assumed to be full (1) if the distance between the service point and the sensor is less than the detection radius of the sensor. Otherwise, the coverage is assumed to be non-effective (0). If any service point in a sensor field can be detected by at least one sensor, we call the field is *completely covered*, as shown in Figure 1.2. In this context, a target can be detected at any place in the field.

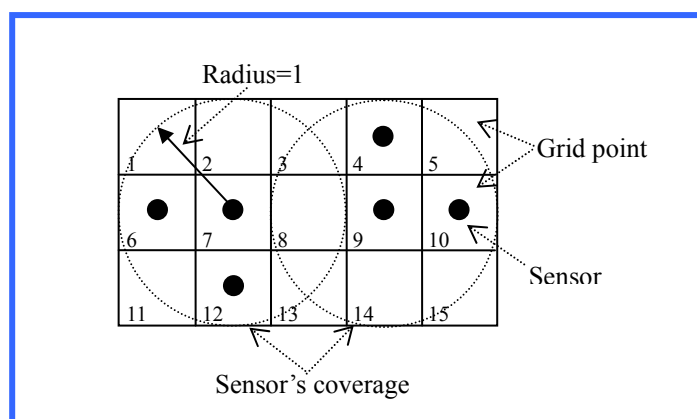


Figure 1.2: A complete covered and discriminated sensor field.

A *power vector* is defined for each service point to indicate whether sensors can cover a service point in a field. As shown in Figure 1.2 the power vector of service point 8 is $(0, 0, 1, 1, 0, 0)$ corresponding to sensor 4, 6, 7, 9, 10, and 12. In a completely covered sensor field, when each service point has a unique power vector, we note the sensor field is *completely discriminated*, as shown in Figure 1.2. In this case, as soon as a target occurs in a grid of the sensor field, it can be located by the back-end according to the power vector of the service point.

In Chapter 3, two sensor-placement problems for positioning targets are discussed. The first problem, given the topography of sensor field, deployment budget, and uniform detection radius of sensors, we determine the position of each sensor to obtain a completely covered/discriminated sensor field when deployment budget is adequate, and to optimize the positioning accuracy when deployment budget is scarce. We classify the research is a pre-deployment algorithm using controlled placement approach to obtain a homogeneous sensor network.

The second problem in Chapter 3, given the location of each sensor by randomly scattering, we design a algorithm to adjust the detection radius of individual sensor such that the quality of positioning service is optimized. We classify the research is a pre-deployment problem using random deployment approach to obtain a heterogeneous sensor network.

In Chapter 4, continuing from Chapter 3, we deal with the sensor deployment problem for supporting differentiated QoS of target positioning. The QoS requirement and level of priority for each region of interest (ROI) in sensor field is given. The locations and detection radius of sensors are decision variables [WY04]. When the given resource is adequate, the objective of the problem is to satisfy the QoS requirements of each ROI. When the given resource is scarce, the objective is to minimize the QoS degradation based on the level of service priority for each ROI. We classify the research is a pre-deployment algorithm using controlled placement approach to obtain a heterogeneous sensor network. Finally, in Chapter 5, we address the

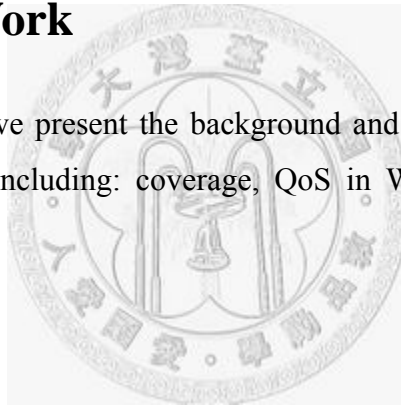
energy-efficient sensor network deployment problem. We consider deploying K independent sets of sensors to monitoring the sensor field in turn and locating the target together. With this strategy, the duty cycle of each sensor is only $1/K$ and the lifetime of the sensor network will be extended up to K times. The objective of the problem is to minimize the deployment cost, where the position and cover of each sensor are decision variables. We classify the research is a pre-deployment algorithm using controlled placement approach to obtain a homogeneous sensor network.



CHAPTER 2 RESEARCH BACKGROUND

2.1 Related Work

In this section, we present the background and related work about sensor placement problem, including: coverage, QoS in WSNs, target location, and sensor deployment.



2.1.1 Coverage

The coverage model and location of all sensors determine the whole network coverage in sensor field. As pointed out in [CW06] [MKP01], *sensor coverage* concept is a measure of the quality of service (QoS) of the sensing function and is subject to a wide range of interpretations due to a large variety of sensors and applications. On the other hand, *network coverage* can be considered as a collective measure of the quality of service provided by sensors at different geographical locations.

2.1.1.1 Coverage Models

The most commonly used sensor coverage model is a sensing *disk model*. All points within a disk centered at sensor are considered to be covered by the

sensor. In the literature of WSNs, however, many papers assume a fixed sensing range and an isotropic detection capability of sensor, i.e., a disk coverage model [Wan06]. The detection ability within coverage of a sensor can be classified as the 0/1 coverage model (binary model), the probabilistic coverage model, and the information coverage model.

A. Binary Model (0/1 Model)

In many cases, the coverage model is simplified as 0/1 model [CIQC02] [MP03]. The coverage is assumed to be full (1) if the distance between the service point and the sensor is less than the detection radius of the sensor, as Figure 2.1 shown. Otherwise, the coverage is assumed to be non-effective (0). For instance, a sensor with camera captures photos around its nearby environment. If an object can be recognized from images, it means the object being covered by the sensor. The resolution threshold determines the detection radius of sensor.

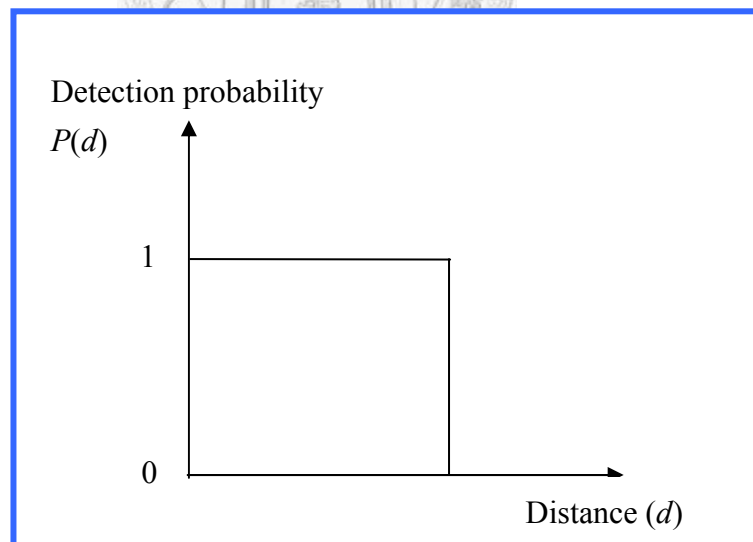


Figure 2.1: Binary (0/1) detection model.

B. Probabilistic Models

(1) Probabilistic Model 1

Several papers [DC03] [NKJ05] adopt probabilistic models, which assume the probability of sensing is function of the distance between sensors and events. Dhillon and Chakrabarty assume that the probability of detection of a target by a sensor varies exponentially with the distance between the target and the sensor [DC02] [DC03]. The model is shown in Figure 2.2. A target at distance d from a sensor is detected by the sensor with probability

$$p(d) = e^{-\alpha d}. \quad (2.1)$$

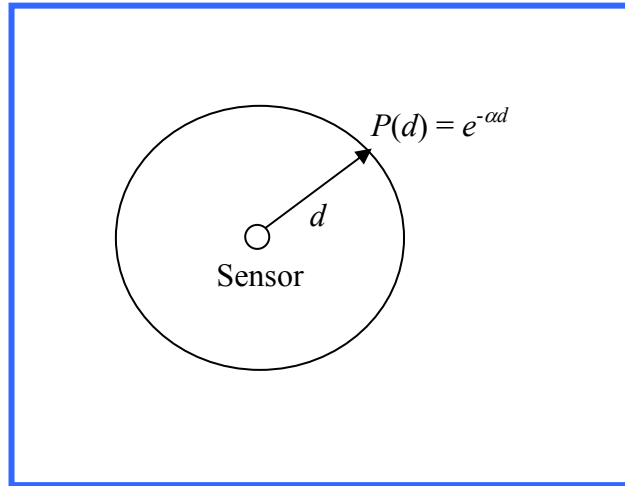


Figure 2.2: Probabilistic detection model.

(2) Probabilistic Model 2

Assume sensor s_i has detection radius r . For any point P , the Euclidean distance between s_i and P is denoted as $d(s_i, P)$. Zou and Chakrabarty [ZC03a] [ZC03b] present the probability-based sensor detection model as Figure 2.3. The coverage of sensor s_i , $c(s_i)$, is represented as follows:

$$c(s_i) = \begin{cases} 0, & \text{if } r + r_e \leq d(s_i, P), \\ e^{-\lambda a^\beta}, & \text{if } r_e > |r - d(s_i, P)|, \\ 1, & \text{if } r - r_e \geq d(s_i, P), \end{cases} \quad (2.2)$$

, where r_e , $r_e < r$, is a measure of the uncertainty in sensor detection. $a = d(s_i, P) - (r - r_e)$. Parameters λ and β are used to measure detection probability when a target is at distance greater than $(r - r_e)$ but within a distance $(r + r_e)$ from the sensor. The probabilistic sensor detection model is shown in Figure 2.3. Different values of parameters λ and β yield different translations reflected by different detection probabilities, which can be viewed as the characteristics of various types of physical sensors. This model reflects the behavior of range sensing devices such as infrared and ultrasound sensors.

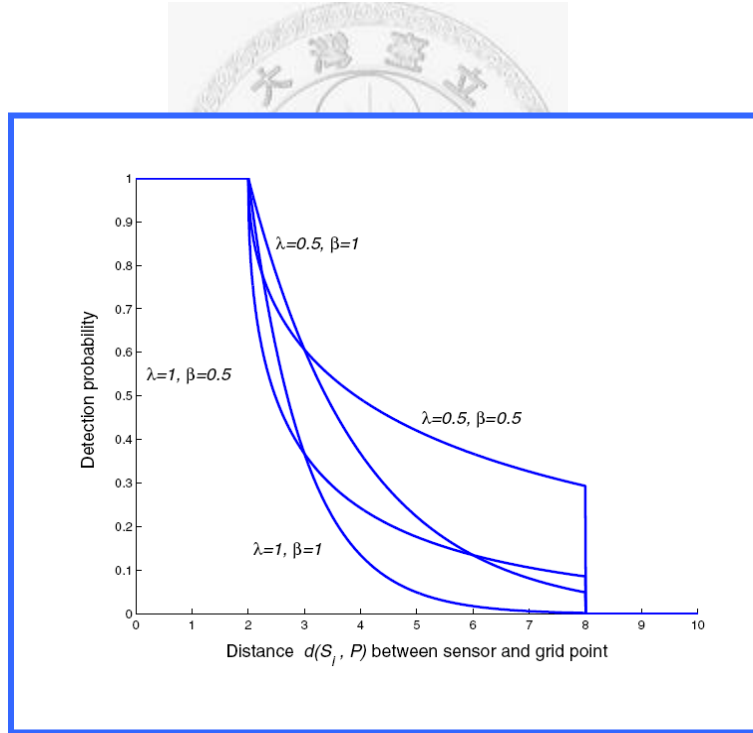


Figure 2.3: Probabilistic detection model.

(3) Probabilistic Model 3

To capture the real world sensing characteristics of sensor nodes, Ahmed et

al. [AKJ05] assume that the signal propagation from a target to a sensor node follows a probabilistic model. This assumption is only valid for certain kind of sensors e.g. radio, acoustic, seismic etc.

This model is based on the *path loss log normal shadowing model*, it can be extended to incorporate different signal decay models e.g. acoustic signal model for acoustic sensors. The probability of detection of a target by a sensor decreases exponentially with increase in distance between the target and the sensor. Using the log-normal shadowing model, the path loss PL (in dB) at a distance d is given as follows [Rap96]:

$$\overline{PL(d)} = \overline{PL(d_0)} + 10 \cdot n \cdot \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (2.3)$$

, where,

d_0 = Reference distance,

n = Path loss component, indicating the rate at which the path loss increases with distance,

X_σ = Zero-mean Gaussian distributed random variable (in dB) with σ -variance (shadowing, also in dB),

$\overline{PL(d_0)}$ = Mean path loss at reference distance d_0 .

Equation (2.3) captures various environmental factors resulting in different received signal values at different locations although the distance between the target and sensor is the same. Parameters n and X_σ can be measured experimentally. Similarly, $\overline{PL(d_0)}$ can be measured experimentally for given event and sensor characteristics or can be calculated using free space path loss model.

Each sensor has a *receive threshold value* γ that describes the minimum signal strength that can be correctly decoded at the sensor. The probability that the received signal level at a sensor will be above this receive threshold, γ , is given by Equation (2.6), requiring Q-function to compute probability involving

the Gaussian process. The Q-function is defined as Equation (2.4).

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^{\infty} \exp\left(-\frac{x^2}{2}\right) dx \quad (2.4)$$

, where

$$Q(z) = 1 - Q(-z) \quad (2.5)$$

$$\Pr[\Pr(d) > \gamma] = Q\left[\frac{\gamma - \Pr(d)}{\sigma}\right]. \quad (2.6)$$

For a given transmit power and receive threshold value, we can calculate the probability of receiving a signal above the receive threshold value, γ , at a given distance using Equations (2.6) and (2.4). Figure 2.4 shows the decrease in detection probability example for a sensor [AKJ05].

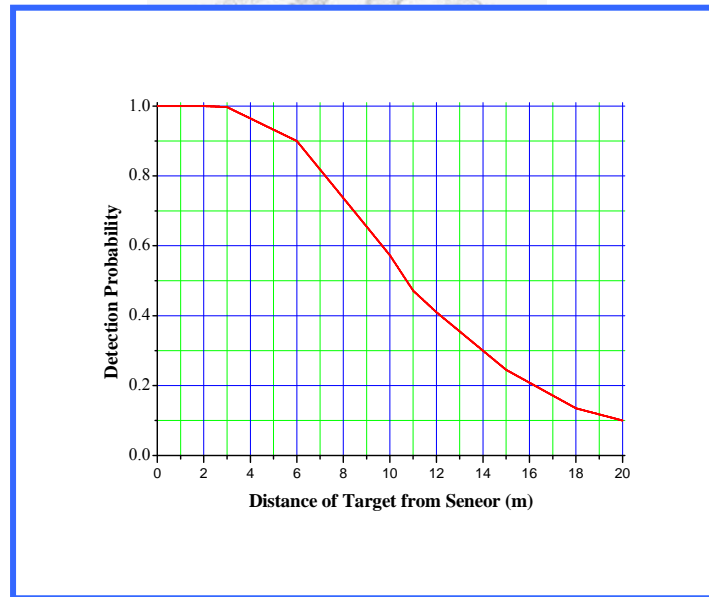


Figure 2.4: Change in detection probability with distance.

(4) Probabilistic Model 4

Liu and Towsley [HA07] [LT04] adopt the following general sensing

model (Equation (2.7)), which has been widely used in several papers, and is reasonable for radio, acoustic and seismic signals, as Figure 2.5 shown [LT04]. For a sensor s , the sensing signal at an arbitrary point p is given by:

$$S(s, p) = \begin{cases} \frac{\alpha}{d(s, p)^\beta} & A \leq d(s, p) < B \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

, where α is the energy emitted by events occurring at point p ; $d(s, p)$ is the Euclidean distance between sensor s and point p ; parameters A and B define the range of a sensor's sensing capability. The sensing signal decays according to a power law with exponent β . The value of the decaying exponent is assumed to be known (or estimated via experiments). For radio signal sensing, the exponent typically ranges from 2.0 to 5.0.

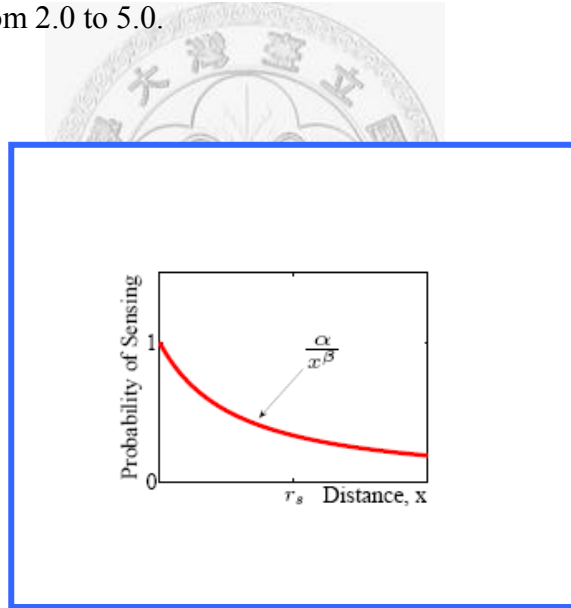


Figure 2.5: Probabilistic sensing model.

Liu and Towsley define the all-sensor field intensity of an arbitrary point p , I_p , to be the sum of the sensing signals of all sensors (s_1, s_2, \dots) at p , i.e.,

$$I_p = \sum_{i=1}^{\infty} S(s_i, p) = \sum_{i=1}^{\infty} \frac{\alpha}{d(s_i, p)^\beta} \quad (2.8)$$

We say that a point p is covered if the all-sensor field intensity at p is greater than or equal to some threshold, θ , i.e., $I_p \geq \theta$. The set of points that are covered according to the above definition is called the *covered region*. Similarly, the complement of the covered region is called the *vacant region*.

C. Information Coverage Model

Recently, Wang et al. [Wan06] calls the disk coverage model as physical model, as well as propose a concept of *information coverage* based on signal estimation theory [WWS05]. Suppose that an event with parameter θ occurs on a space point with the Euclidean distance d to a sensor s_k and θ . For example, θ can be the seismic/acoustic amplitude of a tank. We further assume that the parameter θ decays with distance, and at distance d it is θ / d^α , where $\alpha > 0$ is the decay exponent. The measurement of the parameter, x_k , at a sensor may also be corrupted by an additive noise, n_k . Thus,

$$x_k = \frac{\theta}{d_k^\alpha} + n_k, \quad k = 1, 2, \dots, K. \quad (2.9)$$

A parameter estimator, e.g., a best linear unbiased estimator, can be used to estimate θ based on the measurement x_k , $k = 1, 2, \dots, K$. Let $\hat{\theta}$ and $\tilde{\theta} = \hat{\theta} - \theta$ denote the estimate and the estimation error, respectively. If an estimation error is small, not only the event/target can be claimed to be detected but also the event/target parameter can be obtained within a certain confidence level. Wang et al. define *information exposure* as the probability that the absolute value of the estimation is less than or equal to a predefined constant A , i.e. $\Pr\left\{\left|\tilde{\theta}_k\right| \leq A\right\}$, to measure how well a point is monitored. When it is equal to or larger than a predefined threshold ε ($0 \leq \varepsilon \leq 1$), i.e. $\Pr\left\{\left|\tilde{\theta}_k\right| \leq A\right\} \geq \varepsilon$, this point is said to be

information covered (or (K, ε) -covered) by these K sensors. The information coverage model can be reduced to a sensing disk model if only $(1, \varepsilon)$ coverage is considered for an isotropic sensor. When higher order information coverage is considered, e.g., $(K = 2, \varepsilon)$ coverage, the area covered by these K sensors is no longer a simple union of these K sensors' sensing disks but larger than that. Figure 2.6 illustrates the area that can be $(3, \varepsilon)/(4, \varepsilon)$ -covered with that physically covered when placing sensors at the vertices of a regular triangle/square. It is seen that higher order information coverage increases the area can be covered [Wan 06].

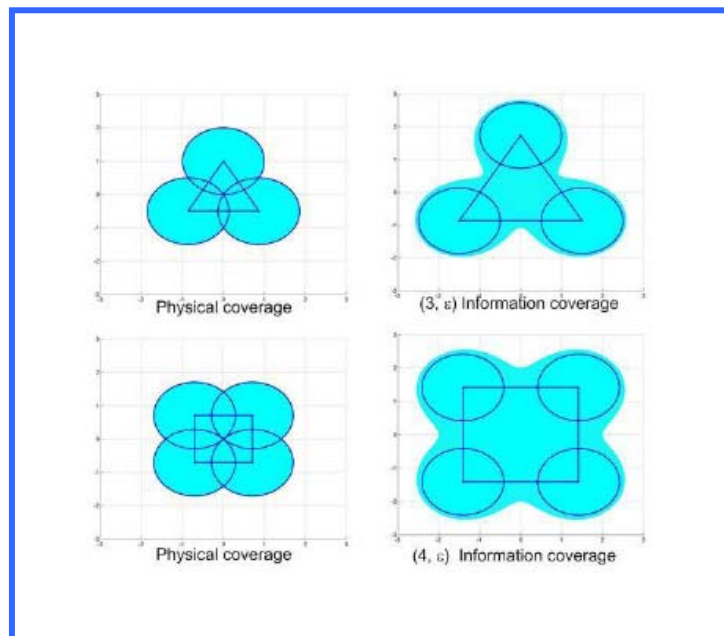


Figure 2.6: Comparison physical and information coverage.

2.1.1.2 Coverage Problems

This section, we survey the main issues of coverage in sensor networks,

including: area coverage, point/target coverage, barrier coverage, energy-efficient area coverage, energy-efficient point/target coverage, set- k -cover problem, and k -coverage problem.

Cardei et al. defines three types of coverage problems in sensor networks [CW04] [CWLP05]: (1) *area coverage*, (2) *point/target coverage*, and (3) *barrier coverage*. As Figure 2.7 shown, area coverage considers the ratio of coverage in whole sensor field [CW06]. Hence complete coverage becomes one of metric of design. Point/target coverage concerns whether the given points are covered. The barrier coverage considers whether existing one path through sensor field such that targets should/shouldn't be detected.

Cardei and Wu [CW06] survey the various *energy-efficient coverage* papers. They classify the literatures as two types: *energy-efficient area coverage* and *energy-efficient point coverage*.

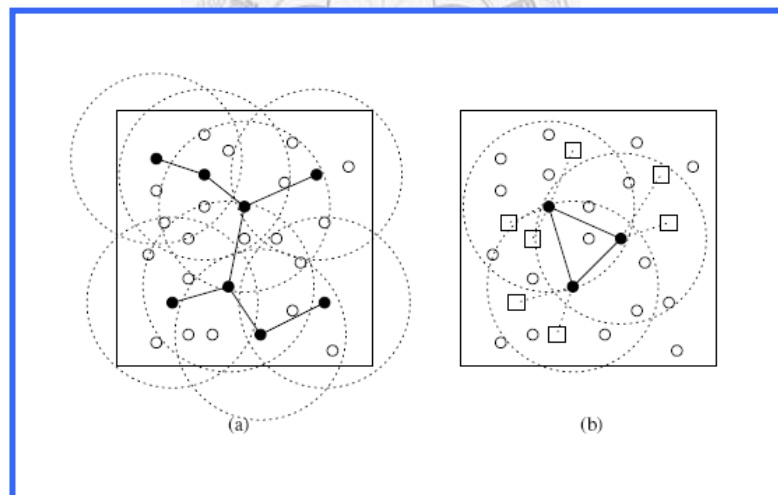


Figure 2.7: (a) Area coverage and (b) Point coverage.

(1) Energy-efficient Area Coverage

Consider a large number of sensors are deployed randomly for area monitoring. The goal of the issue is to achieve an energy-efficient design that

maintains area coverage. As the large number of sensors is deployed to perform monitoring tasks, several papers [CMC02] [SP01] divide these sensor nodes into disjoint sets, such that every set can individually perform the area monitoring tasks. These sets are then activated successively, and while the current sensor set is active, all other nodes are in a low-energy sleep mode. The goal of this approach is to determine a maximum number of disjoint sets, as this has a direct impact on conserving sensor energy resources as well as on prolonging the network lifetime. The problem is indicated as set-k-cover problem in sensor networks.

(2) Energy-efficient Point/Target Coverage

As Figure 2.7(b) shown, a set of sensors are randomly deployed, the aim of point coverage problem is to cover a set of points (small square nodes) in a sensor field.

Cardei and Du [CD05] address the point coverage problem in which a limited number of points (targets) with known locations need to be monitored. A large number of sensors are dispersed randomly in close proximity to the targets and send the monitored information to a central processing node. The requirement is that every target must be monitored at all times by at least one sensor, assuming that every sensor is able to monitor all targets within its sensing range.

One method for extending the sensor network lifetime through energy resource preservation is the division of the set of sensors into disjoint sets such that every set completely covers all targets. These disjoint sets are activated successively, such that at any moment in time only one set is active. As all targets are monitored by every sensor set, the goal of this approach is to determine a maximum number of disjoint sets, so that the time interval between two activations for any given sensor is longer.

In order to considering fault tolerance, several papers investigate

k-coverage in sensor networks design [CW04] [VGD06] [ZDG04]. *K*-coverage means that each point or target in the sensor field is within the sensing range of at least *k* active sensors. With *k*-coverage, the network still works even when any *k*-1 sensors fail at the same time.

2.1.2 QoS in WSNs

Since WSN has a wide variety of applications, it also has many different QoS parameters. Wang, Liu and Yin refer to OSI 7-Layer to defined the QoS parameters for each function layer [WLY06]. We list the QoS parameters [WLY06], shown as Table 2.1.

Table 2.1: QoS parameters and function layers.

| Function layer | QoS parameters |
|--------------------------------|--|
| Application Layer | System lifetime, response time, data novelty, detection probability, data reliability, and data resolution. |
| Transport Layer | Reliability, bandwidth, latency, and cost. |
| Network Layer | Path latency, routing maintenance, congestion probability, routing robustness, and energy efficiency. |
| Connectivity Maintenance Layer | Network diameter, network capacity, average path cost, connectivity robustness, and connectivity maintenance |
| Coverage Maintenance Layer | Coverage percentage, coverage reliability, coverage robustness, and coverage maintenance. |
| MAC Layer | Communication range, throughput, transmission reliability, and energy efficiency, |
| Physical Layer | The capabilities of wireless unit, processor unit and sensing unit |

In this dissertation, we take account of positioning accuracy, which is one

kind of data qualities. Hence, we consider that positioning accuracy belongs to application layer QoS parameter.

Meguerdichian *et al.* [CW06] [MKP01] noted that the coverage of WSN is a measure of the quality of service/surveillance (QoS) of the sensing function. The objective of sensor deployment is to achieve complete coverage, such that each location in the sensor field is within the sensing range of at least one sensor [LC05].

Maleki and Pedram have defined Quality of Monitoring (QoM) as the average of spatial distortion in the reconstructed signal at base stations. They then solved the sensor deployment problem subject to the QoM and network lifetime constraints [MP05].

In some cases, sensor field does not require complete sensing coverage. Yan, He and Stankovic propose the concept of differentiated surveillance service of sensor networks [YHS03]. The authors consider the sensor network, which provides differentiated surveillance service for various target areas with different degrees of security requirements. They developed an adaptable energy-efficient sensing coverage protocol to support differentiated surveillance by sensor networks.

Lu et al. propose a real-time communication architecture, RAP, use velocity monotonic scheduling to prioritize real-time traffic in MAC layer [LBA02].

For providing the target positioning and tracking service, this work refers to the positioning accuracy as the QoS parameter hereinafter. Moreover, the concept of differentiated QoS is also considered to provide weighted discrimination requirement for each ROI in a sensor field.

2.1.3 Energy Efficiency

Energy optimization problems in sensor networks include reducing power

consumption of sensor nodes and extending overall sensor network lifetime. The energy-efficiency is a QoS parameter in each network function layer. Consequently, a lot of research groups are engaged in developing energy-efficient hardware architecture, system software or network protocols to lengthen system lifetime in various aspects [EGH99].

The main components of a sensor node include sensor, data processor, and communication subsystem. Recent studies have demonstrated that data communication is the main consumer of energy in sensor networks. Therefore, several projects have been performed to design power-efficient integrated sensors [PK00] and low-power, low-cost transceiver technologies of sensor node [PME00].

Moreover, link layer techniques usually consider the reliability constraints. The power-efficient MAC protocols minimize the number of times for packet retransmissions, thus reducing the power consumed at both the transmitter and receiver [RSP02] [YHE02].

Furthermore, several power-aware routing protocols have been developed for sensor networks. Almost all of these routing protocols considered energy efficiency as the ultimate objective for maximizing network lifetime [LAR01] [SR02].

From sensors self-organization perspective, some studies have investigated the possibility of partitioning the sensors into many clusters (or covers) such that every sensor cluster provides sufficient service quality [AGP04] [CTL05].

2.1.4 Target Location

Several location systems have been proposed and realized. For instance, the satellite-based Global Positioning System (GPS) is common outdoor location system. However, GPS is not useful in indoor, dense, or harsh environments [RST04].

A. Location Techniques

Hightower and Borriello present three main techniques for location-sensing, including: *triangulation*, *scene analysis*, and *proximity*. Location systems may employ them individually or in mixing [HB01]. We illustrate the classification in Figure 2.8.

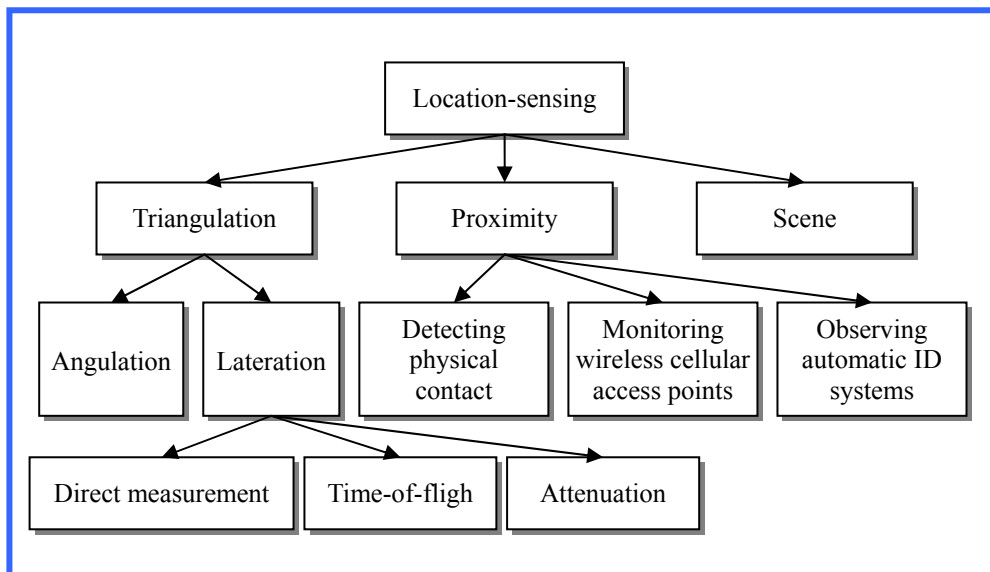


Figure 2.8: The classification of location sensing techniques.

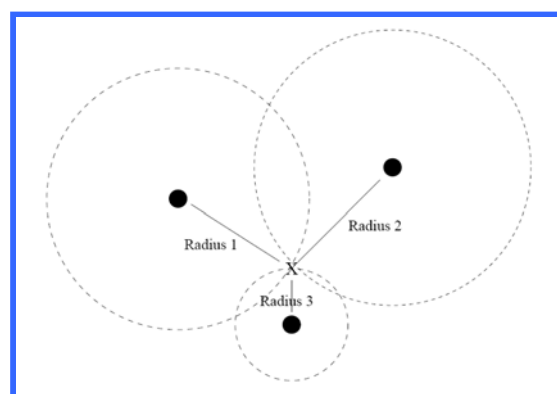


Figure 2.9: Triangulation location-sensing technique.

The triangulation location-sensing technique uses the geometric properties of triangles to compute object locations. Triangulation is divisible into two sub-categories: *lateration*, using distance measurements, and *angulation*, using primarily angle or bearing measurements.

Lateration computes the position of an object by measuring its distance from multiple reference positions. For instance, calculating an object's position in two dimensions requires distance measurements from 3 non-collinear points as shown in Figure 2.9 [HB01].

In general, there are three approaches to measuring the distances required by the lateration technique.

1. **Direct measurement:** distance measurement using a physical action or movement.
2. **Time-of-flight:** it measures distance from an object to a specific point by emitting signal with known velocity (for example, ultrasound, light, or radio) and measuring the traveling time between the object and point P to calculate the distance.
3. **Attenuation:** the intensity of an emitted signal decreases as the distance from the emission source increases. Given an attenuation function, it is possible to estimate the distance between an object and source by measuring the strength of the emission when it returns to source.

Angulation technique is shown in Figure 2.10 [HB01], two dimensional angulation requires two angle measurements and one length measurement such as the distance between the reference points. Sometimes, angulation technique adopts a constant reference vector (e.g. magnetic north) as 0° .

The *scene analysis* location-sensing technique uses features of a scene observed from a particular vantage point to draw conclusions about the location of the observer or of objects in the scene. Usually, the observed scenes are simplified to obtain features that are easy to represent and compare (e.g., the

shape of horizon silhouettes as seen by a vehicle mounted camera). In static scene analysis, observed features are looked up in a predefined dataset that maps them to object locations. In contrast, differential scene analysis tracks the difference between successive scenes to estimate location. Differences in the scenes will correspond to movements of the observer and if features in the scenes are known to be at specific positions, the observer can compute its own position relative to them.

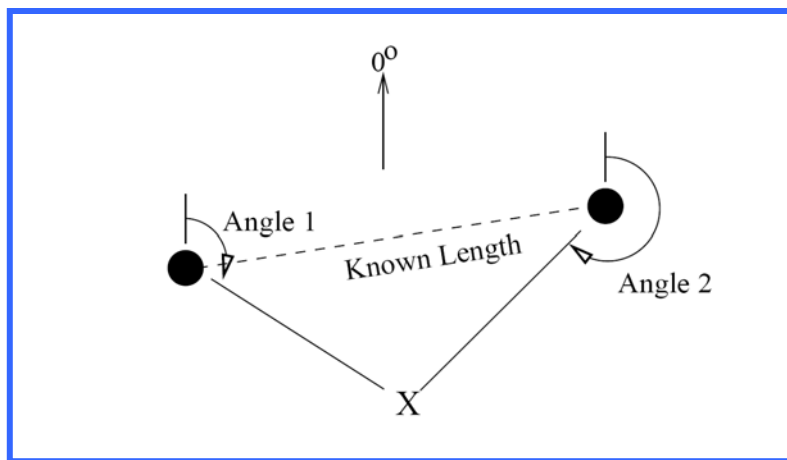


Figure 2.10: An example of 2D angulation technique.

A *proximity* location-sensing technique determines target location when it is “near” a known location. The object's presence is sensed using a physical phenomenon with limited range. There are three general approaches to sensing proximity:

1. **Detecting physical contact:** detecting physical contact with an object is the most basic sort of proximity sensing. Technologies for sensing physical contact include pressure sensors, touch sensors, and capacitive field detectors. Touch Mouse and Contact systems are implemented by this approach.
2. **Monitoring wireless cellular access points:** monitoring when a mobile device is in range of one or more access points in a wireless cellular

network is another implementation of the proximity location technique. Examples of such systems include the Active Badge Location System and the Xerox ParcTAB System, both using diffuse infrared cells in an office environment, and the Carnegie Mellon Wireless Andrew using a campus-wide 802.11 wireless radio network.

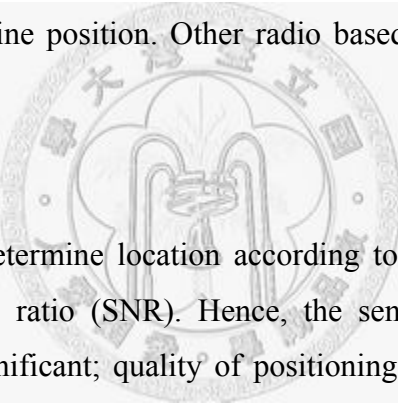
3. **Observing automatic ID systems:** a third implementation of the proximity location-sensing technique uses automatic identification systems such as credit card point-of-sale terminals, computer login histories, electronic card lock logs, and identification tags such as electronic highway e-toll systems, RFID system, etc. While detection devices scan the label, interrogate the tag, or monitor the transaction, the location of the mobile object can be inferred.

In reference paper [RST04], base on wave frequency, the authors classify the indoor location systems into three categories: *infrared*, *ultrasound*, and *radio*.

1. **Infrared:** The Active Badget location system first adopts infrared technique to design an indoor location detection system [RSTU04]. This system provides each person with a badge that periodically emits a unique ID using diffused IR that is received by one of several receivers scattered throughout a building. Badge location is then resolved by proximity to the nearest receiver. In harsh settings, however, the communication environment can be very dynamic, as people move about, smoke or other impurities fill the air, or walls collapse. In such settings, proximity to a single receiver is not sufficiently robust or flexible to provide reliable location detection.
2. **Ultrasound:** Ultrasound based systems also provide location detection based on proximity, but improve accuracy by measuring ultrasound time-of-flight with respect to a reference RF signal. Systems such as the Active Bat or MIT's Cricket compare the arrival time of the two signals

from various known sensors in order to calculate a listener's location. As with the infrared based schemes, current ultrasound based systems are not designed for robustness, since line-of-sight paths may get obstructed or altered in the face of changing room dynamics. In addition, these systems are particularly sensitive to the possible destruction of sensors.

3. **Radio**: Radio waves provide a powerful means of location detection because of their ability to penetrate many types of surfaces and objects, and due to their range, scalability, and maintenance benefits. Rather than using differences in arrival time, as done by ultrasound systems, radio based location detection systems determine location based on received signal strength, predicated on a known signal-to-noise ratio (SNR). RADAR system pre-computes an SNR map for a building. A vector of signal strengths received at various base-stations is compared with this map to determine position. Other radio based systems include SpotON and *Nibble*.



These systems determine location according to the signal strength and a known signal-to-noise ratio (SNR). Hence, the sensitivity for environmental conditions is very significant; quality of positioning is injured by interference frequently. In addition, these location detection systems work well for their designed purposes, but cannot handle significant changes in communications paths or building topology.

Bulusu et al. suggest placing multiple beacons (reference points) in a positioned field with overlapping regions of coverage and transmitting periodic beacon signals. Targets can be localized to the centroid of their proximate reference points [BHE00].

B. Identifying Codes

In paper [KCL98], *identifying codes* first are proposed as a means for

uniquely identifying malfunctioning processors in multiprocessor systems. Such a system can be modeled as a graph $G = (V, E)$, where V is the set of processors and E the set of links between processors. Assume that at most one of the processors is malfunctioning. For testing the system and locating the faulty processor, some processors (which constitute the code) will be selected and assigned the task of testing their neighborhood. Whenever a selected processor (i.e. a codeword) detects a fault, it sends an alarm signal. Hence, the set of detecting processors must be selected carefully to uniquely locate the malfunctioning processor based only on the information which one of the codewords gave the alarm.

Therefore, it is important to obtain results on the complexity of this issue. Given an undirected graph G and integer r (r -cover for each vertex), Charon et al. [CHL03] have proved the decision problem of the existence of an r -identifying code of size at most k codewords in G , is NP-complete for any r . Due to definition of NP-hard, the corresponding optimization problem, i.e., minimizing size of identifying code, is NP-hard.

C. Using Identifying Codes to Target Location Problem

Several researchers adopt concept of *identifying codes* (i.e., *power vectors* in this dissertation) to construct location systems.

(1) CIQ Approach

In papers [CIQ01] [CIQ02], the authors propose *target location* problem in sensor networks. The sensor field is presented as a (two or three dimensional) grid of points; target location refers to the problem of pinpointing a target at a grid point at any point in time. If the coverage areas of multiple sensors overlap, they may all report detecting a target, then the precise location of the target can be determined by overlap of these sensor's detection zones. If every grid point in the sensor field is covered by a unique subset of sensors, we can easily

determine the target occurring and its location according to the set of reporting sensors.

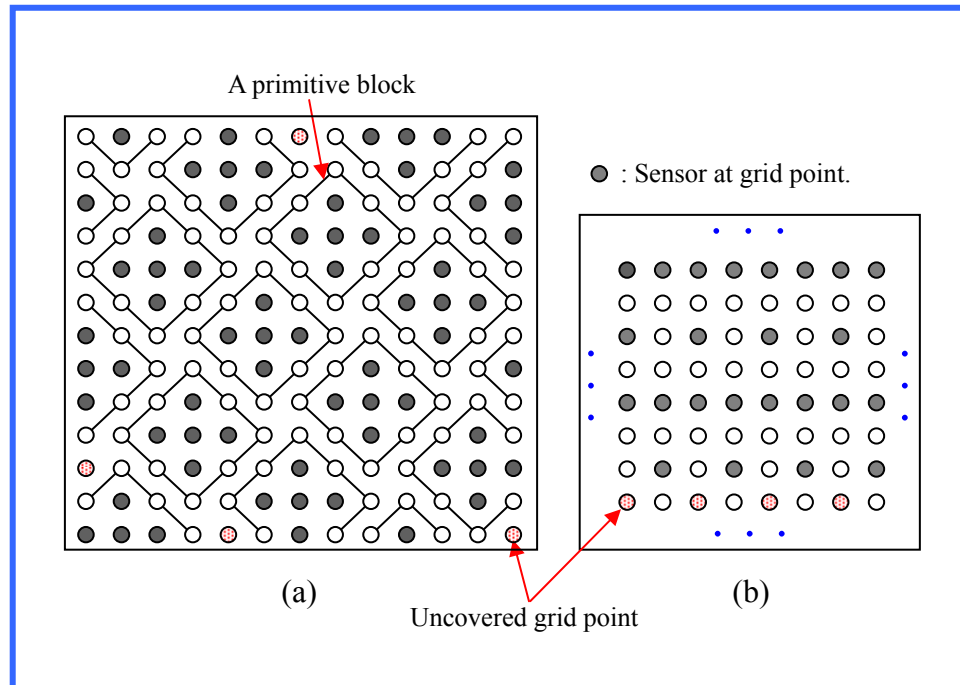


Figure 2.11: Sensor placement based on the coding theory.

Chakrabarty et al. [CIQ02] solve the problem of placing sensors for unique target identification by the theory of identifying codes [KCL98], is called CIQ approach in this dissertation. They first build a primitive block which is completely discriminable by sensors on the block. Each grid point on the field has a unique identifying code which is composed by the reporting messages of these sensors. Then, a larger sensor field can be constructed by tiling primitive blocks on the sensor field. For example, as shown in Figure 2.11(a), a primitive block which contains 13 grid points is deployed 5 sensors with sensing radius 1 on it. A 13 by 13 sensor field is tiled by the primitive block and the field is therefore completely discriminable sensor. Figure 2.11(b) illustrates an 8 by 8 sensor field which is a primitive block is composed by 35 sensors. It can be used to build any $8n$ by $8n$ sensor field ($\forall n \in \mathbb{Z}, n \geq 1$) directly. However, this placement manner can only use in regular sensor field and fixed sensor detection

radius. In addition, the grid points in boundary of sensor fields do not take into consideration. In Figure 2.11(a), there are 4 uncovered grid points, to achieve completely discriminable, it requires to deployed additional sensors for the sensor field. It will harm the applicability of the approach, especially in irregular sensor fields.

(2) ID-CODE Algorithm

Ray et al. apply *identifying code* theory to design a location system in sensor networks [RST04]. They divide a continuous sensor field into a finite set of locatable regions represented by a *designated point* (It is called a service point in the dissertation). Each *designated point* can be identified unambiguously. The system operates in location service mode and periodically broadcasts ID packets from designed sensors. An observer can determine her location due to the unique collection of received ID packets.

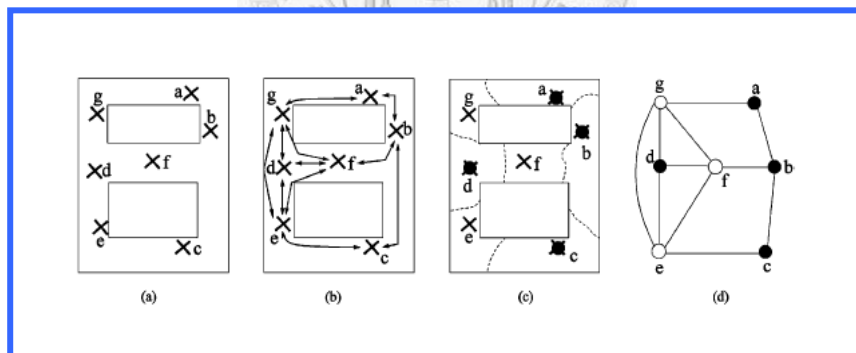


Figure 2.12: The location system proposed by Ray et al.

The sensor network design procedure is as follows. A set of designated points is selected for a given field. Then, based on physical point connectivity, a corresponding graph can be obtained. The vertices are designated points and connectivity between any two points determines whether edge exists. For

example, the points set $P = \{a, b, c, d, e, f, g\}$ on a floor plan is depicted in Figure 2.12(a). Figure 2.12(b) shows the connectivity among these points represented by the arrows.

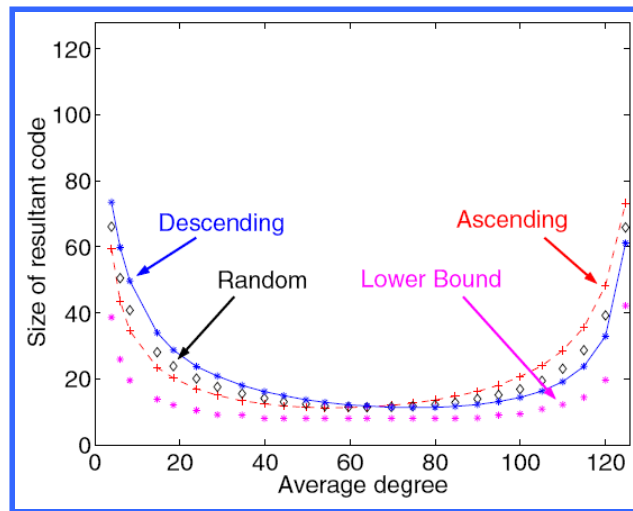


Figure 2.13: Performance of various heuristics for $|V|=128$ vertices graphs.

Ray et al. propose ID-CODE algorithm to deploy sensors and build an identifying code for the given graph as shown in Figure 2.12(d). First, every vertex is deployed a sensor, as a codeword. At each loop of ID-CODE algorithm, one of codeword is checked whether it can be deleted and results in identifying code. The authors suggest three predetermined sequences to visit vertices: random, descending, and ascending orders. The simplest approach is to visit all vertices in *random order*. If the average degree of vertices in the graph is low, the good codewords are likely to have high degree such that the number of codewords required to cover all the vertices is reduced. Hence, the authors propose visit the vertices by *descending order*. On the contrary, if the average degree of vertices in the graph is high, the good codewords are likely to have lower degree. Hence, the authors propose visit the vertices by *ascending order*. The performance of ID-CODE algorithm depends on the sequence of vertices. Therefore, the authors propose a hybrid heuristic for ordering. When the average

degree of graph is greater than half the number of vertices, the descending order of degree is used. Otherwise, the ascending order is used in ID_CODE algorithm.

Figure 2.12(c) and (d) illustrate the sensors deployment, and the corresponding graph. Four sensors are placed at codewords a, b, c, and d. Therefore, each vertex has unique identifying code, e.g., $ID(a)=\{a, b\}$, $ID(b)=\{a, b, c\}$, $ID(c)=\{b, c\}$, $ID(d)=\{d\}$, $ID(e)=\{c, d\}$, $ID(f)=\{b, d\}$, and $ID(g)=\{a, d\}$.

Figure 2.13 shows the size of identifying code of various visiting order in $|V|=128$ vertices graphs with various average degree [RUP03].

(3) Power Vectors

A number of papers investigate the sensor placement problems with grid based sensor field [CIQ01] [DC02] [LRS05] [SSS03]. A grid-based sensor field can be represented as a collection of two- or three-dimensional *grid points*. In this dissertation, we adopt grid based placement method. A set of sensors can be deployed on the grid points to monitor the sensor field.

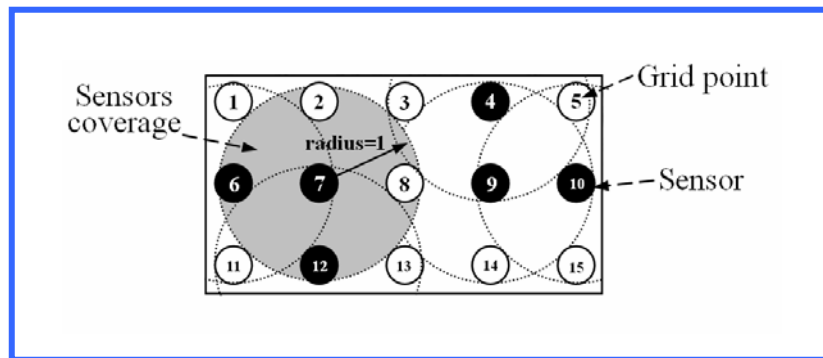


Figure 2.14: Grid-based sensor field and power code.

If any grid point in a sensor field can be detected by at least one sensor, we

call the field is *completely covered*, as shown in Figure 2.14. In this context, a target can be detected at any place in the field [CL04] [LC05].

A *power vector*, which is analog to *identifying code*, is defined for each grid point to indicate whether sensors can cover a service point in a field. As shown in Figure 2.14, the power vector of service point 8 is (0, 0, 1, 1, 0, 0) corresponding to sensor 4, 6, 7, 9, 10, and 12. In a completely covered sensor field, when each service point has a unique power vector, we note the sensor field is *completely discriminated*, as shown in Figure 2.14. In this case, as soon as a target occurs in a grid of the sensor field, it can be located by the back-end according to the power vector of the grid.

2.1.5 Sensor Deployment

Sensor deployment is a key step for sensor network designs, which greatly influences the effectiveness of networks. In general, there are three deployment objectives are major concerned in the sensor deployment phase: reducing the deployment cost, improving quality of surveillance, and prolonging the network [IMP05]. Moreover, there are several factors have to be take account into the sensor placement problem, for instance, the placement methods, the nature of the terrain, types of sensors, network coverage, connectivity, fault tolerance, network architecture, etc. Form perspective of deployment method, we divide related works into two categories: random and controlled deployment methods. In the rest of this subsection, we will review related literatures from this perspective.

A. Random Deployment

(1) To Determine the Drop Strategy

Typically, a WSN used for environmental monitoring is designed to

provide a certain level of QoS, which essentially measures the degree of spatio-temporal accuracy required by the particular application. In paper [LRS05], Leoncini et al. focus on *degree of coverage* (DoC), which evaluates the percentage of a given region sensed by a set of connected sensors. The DoC, as well as other QoS measures provided by a sensor network, depends on the number and positions of the sensors used to monitor the area, and thus on the deployment strategy. In situations where manual deployment is not feasible, random distribution is typical alternative, which drops sensors from a moving vehicle, such as an airplane. Hence, the human operator can only control the *sensor drop point(s)*.

Leoncini et al. consider the following network planning problem. Given a sensor field and a certain DoC requirement, the spatial distribution of the sensors released at a drop point is modeled by a certain probability density function F . Leoncini et al. consider the problem of determining the *optimal drop strategy*, i.e. the strategy such that the DoC requirement is satisfied, and the total number of deployed nodes is minimized. They assume the candidate drop points are arranged in grids of arbitrary side. Further, we assume that probability density function F is the normal distribution. Through analysis and simulation, they can identify the optimal deployment strategy.

(2) To Determine the Critical Sensor Density

An important problem in random distributed sensor networks is to estimate the number of sensors required to achieve complete coverage for a desired region. In [AS03], Adlakha and Srivastava address the problem finding the critical density of sensors for complete coverage by presenting analytical result. They adopt exposure coverage model, that the sensor detects the change in the signal strength over time. In addition, the integrator model be used, i.e., when the total signal energy or exposure exceeds a threshold, the sensor states the target is detected. The authors also model the properties of target as an object moving with constant speed for a distance. Through analysis, they derive an

equation for an effective sensor radius. Then, using this effective radius, they can estimate the critical density for complete coverage in sensor field.

(3) Heterogeneous Sensor Networks

Mhatre et al. consider two types of sensors [MRK05]: cluster header and sensor node. They assume the nodes are organized as a collect of clusters. Cluster heads perform data aggregation for the sensors in the cluster. Sensors are responsible for sensing and relaying data to the nearest cluster head. An aircraft (i.e. a mobile sink) visits the sensor field periodically and gathers data from all of the sensors. Each visiting of the aircraft triggers a sensing and data gathering cycle on the ground during which every sensor node sends a packet to its cluster head and the cluster head directly sends aggregated data to aircraft. The authors also model the cost functions for cluster and sensor node, they consider both hardware and battery cost. Under the coverage and connectivity constraints, the aim of this paper is to determine the sensor deployment parameters such that the network lifetime is enlarged.

(4) Energy Efficiency

Using random deployment, usually a large number of excess sensors are scattered in a monitor field for satisfying quality of surveillance. Many researchers use these redundant sensors to prolong the network lifetime.

Slijepcevic and Potkonjak [SP01] propose a heuristic to organize the sensors into mutually exclusive sets where each set can completely cover the sensor field. These sets of sensors work in turn. Any moment, only one of these sets is active and the other sets operate at sleeping mode. Generally, “mutually exclusive sets of sensors” is also denoted as “cover”. The goal of this work is to maximize the number of covers. They present the problem as set k-cover problem, and propose a heuristic approach for solving the set k-cover problem. In [AGP04], the requirement of complete coverage of cover is relaxed. Abrams et al. design three algorithms to maximize the number of covers.

(5) Target Coverage

Cardei et al. [CWL05] address the target coverage problem in random deployment sensor networks. They consider a large number of sensors with adjustable sensing radius that are randomly deployed to monitor targets. The authors define the Adjustable Range Set Covers (AR-SC) problem. By the property that sensors have adjustable sensing ranges, the objective is to maximize the number of covers and to reduce the radius associated with each sensor, such that targets can be covered by each cover [CWL05]. The authors first introduce the mathematical model for AR-SC problem as the integer programming model, and then propose efficient heuristics by greedy approach to solve it.

(6) Incremental Sensor Deployment

Vieira et al. [VVB04] propose an efficient algorithm for incremental deployment of nodes. A number of sensors randomly deployed in sensor field, suppose amount of sensor, the location and energy level of each sensor are known, the paper discuss what is the minimum number of new sensors that should be added so that it does not lose any covering area? Where should be the new sensors placed? The authors propose algorithm which improves the coverage iteratively for solving the problem [VVB04].

(7) Mobility

Random scattering approach can not always achieve effective coverage, especially if the sensors are overly clustered. There is a small concentration of sensors in certain parts of the sensor field. In [ZC03a], a lot of mobile sensors are scattered in sensor field, Zou and Chakrabarty propose a virtual force algorithm as a sensor placement strategy to improve coverage after initial random deployment. The coverage is extended under the constraint of reducing energy consumption for moving.

B. Controlled Deployment

(1) Grid-based Deployment

Dhillon *et al.* [DC02] [DC03] consider the probabilistic detection model, and propose two polynomial-time sensor placement algorithms to address the homogeneous sensor placement problem. The authors represent the sensor field as a collect of grid points. When a target enters to the sensor field, it will be detected by a set of sensors. They proposed an algorithm optimizes coverage under constraints of a threshold of detection probability and terrain properties. Based on relative measures of security and tactical importance, the preferential coverage of grid points is also modeled in the work.

Dhillon *et al.* [DC02] adopt the probabilistic detection model, which is illustrated in Section 2.1.1, Equation (2.1). The authors assume that knowledge of the terrain is given, e.g, through satellite imagery. Hence, obstacles and the detection probability for each grid point are known. They develop two polynomial-time sensor placement algorithms, MAX_NIN_COV and MAX_AVG_COV, to minimize the number of sensors, and to determine the location of each sensor such that every grid point satisfy the given coverage threshold [DC02]. However, the algorithms can construct a sensor network to support surveillance-only service.

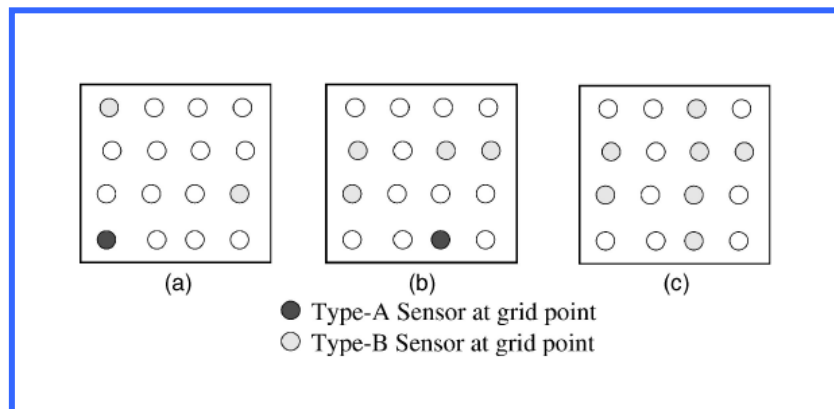


Figure 2.15: Sensor placements for 4 by 4 sensor field. (a) $m=1$, (b) $m=2$, (c) $m=3$.

Chakrabarty *et al.* [CIQ01] [CIQ02] formulate the heterogeneous sensor placement problem in terms of cost minimization under coverage constraints. They first formulate the optimization problem as an integer programming model and then solve problem using the *lpsolve* package. Given the sensor field, two types of sensors (with different cost and radius), parameter $m \geq 1$, they determine the sensors' location such that number of sensor is minimized and each grid point is covered by at least m sensors. The placement examples are shown in Figure 2.15. They also propose a divide-and-conquer approach to cope with the large size problem. In addition, the authors developed sensor placement approaches based on identifying code theorem in graph theory to construct a sensor network providing target location service, which is reviewed in Section 2.1.4.

(2) Well-known Placement Patterns

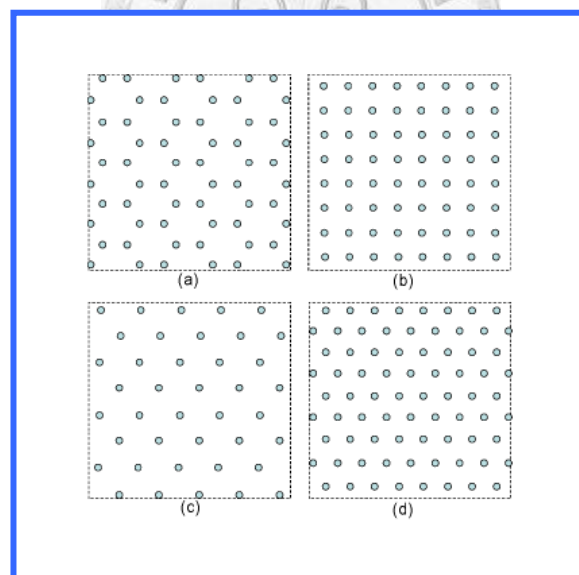


Figure 2.16: Four common regular patterns of deployment. (a) Hexagon. (b) Square. (c) Rhombus. (d) Triangular lattice.

A number of papers adopt the well-known manners about placing disks on the vertices of a triangular lattice or at the centers of regular hexagons, as illustrated in Figure 2.16 [BKX06] [ZH04]. Triangular lattice is optimal pattern in terms of the number of disks needed to achieve full coverage of a plane [BKX06] [ZH04]. On the problem of achieving both coverage and connectivity at the same time, a few results are known in the literature [BKX06]. First, when the communication range r_c is at least twice of the sensing range r_s (i.e., $r_c \geq 2r_s$), then coverage of a region implies connectivity in the sensor network [XWZ05]. Second, while $r_c \geq \sqrt{3}r_s$, then deploying sensors in the triangular lattice pattern provides both coverage and connectivity. And the deployment is optimal in terms of the number of sensors needed. Third, when $r_c = r_s$, a strip-based deployment pattern is near optimal [IKB05].

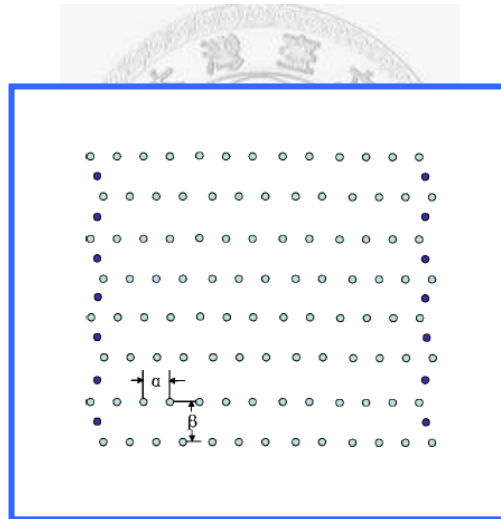


Figure 2.17: Strip-based deployment pattern to achieve coverage and 2-connectivity.

In [BKX06], Bai et al. propose and prove the asymptotic optimality of a deployment pattern to achieve both coverage and 2-connectivity for all values of r_c / r_s . In Figure 2.17, the light-filled dots show the sensor locations that form the horizontal strip, while the dark-filled dots form the two vertical strips. Here, $\alpha = \min\{r_c, \sqrt{3}r_s\}$ and $\beta = r_s + \sqrt{r_s^2 - (\alpha^4 / 4)}$. The vertical strip of sensors may

be removed when $r_c / r_s \geq \sqrt{3}$. The authors also show that the strip based deployment pattern is not only near-optimal but asymptotically optimal for achieving both full coverage and 1-connectivity. Moreover, its optimality holds not only for $(r_c / r_s) = 1$ but for all $r_c / r_s < \sqrt{3}$. They find the ideal ratio between the communication radius and detection radius for sensors. In Figure 2.18, the light-filled dots show the sensor locations that form the horizontal strip, while the dark-filled dots form the one vertical strip. Here, $\alpha = \min\{r_c, \sqrt{3}r_s\}$ and $\beta = r_s + \sqrt{r_s^2 - (\alpha^4 / 4)}$.

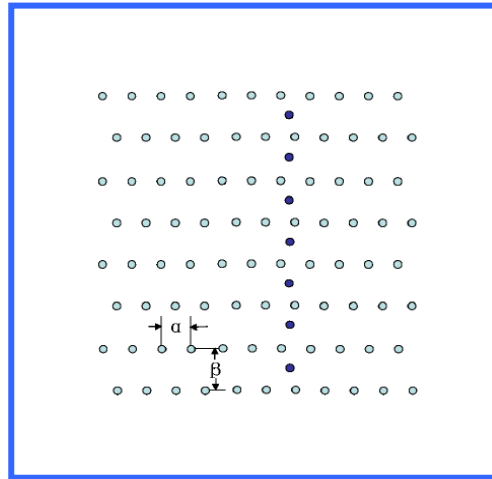


Figure 2.18: Strip-based deployment that is optimal for achieving coverage with 1-connectivity.

(3) Subject to Connectivity and Data Distortion

Ganesan et al. [GCB06] consider the problem of deploying a finite number of sensor nodes in a sensor field, and determine the *communication architecture* among the nodes of the corresponding network. They assume single sink node is responsible for gathering the data from sensors. The authors interest in the relation between the *data reconstruction distortion* that results from the node

placement, and the *power requirements* of data gathering from the sensors. The important goal is to minimize the total energy consumption of data gathering under the sensing distortion constraints [GCB06].

Based on multi-hop forwarding scheme, a routing tree rooted at the sink is built to connect the sensor nodes, and transmit data along the tree [GCB06]. Two kinds of coding schemes include: joint entropy coding with explicit communication and Slepian-Wolf. These coding schemes have tradeoff between computation and communication. The former offers simple coding computation and the latter offers larger communication gains. Ganesan et al. [GCB06] use energy related cost function as objective and consider radius limitation, coverage, and distortion constraints. Due to performance evaluation, the placement outperforms the random placement scheme in terms of power efficiency.

(4) Flat/Hierarchical Architecture

Both flat and hierarchical architectures are commonly adopted for designing a sensor network. Generally speaking, homogeneous sensors are deployed under flat architecture, as well as each sensor takes on collecting data and forwarding it to sinks by multi-hop communication. Hierarchical network architectures use heterogeneous sensors, the networks consists two (or more) layers of sensors. The first layer of wireless sensor network includes several clusters of sensors, which connect to a fixed cluster head. The second layer is a collection of cluster heads, which have more energy and powerful capabilities. These cluster heads with connectivity aggregate data and forward compressed data to sinks. Hierarchical architectures greatly improve the performance of the overall system in terms of throughput, reliability, lifetime, and flexibility [IMP05].

In [IMP05], Iranli et al. discuss the impact of hierarchical network architectures on network lifetime. Given number of cluster heads and total energy budget, the authors address cluster heads deployment, clustering, an

energy allocation to cluster heads such that the network lifetime is maximized. They consider both collinear and planar deployment, and the experimental results showed that 2-level WSN architecture outperforms the flat architecture in terms of lifetime.

(5) Others

Table 2.2: Classification of literatures for the sensor deployment problem.

| R/C | Pre-/Post- | Homo-/Hetero- | Papers | Goals | Considerations |
|-----|------------|---------------|--------------------|-----------------------|---|
| R | Pre- | Homo- | [AS03] | Min sensor density | Coverage |
| R | Pre- | Homo- | [LRS05] | Min sensor density | Coverage, sensor distribution |
| R | Pre- | Hetero- | [MRK05] | Min cost | Two type of sensors, Coverage and connectivity |
| R | Post- | Homo- | [AGP04] [SP01] | Energy-efficiency | k -cover |
| R | Post- | Hetero- | [CWL05] [DVZ06] | Max number of covers | adjustable sensing ranges reduce radius |
| C | Pre- | Homo- | [DC03] [DCI02] | Min number of sensors | Coverage threshold Grid based method |
| C | Pre- | Homo-Hetero- | [CIQ02] | Min number of sensors | Coverage Target location |
| C | Pre- | Homo- | [RST04] | Min number of sensors | Identifying code Target location Robust |
| C | Pre- | Homo- | [CCZ05] | Number of sensor | Lifetime/cost |
| C | Pre- | Homo- | [BXX06] | Number of sensor | Coverage, 2-connectivity |
| C | Pre- | Homo- | [GCB06] | Energy efficiency | Data distortion, connectivity |
| C | Post- | Homo- | [ZC03a] [ZC03b] | Coverage | Mobile sensors Target location |
| C | Post- | Hetero- | [XWH05] | Min cost | Add relay nodes Lifetime Connectivity Sensing/ Relay nodes |
| R/C | Post- | Homo- | [VVB04] | Coverage | Incremental deployment |
| R/C | Post- | Homo- | [KGG06] | Min number of sensor | Coverage Communication efficiency |
| R/C | Post- | Hetero- | [IMP05] | Number of cluster | Lifetime, Two-level |

In [KGG06], the paper proposes the problem that the sensors are dropped randomly. And the probabilistic models for sensor data quality and

communication cost are collected. The proposed algorithm attempts to add the minimal sensors or redeploy the existing sensors subject to coverage and communication efficiency.

Chen et al. [CCZ05] define new performance metric, called lifetime per unit cost. They address the sensor deployment problem to optimize the number of sensors, determine the sensor placement for maximizing the lifetime per unit cost. They propose greedy strategy and numerical approximation to solving the problem.

To obtain the profile of sensor deployment research, in Table 2.2, we arrange the previous survey papers and classify based on the proposed research scope in Section 1.2.



2.2 Lagrangean Relaxation Method

Optimization plays an important role in many application fields. In engineering, for instance, design tasks are routinely cast as optimization problems and algorithms are applied to search for parameters. Actually, optimization techniques could be widely used to address a number of problems found in computer networks, such as traffic routing challenges that have recently emerged with the arrival of connection-oriented architectures. In this dissertation, sensor deployment problems are modeled as optimization problems that are computationally hard and for which no polynomial-time algorithm is known.

Lagrangean methods were originally used in both scheduling and general integer programming problems. However, it has become one of the best tools for solving optimization problems like integer programming, linear programming combinatorial optimization, and non-linear programming [Fis81] [Fis85] [Geo74]. Adopting Lagrangean relaxation as our approach has the following advantages:

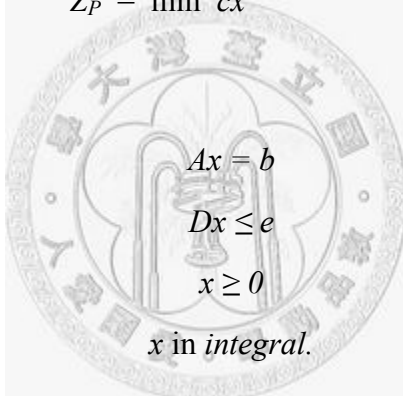
1. The approach is very flexible, since it is often possible to divide and conquer models in several ways and properly apply Lagrangean relaxation to each subproblem.
2. In decomposing problems, Lagrangean relaxation solves primal problems as individual components. Consequently, the solution approach permits us to exploit any known methodology or algorithm to solve the problem.
3. We can use Lagrangean relaxation methods to devise effective heuristic solutions to solve complex combinatorial optimization problems and integer problems.

Lagrangian relaxation also permits us to remove constraints from the original problem and place them in the objective function with associated Lagrangian multipliers instead. The optimal value of the relaxed problem is always a lower bound (for minimization problems) on the objective function value of the problem. By adjusting the multiplier of Lagrangian relaxation, we can obtain the upper and lower bounds of the problem. Although the Lagrangian multiplier problem can be solved in a variety of ways, the subgradient optimization technique is probably the most popular approach.

We now present an example of an optimization problem (P). By relaxing constraint $Ax=b$, the original primal problem (P) is transformed into an LR problem, where $Z_D(u) \leq Z_P$. In other words, the solution of (LR) is a lower bound of the primal problem (P).

$$Z_P = \min cx \tag{P}$$

subject to:



$$\begin{aligned} Ax &= b \\ Dx &\leq e \\ x &\geq 0 \\ x &\text{ in integral.} \end{aligned}$$

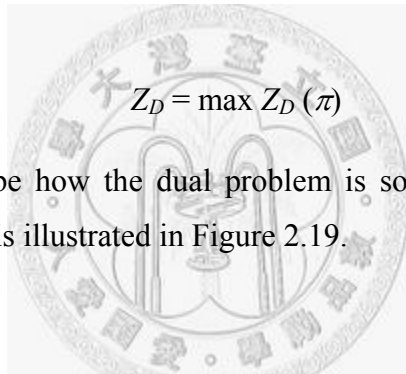
$$Z_D(u) = \min cx + u(Ax-b) \tag{LR}$$

subject to:

$$\begin{aligned} Dx &\leq e \\ x &\geq 0 \\ x &\text{ in integral.} \end{aligned}$$

With respect to the optimization problem (LR), we denote $\pi = (u^1, u^2, \dots) \geq 0$ as the vector of Lagrangian multipliers with respect to relaxed constraints. According to the weak Lagrangian duality theorem, for any $\pi \geq 0$, the objective value of $Z_D(\pi)$ is a lower bound (LB) of Z_P . Thus, the dual problem (D) is

constructed to calculate the tightest LB by adjusting multipliers, subject to $\pi \geq 0$. Then, the sub-gradient method is used to solve the dual problem [HWC74]. Let the vector g be a sub-gradient of $Z_D(\pi)$ at $\pi \geq 0$. In iteration k of the sub-gradient optimization procedure, the multiplier vector is updated by $\pi^{k+1} = \pi^k + t^k g^k$. The step size, t^k , is determined by $t^k = \lambda(Z_p^* - Z_D(\pi^k)) / \|g^k\|^2$, where Z_p^* is an upper bound (UB) of the primal objective function value after iteration k ; and λ is a constant, where $0 < \lambda \leq 2$. To calculate the UB of (P), an algorithm to find primal feasible solutions must be developed. The maximum number of iterations and the *improvement counter* for the problem are decided on a case-by-case basis. We present our experiment settings in Chapter 4 and 5. The parameter λ adopted in the sub-gradient method is initialized to be 2, which is halved when the dual objective function value does not improve for improvement counter iterations.



$$Z_D = \max Z_D(\pi) \tag{D}$$

To better describe how the dual problem is solved, the detailed concept adapted from [Fis81] is illustrated in Figure 2.19.

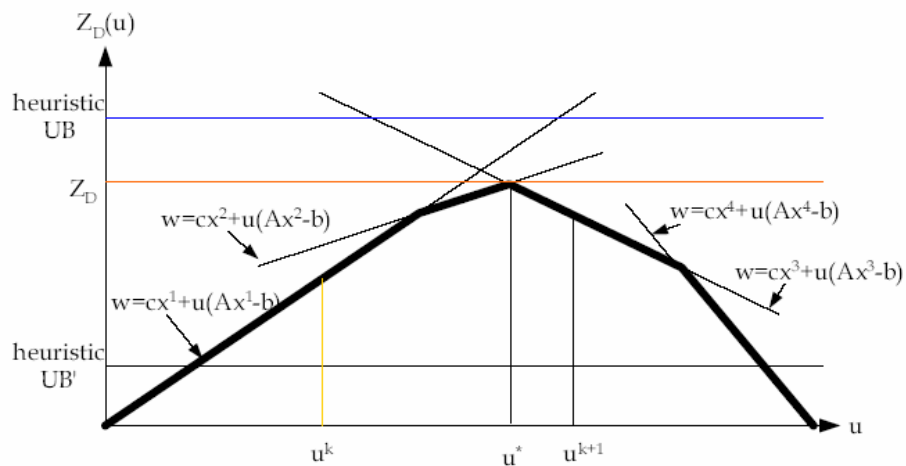


Figure 2.19: The concept of the dual problem.

The overall procedure of the LR approach is shown in Figure 2.20, but the algorithms to find primal feasible solutions must still be developed. After optimally solving the dual problem (D), we get a set of decision variables. However, this solution is not feasible for the primal problem, since some of constraints are not satisfied. Thus, minor modifications of the decision variables must be made to get a primal feasible solution for problem (P). Generally speaking, the UB of problem (P) is the better primal solution, while the solution of problem (D) guarantees the LB of problem (P). Iteratively, by solving the Lagrangean dual problem and getting a primal feasible solution, we get the LB and UB, respectively. So, the error gap between UB and LB, computed by $(UB-LB)/LB*100\%$, illustrates the optimality of the solution. The smaller the gap computed, the better the optimality achieved.

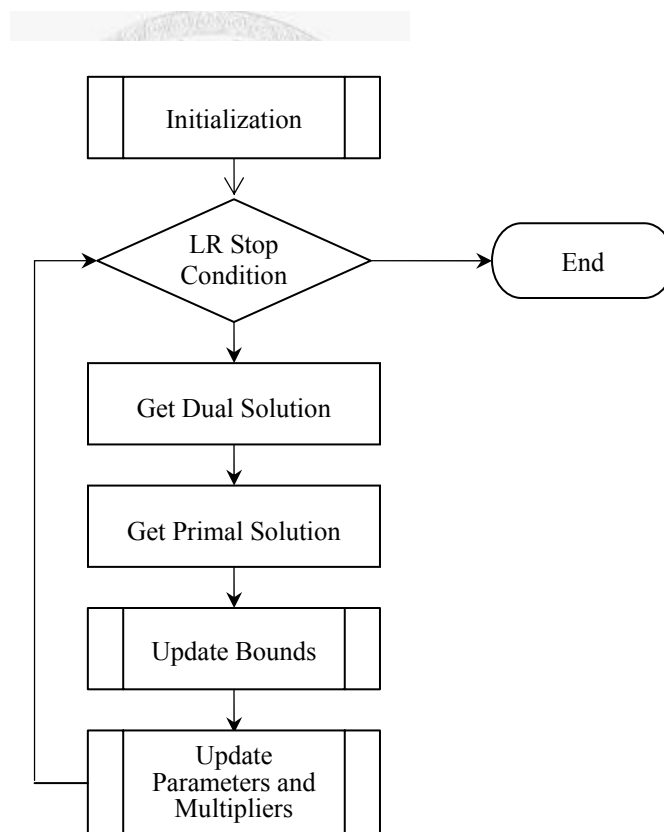


Figure 2.20: The overall procedure of the LR approach.

In reference [Fis85] [Fis81], Fisher either provides a guide to use Lagrangean relaxation, or describes several instances in which Lagrangean relaxation method has been used to solve many well-known hard problems. In this section, we only list partial instances in Table 2.3.

According to the definition of NP-hard [NN04], the optimization problem corresponding to any NP-complete (decision) problem is NP-hard. Hence, the optimization problems corresponding to the well known NP-complete problems listed in Table 2.3 are NP-hard. From Table 2.3, we can claim that Lagrangean relaxation approach can be used for solving NP-hard problem.

Table 2.3: Applications of Lagrangian Relaxation.

| Problem | Complexity | Researchers |
|---|-------------------|----------------------------|
| TSP (Traveling Salesman Problem) | NP-hard | Bazarrá & Goode [BG77] |
| | | |
| Scheduling | | |
| n/m weighted tardiness | NP-hard | Fisher [Fis73] |
| One machine weighted tardiness | NP-hard | Fisher [Fis76] |
| | | |
| GAP (Generalized Assignment Problem) | NP-hard | Fisher & Shapiro [FS74] |
| | | |
| Set Covering | NP-hard | Etcheberry [Etc77] |

2.3 Simulated Annealing Method

Simulated annealing (SA) approach is a generic and probabilistic meta-heuristic for solving the difficult optimization problems. It can solve the combinatorial optimization problems in large search space, namely can find good approximated solutions to the global optimums by randomized heuristic. In 1983, Kirkpatrick, Gelatt, and Vecchi first invented the approach [KGM83]. Afterward, Eglese arranged the previous researchers' studies about simulated annealing and proposed more complete discussions [Egl90].

The idea of the meta-heuristic refers to the annealing technique in metallurgy. Initially, the material is heated to a higher energy state and the structure of atoms is unstuck. The cooling procedure has to be controlled to yield crystal, such that the structure of atoms is tighter and internal energy is lower. The annealing temperature decreases slowly to give atoms more chances for finding configuration with lower internal energy. If cooling is not slow enough, the material may crystallize with defect. Hence, the material is not on an approximated lowest energy state.

For a minimization problem, each feasible solution and corresponding objective value of the problem are analog to a state of material and internal energy on the state. Therefore, the goal of minimization, i.e. getting a feasible solution with global minimum possible, can be analog to yield crystal and decrease defects.

Initially, the SA heuristic selects an initial feasible solution randomly. Then, the following loop of SA heuristic is executed repeatedly. At each state, the heuristic selects a new state x_{i+1} , which is neighbor of current state x_i , and probabilistically decides whether changing current state to the new state or not. The neighbors of each state are dependent on the solution structure of problem; usually it is determined by the user. The transition probability $p = \exp(-\Delta E/T)$ is a function of energy difference $\Delta E = E(x_{i+1}) - E(x_i)$ between new state and

current state, and of parameter T called temperature. For a minimization problem, when $\Delta E \leq 0$, it means the objective value of new state is lower than or equal to current state's, the transition probability $p \geq 1$. Contrarily, $\Delta E > 0$ means the new state has a higher objective value than current state's, the transition probability $p < 1$. To prevent to find a local minimum, the heuristic should have chance to accept the worse new state. Initially, temperature $T=T_0$ is high so that the probability p for accepting the worse solution will be relatively high. On a fixed temperature, $T=T_t$, the algorithm executes $r(T_t)$ iterations at least. The cooling ratio $\alpha < 1$, such that the next temperature T_{t+1} is lower than T_t , and $T_{t+1} = \alpha \times T_t$. The number of iterations on temperature T_{t+1} is $r(T_{t+1}) = \beta \times r(T_t)$, parameter β is greater than one. Hence the number of iterations on fixed temperature should increase slowly, while temperature decreases. The heuristic controls temperature T decreasing slowly, the probability p for accepting the worse solution will reduce, even approaches to zero. When the temperature is lower than the frozen temperature, T_f , the system is frozen, that is, obtaining an approximated optimum solution.

Algorithm 2.1: The skeleton of the SA heuristic.

1. Select an energy function $E(x)$;
2. Select an initial temperature $T_0 > 0$, and $T = T_0$;
3. Select initial number of repetitions on initial temperature, $r(T_0)$;
4. Set repetition counter $t = 0$;
5. **Repeat**
6. Set repetition counter $n = 0$;
7. **Repeat**
8. Generate new state x_{i+1} , a neighbor of x_i ;
9. Calculate $\Delta E = E(x_{i+1}) - E(x_i)$;
10. **If** $\Delta E \leq 0$ **then** $x_i = x_{i+1}$;
11. **else if** random $(0,1) < \exp(-\Delta E/T_t)$ **then** $x_i = x_{i+1}$;
12. $n = n + 1$;
13. **Until** $n = r(T_t)$;
14. $t = t + 1$;
15. $T_t = \alpha \times T_{t-1}$, $r(T_t) = \beta \times r(T_{t-1})$;
16. **Until** stopping criterion, $T_t < T_f$ is true.

To obtain a good approximated optimum solution, users should carefully design cooling schedule, which includes initial temperature T_0 , cooling ratio, α , number of iteration $r(T)$, and stopping criterion. We rewrite the generic heuristic as listed in Algorithm 2.1.





CHAPTER 3 SENSOR PLACEMENT ALGORITHMS FOR ACHIEVING COMPLETE COVERAGE AND DISCRIMINATION



3.1 Overview

Sensor placement strategy depends on the WSN's application. To support surveillance, the network coverage is one of key issues. Furthermore, positioning accuracy must be regarded in sensor deployment phase when WSNs support target location services.

In this chapter, we intend to solve two problems. Both problems have the same goal, which is to optimize the positioning accuracy supporting by sensor networks. First, we solve controlled sensor placement problem under budget and coverage limitations. Next, we consider the random manner, and determine the radius for each sensor such that the positioning accuracy can be improved.

3.2 Problem Description

A grid-based sensor field can be represented as a collection of two- or three-dimensional *grid points* [DCI02]. A set of sensors can be deployed on the grid points to monitor the sensor field. The grid point, which requires the surveillance or positioning service, is also called *service point*. In this chapter, we consider the detection model of a sensor to be a 0/1 coverage model. The coverage is assumed to be full (1) if the distance between the service point and the sensor is less than the detection radius of the sensor. Otherwise, the coverage is assumed to be non-effective (0). If any service point in a sensor field can be detected by at least one sensor, we call the field is *completely covered*, as shown in Figure 3.1. In this context, a target can be detected at any place in the field.

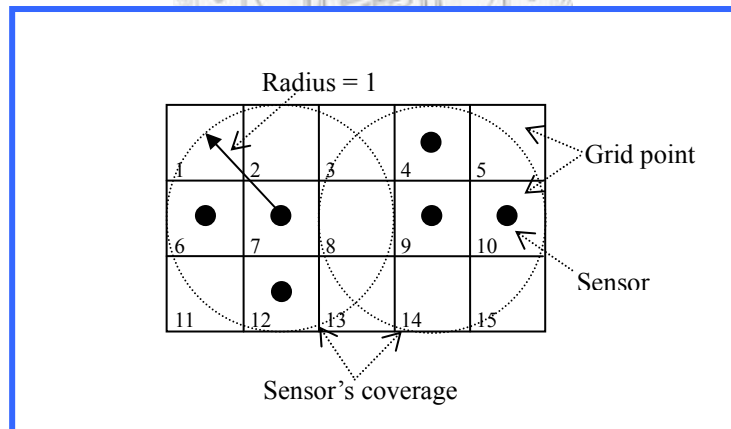


Figure 3.1: A complete covered and discriminated sensor field.

A *power vector* is defined for each service point to indicate whether sensors can cover a service point in a field. As shown in Figure 3.1 the power vector of service point 8 is $(0, 0, 1, 1, 0, 0)$ corresponding to sensor 4, 6, 7, 9, 10, and 12. In a completely covered sensor field, when each service point has a

unique power vector, we note the sensor field is *completely discriminated*, as shown in Figure 3.1. In this case, as soon as a target occurs in a grid of the sensor field, it can be located by the back-end according to the power vector of the service point.

Sometimes, due to some resource limitations, a completely discriminated sensor field cannot be constructed. Consequently, these may lead to wrong determinations, whenever a target occurs at any one of the service points. Positioning accuracy, therefore, becomes a major consideration in solving the problem. *Error distance* is one of the most natural criteria to measure positioning accuracy. The error distance of two indistinguishable service points is defined as the Euclidean distance between them. Hence, when complete discrimination is impossible, the goal of this problem is to minimize the maximum error distance, that is, to optimize the positioning accuracy.



3.3 Sensor Placement Problem for Achieving Complete Coverage and Discrimination

3.3.1 Mathematical Model

The sensor placement problem is formulated herein as a combinatorial optimization problem. Depends on the cost limitation, the formulation can plan a completely discriminable sensor network or a discriminable sensor network with a minimum error distance. If the minimum Hamming distance of the power vectors associated with any pair of grid points doesn't equal to zero, the sensor network is completely discriminable. Otherwise, it leads to an error distance for the target positioning. Hence, the objective of the formulation is to minimize the maximum error distance for the sensor network. The problem is, therefore, defined as a min-max model.

Given Parameters:

- A : Index set of service points in the sensor field.
- B : Index set of the sensors' candidate locations.
- C : Index set of sensor cost.
- G : Budget limitation.
- \bar{K} : A large number.
- r_k : Detection radius of sensor located at k , $k \in B$.
- d_{ij} : Euclidean distance between location i and j , $i, j \in A$.
- c_k : The cost of the sensor allocated at location k ; $k \in B$, $c_k \in C$.

Decision Variables:

- $v_i = (v_{i1}, v_{i2}, \dots, v_{ik})$: A power vector of location i , where v_{ik} is 1 if the target at location i can be detected by the sensor at position k and 0 otherwise, $i \in A$, $k \in B$.
- y_k : 1, if a sensor is allocated at location k and 0 otherwise, $k \in B$.

Objective Function:

$$Z_{IP3.1} = \min_v \max_{\forall i, j \in A} \frac{d_{ij}}{1 + \bar{K} \sum_{\forall k \in B} (v_{ik} - v_{jk})^2} \quad (IP3.1)$$

subject to:

$$v_{ik} d_{ik} \leq y_k r_k \quad \forall k \in B, i \in A, i \neq k \quad (3.1)$$

$$\frac{d_{ik}}{r_k} > y_k - v_{ik} \quad \forall k \in B, i \in A, i \neq k \quad (3.2)$$

$$v_{kk} = y_k \quad \forall k \in A \cap B \quad (3.3)$$

$$\sum_{\forall k \in B} c_k y_k \leq G \quad (3.4)$$

$$\sum_{\forall k \in B} v_{ik} \geq 1 \quad \forall i \in A \quad (3.5)$$

$$v_{ik}, y_k = 0 \text{ or } 1 \quad \forall k \in B, i \in A. \quad (3.6)$$

When $\sum_{\forall k \in B} (v_{ik} - v_{jk})^2 = 0$, objective function (IP3.1) introduces a penalty d_{ij} , $d_{ij} \geq 1$. As $\bar{K} \rightarrow \infty$ and $\sum_{\forall k \in B} (v_{ik} - v_{jk})^2 > 0$, $Z_{IP3.1}$ introduces a penalty $d_{ij}/(1 + \bar{K})$ which approaches zero.

Constraints (3.1), (3.2), and (3.3) require the relationship between sensor transmission radius r and detection distance d_{ik} . If a target appears at service point i and the service point is inside the coverage of sensor k , the target should be detected by the sensor if it is available. Constraint (3.4) states that the total deployment cost of sensors must be limited by cost G . Constraint (3.5) is the complete coverage limitation. It guarantees that any service point in the field will be covered by at least one sensor. Constraint (3.6) is an integer constraint.

3.3.2 Algorithm

Simulated annealing (SA) is a highly reliable method for solving hard

combinatorial optimization problems. The concept of SA is applied to derive an efficient method for solving the problem approximately.

Algorithm 3.1: Simulated-annealing-based pseudo code for sensor placement.

```

1. Deploy sensors on all grid points;  $E_{old} \leftarrow \infty$ ,  $E_{min} \leftarrow E_{old}$ .
2.  $t \leftarrow t_0$ ,  $b \leftarrow b_0$ .
3. While  $t > t_f$  do
4.   Repeat Steps (5)~(27)  $b$  times.
5.     If budget constraint is satisfied then goto Step (17).
6.     Configure a new deployment by removing a sensor randomly.
7.     If coverage constraint is violated then goto Step (16).
8.     Calculate  $E_{new}$  and  $\Delta E = E_{new} - E_{old}$ .
9.     Generate a random number  $p$ ,  $0 < p < 1$ .
10.    If  $\exp(-\Delta E/t) \leq p$  then goto Step (16).
11.    Accept the new deployment;  $E_{old} \leftarrow E_{new}$ .
12.    If  $E_{min} \leq E_{old}$  then goto Step (4).
13.     $E_{min} \leftarrow E_{old}$ ; save the new configuration as the best solution.
14.    If  $E_{min}$  is a desired solution then goto Step (30).
15.    Goto Step(4).
16.    Recover the action in step (6);
17.    Configure a new deployment by moving one sensor's location randomly.
18.    If coverage constraint is violated then goto Step (26).
19.    Calculate  $E_{new}$  and  $\Delta E = E_{new} - E_{old}$ .
20.    Generate a random number  $p$ ,  $0 < p < 1$ .
21.    If  $\exp(-\Delta E/t) \leq p$  then goto Step (26).
22.    Accept the new deployment;  $E_{old} \leftarrow E_{new}$ .
23.    If  $E_{min} \leq E_{old}$  then goto Step (4).
24.     $E_{min} \leftarrow E_{old}$ ; save the new configuration as the best solution.
25.    If  $E_{min}$  is a desired solution then goto Step (30).
26.    Recover the action in step (17).
27.  End_repeat
28.   $t \leftarrow t \times \alpha$ ,  $b \leftarrow b \times \beta$ .
29.  End_while
30.   $Z_{IP3.1} \leftarrow E_{min}$ .

```

Here, the cooling schedule of the algorithm is stated briefly. Initially, we assume the sensors are deployed at all grid points. In each loop, an attempt is made to remove one sensor if the cost constraint is not met. Otherwise, an attempt is made to move a sensor to another randomly chosen position.

Moreover, the stopping criterion is modified to improve efficiency. Besides reaching the frozen temperature t_f , when both complete coverage and discrimination are achieved, that is $Z_{IP1} = I / (I + \bar{K})$, the procedure will also be stopped. The solution with complete coverage and discrimination may not be optimal. However, the solution is the desired solution to this problem.

Algorithm 3.1 shows a pseudo code of the algorithm. The energy, E , is defined as follows:

$$E = \max_{\substack{\forall i, j \in A \\ i \neq j}} \frac{d_{ij}}{1 + \bar{K} \sum_{\forall k \in B} (v_{ik} - v_{jk})^2} \quad (3.7)$$

3.3.3 Computational Results

This section presents the computational results. First, the performance of the proposed algorithm is evaluated when small sensor fields are deployed. The purpose of the experiment is to examine whether the algorithm can find the optimal solution under a minimum cost constraint. Then, the performance results in the case of larger sensor fields are presented under various cost constraints.

The parameters of the cooling schedule are $\alpha = 0.75$ and $\beta = 1.3$. The initial values of b_0 and t_0 are $5n$ and 0.1 , respectively; and n is the amount of grids in the sensor field. The frozen temperature, t_f , is set to $t_0/30$. \bar{K} is 10000 and the cost of sensor, c_i , $\forall 1 \leq i \leq n$, is set to one.

As all sensors have same cost, the cost constraint, Constraint (3.4), can be expressed as a limit on the number of sensors. This section uses a normalized term, sensor density, in the constraint. Sensor density is defined as follows:

$$\text{Sensor density (\%)} = \left(\sum_{\forall k \in B} y_k / n \right) * 100\%$$

A. Experiment I

Experiment I evaluates the performance of the proposed algorithm for smaller rectangular sensor fields that have no more than 30 grid points. The results are compared with those obtained in an exhaustive search.

First, we find a minimum sensor density for a complete covered and discriminated sensor field. Then, an attempt is made to obtain the same result by using the proposed algorithm under a sensor density constraint.

Table 3.1 shows the results. In all cases, the proposed algorithm achieves the same deployment of sensor fields with a minimum sensor density. The required sensor density ranges between 40% and 45%, except for the case of the 4×3 rectangular sensor field. The exhaustive search for the 10×3 sensor field exceeds 65 minutes. However, the proposed algorithm finds the solution in 0.1 second.

Table 3.1: Comparison between exhaustive search and the SA algorithm.

| Area | # of sensors | | Sensor density | Area | # of sensors | | Sensor density |
|------|--------------|----|----------------|------|--------------|----|----------------|
| | Opt. | SA | | | Opt. | SA | |
| 3x3 | 4 | 4 | 44.44% | 6x4 | 10 | 10 | 41.67% |
| 4x3 | 6 | 6 | 50.00% | 6x5 | 12 | 12 | 40.00% |
| 4x4 | 7 | 7 | 43.75% | 7x3 | 9 | 9 | 42.86% |
| 5x3 | 6 | 6 | 40.00% | 7x4 | 12 | 12 | 42.86% |
| 5x4 | 8 | 8 | 40.00% | 8x3 | 10 | 10 | 41.67% |
| 5x5 | 10 | 10 | 40.00% | 9x3 | 11 | 11 | 40.74% |
| 6x3 | 8 | 8 | 44.44% | 10x3 | 12 | 12 | 40.00% |

B. Experiment II

In this experiment, two larger sensor fields with 10×10 and 30×30 grid points are considered. The radius of the sensor is one. The values of $Z_{IP3,1}$ are

determined by various sensor densities. The results obtained by the proposed algorithm are compared with the best solution obtained by the random placement approach. The best solution that has a minimum objective value is found in 1000 arbitrarily generated solutions.

The time needed to compute 1000 arbitrary solutions for the 30 by 30 sensor field with 70% sensor density is 700 seconds. There are only a couple of minutes for the proposed algorithm. Figure 3.2 and Figure 3.3 show that the required density for the desired solution obtained using the proposed algorithm ranges from 40% to 45%. This result is consistent with Table 3.1. In contrast, the random placement approach is associated with a relatively high density (54% and 69% for Figure 3.2 and Figure 3.3, respectively). The proposed algorithm is, therefore, very effective and scalable.

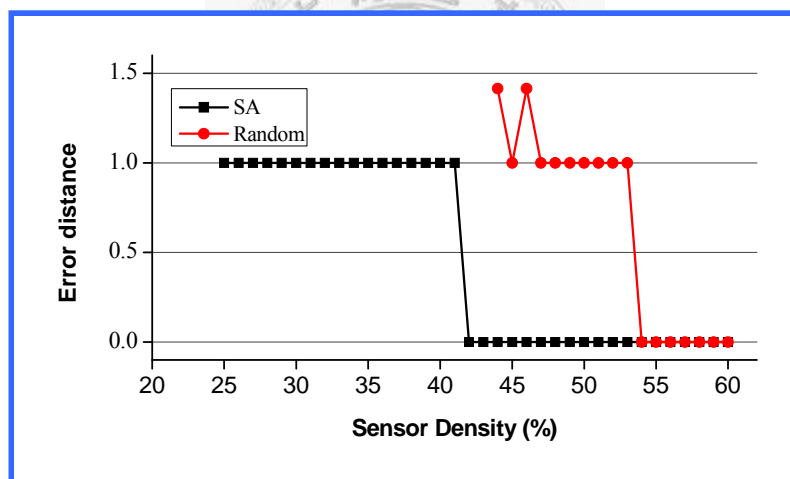


Figure 3.2: Error distance vs. sensor density. (10x10, R=1)

The proposed algorithm can achieve completely covered placement at a very low sensor density. The minimum required sensor densities in Figure 3.2 and Figure 3.3 are only 25% and 24% respectively. The results are very close to the theoretical lower bound. (When the sensor radius is 1, a sensor can cover 5 grid points. Hence, the lower bound of the sensor density for complete coverage is 20%). However, with the random placement approach, the required density

for a completely covered placement is very high (44% and 63% for Figure 3.2 and Figure 3.3, respectively). The results show that the probability of finding the feasible solution using the random placement approach is very low when the sensor field area increases.

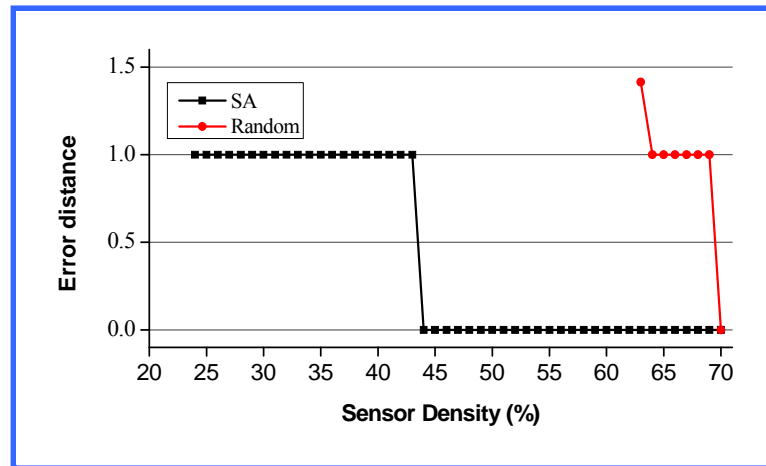


Figure 3.3: Error distance vs. sensor density. (30x30, R=1)

Figure 3.4 and Figure 3.5 show the required density for the case of radius 2. The desired solution obtained by the proposed algorithm ranges from 25% to 30%. In contrast, the random placement approach is associated with a relatively high density (40% and 52% for Figure 3.4 and Figure 3.5, respectively). The results confirm that the proposed algorithm is very effective and scalable again.

The minimum required sensor densities to achieve complete covered placement for the SA algorithm in Figure 3.4 and Figure 3.5 are only 10% and 11% respectively. The results are very close to the theoretical lower bound. (When the sensor radius is 2, a sensor can cover 13 grid points. Hence, the lower bound of the sensor density for complete coverage is 7.69%). However, with the random placement approach, it requires very high density for a complete

covered placement (40% and 52% for Figure 3.4 and Figure 3.5, respectively). The results show that the random placement approach becomes more difficult to get feasible solutions when the area of the sensor field increases.

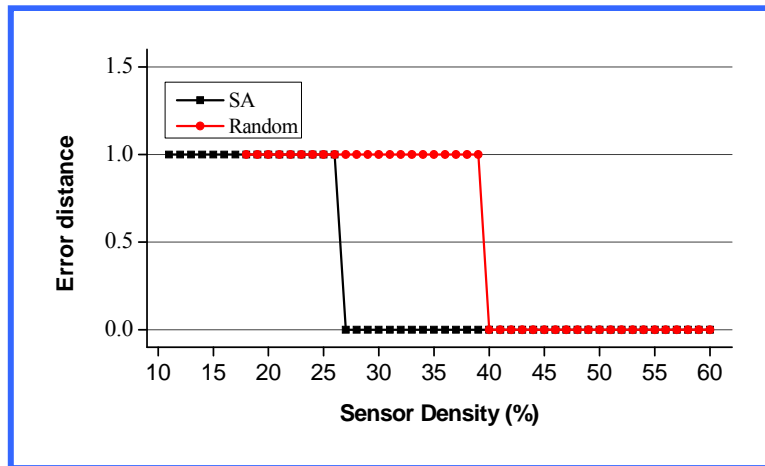


Figure 3.4: Error distance vs. sensor density. (10x10, R=2)

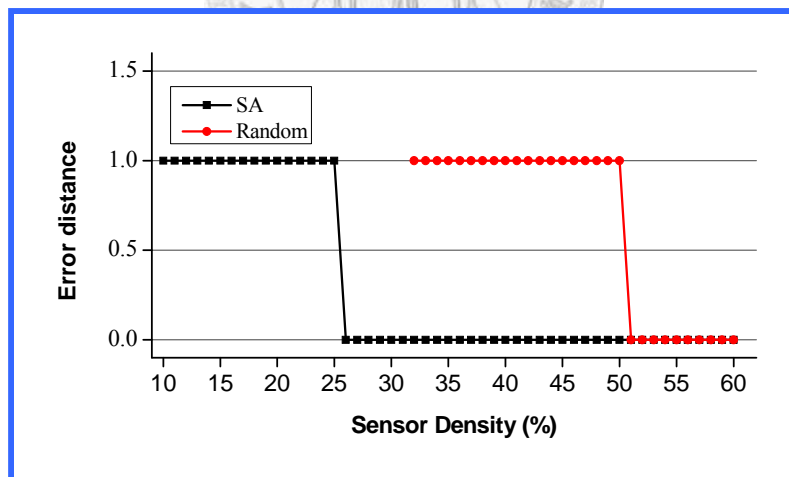


Figure 3.5: Error distance vs. sensor density. (30x30, R=2)

From Figures 3.2 to 3.5 indicate that the placement of sensors by the proposed algorithm has a minimum error distance, 1, when the sensor density is

insufficient. The random placement approach cannot achieve the same result.

3.3.4 Concluding Remarks

This section considers the sensor placement problem for locating targets under cost constraints. We first formulate this problem as a min-max mathematical optimization model where the positioning accuracy is the objective. Then, the simulated annealing-based algorithm is developed to solve the optimization problem. The experimental results show that the proposed algorithm can efficiently obtain a high-quality solution. Additionally, the proposed algorithm is very effective, scalable, and robust.



3.4 Consider Adjustable Detection Radius on Random Sensor Placement Problem

Several papers about sensor deployment consider the sensors to be scattered in the sensor field, and locations of sensors are determined randomly. Hence, determining the number of sensors for achieving ratio of coverage is one of important issues in pre-deployment phase. It motivates us to investigate whether the positioning accuracy can be improved by adjusting detection radius of each sensor when the location of each sensor is given by random manner. This section, we develop another simulated annealing algorithm and many sets of experiments to study the problem.

3.4.1 Simulated Annealing Based Algorithm

In this section, the cooling schedule of the algorithm is stated briefly. In this study, the set of candidate radius, R , as well as the number of sensors, N , are given. Initially, all the sensors are randomly deployed in sensor field. We separate those samples, whose configurations satisfy complete coverage, as well as to determine radius of each sensor by the proposed algorithm.

Algorithm 3.2 shows a pseudo code of the simulated-annealing-based algorithm. The energy function is still Equation (3.7), which means the maximum error distance.

Initially, sensors are randomly deployed under the number limitation. The radius of all sensors is set to an *initial radius*, which belongs to the candidate radius set, R . If the configuration still doesn't satisfy the coverage constraint, it should be abandoned.

In each loop, the solution configuration is randomly altered by one of following two actions: increasing one sensor's radius, or reducing one sensor's

radius. Each action only changes the status of one sensor. The solution with the minimum energy is saved as the best found solution. The terminate condition of the algorithm is either reaching the frozen temperature t_f or getting the desired solution, *i.e.* complete discrimination. The energy of best found solution, Z_{min} , can be used to verify whether the latter condition is reached or not. When $Z_{min} < 1$, the solution configuration is a completed discriminated sensor network, the algorithm should be terminated. Otherwise, the proposed algorithm will determine the new detection radius of each sensor to minimize the maximal error distance of the configuration until the terminated temperature is satisfied.

Algorithm 3.2: The pseudo code of simulated annealing algorithm.

| | |
|-----|--|
| 1. | According to the budget, randomly deploy sensors in the sensor field. |
| 2. | The detection radius of all sensors is set to the <i>initial radius</i> , which belongs to R . |
| 3. | If the initial configuration doesn't satisfy the complete coverage constraint then goto step (23). |
| 4. | Calculate initial energy E_{old} , $E_{min} \leftarrow E_{old}$; save the initial configuration as the best solution. |
| 5. | If the initial configuration is a desired solution, $E_{min} < 1$, then goto step (22). |
| 6. | $t \leftarrow t_0$, $b \leftarrow b_0$. |
| 7. | While $t > t_f$ do |
| 8. | Repeat step (9)~(19) b times. |
| 9. | Randomly choose one of sensors; alter its radius by increasing or decreasing one unit. |
| 10. | If coverage constraint is violated then recover the action in step (9); goto Step (19). |
| 11. | Calculate E_{new} for the new configuration. |
| 12. | Evaluate the difference in energy between the two configurations, $\Delta E \leftarrow E_{new} - E_{old}$. |
| 13. | Generate a random number p , $0 < p < 1$. |
| 14. | If $\exp(-\Delta E/t) \leq p$ then recover the action in step (9) ; goto Step (19). |
| 15. | Accept the new solution; $E_{old} \leftarrow E_{new}$. |
| 16. | If $E_{min} \leq E_{old}$ then goto Step (19). |
| 17. | $E_{min} \leftarrow E_{old}$; save current configuration as the best solution. |
| 18. | If the best solution is a desired solution, $E_{min} < 1$, then goto step (22). |
| 19. | End_repeat |
| 20. | $b \leftarrow b * \beta$, $t \leftarrow t * \alpha$ |
| 21. | End_while |
| 22. | $Z_{IP} \leftarrow E_{min}$. |
| 23. | End |

3.4.2 Computational Results

This section presents the results of experiments. The experiments include two steps: Step1, the initial configurations are deployed by random approach, and we observe the average ratio of coverage and discrimination. Step 2, the radii are adjusted by the proposed algorithm as well as we evaluate the results.

We develop many sets of experiments with varied number of sensors and initial radius. The parameters of the cooling schedule are $\alpha=0.7$, $\beta=1.3$. The initial value of b_0 and t_0 are 2000 and 1.0, respectively. The frozen temperature t_f is $t_0/2000$. Furthermore, the sensor field is 150 (10 by 15) service points. Sensor cost, c_j , is set to one, and the set of candidate radius R is $\{1, 2, \dots, 8\}$.

A. Random Placement

We randomly scatter lots of sensors in a sensor field with 150 service points. Let radius r_k ranging from 5 to 8 length units; and we calculate the objective values of the configurations satisfying complete coverage by initial radius. Each experiment with specific number of sensors and initial uniform radius repeats 1000 times, and the statistical results are shown in Figures 2 to 5. The *sensor density*, d , which equals the number of sensors divided by the number of service points ($d=N/|A|$), is replaced the number of sensor to clearly show the results in Figures 3.6 to 3.9.

Figure 3.6 shows the average objectives decrease monotonously when the sensor density increases. The ideal uniform radius is 5, 6, or 7, whose average objectives are lower than radius 8.

Figure 3.7 depicts the best found objective in 1000 samples. The radius 8 is still the worst one, when sensor density is about 60%, the completely discriminated configuration just probably occurs. Therefore, due to Figures 3.6 and 3.7, radii 5, 6, and 7 are still ideal choices for minimizing the maximum error distance by random placement.

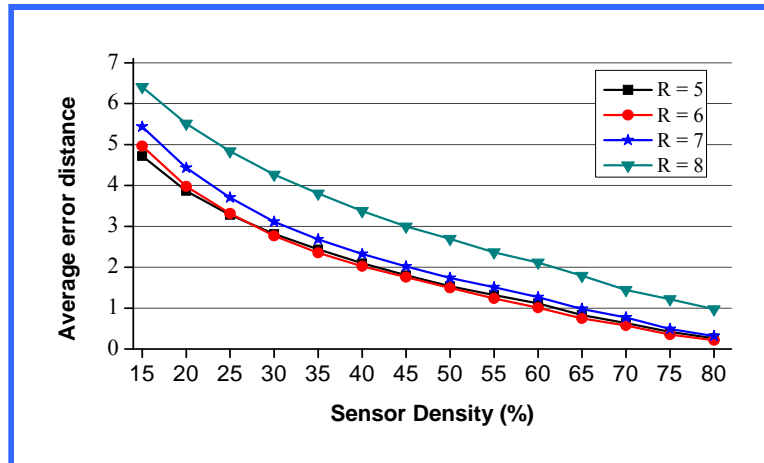


Figure 3.6: Average error distance of the random approach.

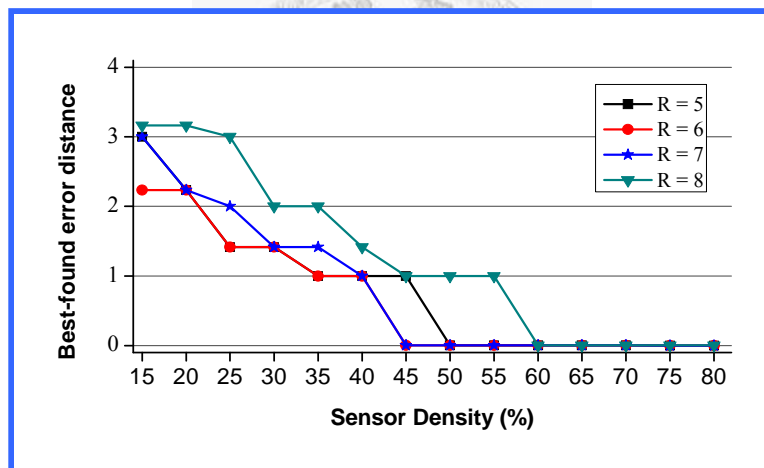


Figure 3.7: Minimum error distance of the random approach.

Figure 3.8 shows the probability of achieving complete coverage for varied sensor density. It is reasonable that the probability of achieving complete coverage will increase if the detection radius increases. But, with adequate sensor density, over 40%, the complete coverage almost can be achieved with 100% probability for each kind of radius. For achieving complete coverage, using uniform radius 5 is not a good choice.

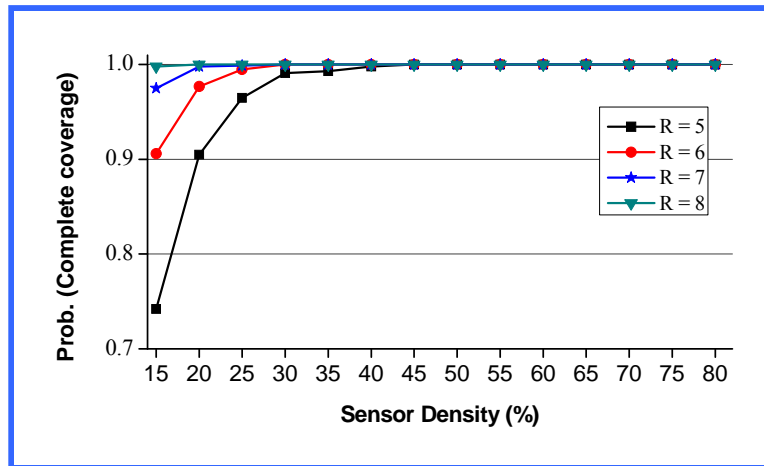


Figure 3.8: Probability to achieve complete coverage. (Random approach)

Figure 3.9 depicts the probability of completely discriminated configuration for varied sensor density by random deployment. For achieving complete discrimination, radius 6 is the best choice in the candidate radius set.

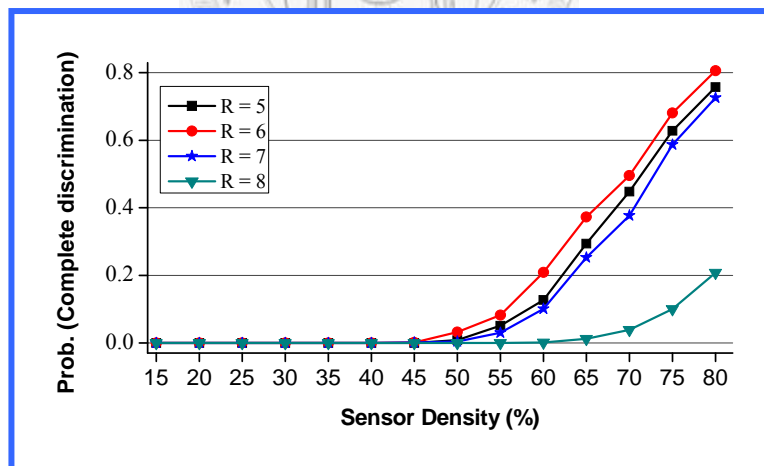


Figure 3.9: Probability to achieve complete discrimination. (Random approach)

Synthetically, the radius 8 is the best choice for coverage. In addition, radius 5 and 6 are ideal choices for minimizing average error distance and

maximizing probability of discrimination, respectively. Radius 5 is worse than radii 6 and 7 for satisfying coverage constraint. We will adopt radius 5 as the initial radius of random deployed configuration to check the coverage constraint. And we keep the samples with complete coverage to adjust individual radius in the next experiment.

B. Adjusting Radii

Subsequently, for each kind of sensor density, we randomly select 100 configurations with complete coverage to test Algorithm 3.2. The simulated annealing based algorithm generates new configurations with various radii. A lot of statistical data are obtained, including average objective values, probability of complete discrimination, and the best found solution. The statistic results are compared with random placement with uniform radius 5, and are depicted from Figures 3.10 to 3.12.

Figure 3.10 shows the average objective values for random placement with uniform radius 5, and adjusting radii by Algorithm 3.2. Obviously, after radii are adjusted, the average objective values are reduced about 50%. The average objective value of original configurations will be lower than 1 when the given of sensor density is over 65%. After the radii are adjusted, the average objective will be lower than 1 when the given sensor density is over 40%.

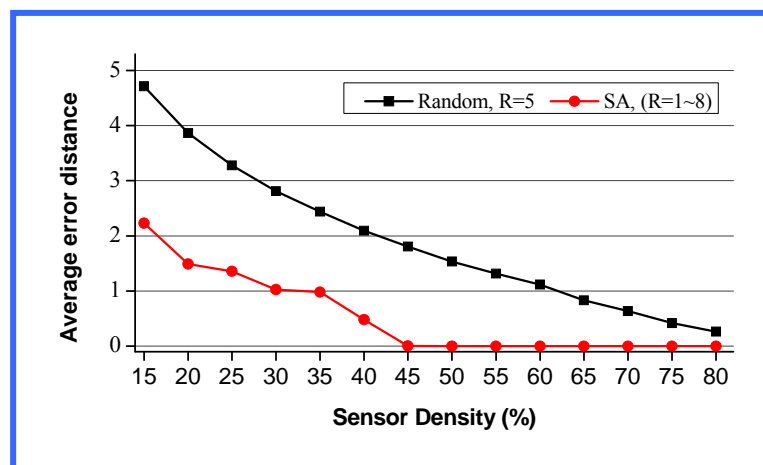


Figure 3.10: Average error distance.

Figure 3.11 depicts the best found case for initial uniform radius 5 and varied radii determined by Algorithm 3.2. The best objective value of original configurations will be lower than 1 with 50% sensor density. After the radii are adjusted, the best objective will be lower than 1 when the sensor density is over 30%.

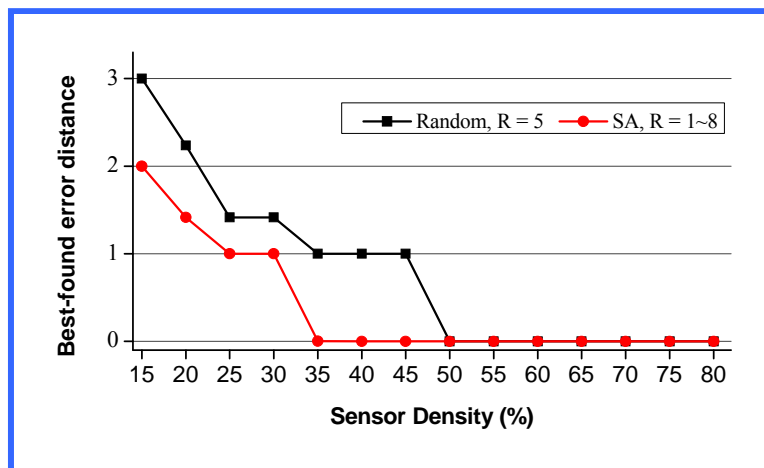


Figure 3.11: Minimum error distance.

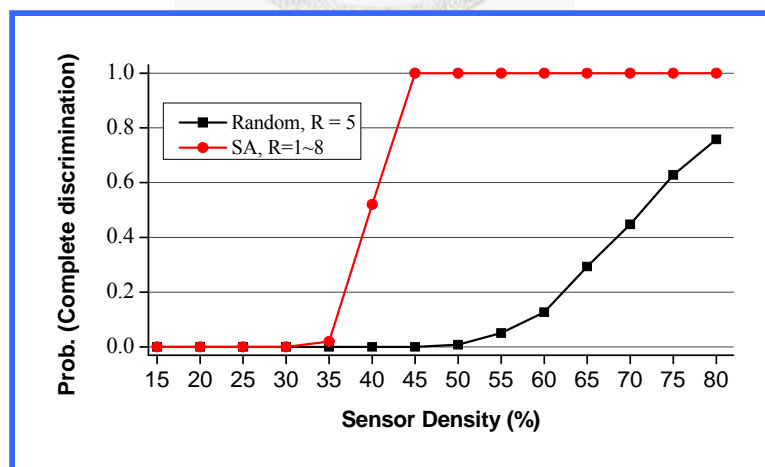


Figure 3.12: Probability to achieve complete discrimination.

Figure 3.12 shows the probability of the complete discrimination of original configurations with uniform radius 5 and various radii by adjusting. The latter can obtain probability of complete discrimination approaching 100% with 45% sensor density. But probability of complete discrimination for initial random configurations with uniform radius doesn't over 80% until given 80% sensor density.

Synthesizing the previous results, we can claim the proposed adjusting radius algorithm is very effective for reducing the maximum error distance and improving the probability of complete discrimination for the original sensor networks, which are deployed by random approach.



3.5 Concluding Remarks

This chapter considers two issues: 1, sensor placement problem for locating target under cost constraints, 2, by adjusting individual radius, target positioning ability of sensor networks is improved.

We first formulate problem one as a min-max mathematical optimization model where the positioning accuracy is the objective. Then, the simulated annealing-based algorithm, Algorithm 3.1, is developed to solve the optimization problem. The experimental results show that the proposed algorithm can efficiently obtain a high-quality solution. Additionally, the proposed algorithm is very effective, scalable, and robust.

Next, we consider given sensor network by random placement with uniform radius initially. By statistical approach, we observe that neither smaller nor larger candidate radius set is an ideal choice for getting completely covered and discriminated sensor networks. The experimental results for the random deployment are analog to our previous work, which focuses on controlled deployment to determine the position of each sensor.

Subsequently, we develop a simulated-annealing-based algorithm, Algorithm 3.2, to determine radius for each sensor. The objective is to improve target positioning ability of the sensor networks. The experiment results show that the proposed algorithm is very effective for reducing the maximum error distance and improving the probability of complete discrimination.





CHAPTER 4 SENSOR PLACEMENT PROBLEM FOR DIFFERENTIATED QUALITY OF POSITIONING AND SURVEILLANCE SERVICES

4.1 Overview

In this chapter, surveillance and target positioning ability are adopted as the QoS parameters for the sensor placement problem. Different levels of QoS require different deployment schemes and amount of resources. For example, in a surveillance-only service, a target anywhere in the sensor field has to be watched by at least one sensor node. However, WSN can deploy more sensor nodes for a target positioning service than for a surveillance-only service. Previous studies focus on how to deploy a WSN that can provide the same level

of QoS for the entire sensor field [CL04] [LC05]. The WSN should ideally provide a uniform QoS service throughout the sensor field. However, in practice, a differentiated quality of positioning service (i.e., differentiated QoS) for different regions in a sensor field is likely to be needed. Some important regions, called *regions of interest* (ROI), require a high accuracy positioning service. Conversely, some regions only need surveillance service. Providing a uniform QoS to comply the QoS requirement for all regions of the field leads to a high resource consumption.

This study presents a generic framework for the sensor placement problem to support differentiated QoS for WSNs. We assume that the terrain of a sensor field is predetermined; hence, sensors with an adjustable sensing radius are deployed at candidate locations to meet a certain QoS requirement. The field has some placement limitations — sensors may be difficult or impossible to place at some locations (e.g., lakes or wetlands) in the field. Furthermore, the field can be divided into several regions. Each of which can have different levels of QoS requirements and service priority. All the QoS requirements can be satisfied if sufficient resources are available. If resources are scarce, then the QoS requirements of regions with higher service priority are satisfied first.

The main parameter of QoS for the target positioning service is the positioning accuracy. Our previous studies define a sensor field as completely discriminable, which is the best positioning accuracy for a sensor field, if each service point in the field has a unique power code. In contrast, if multiple service points with the same power code, it incurs a positioning error. The positioning accuracy is defined in terms of the error distance, i.e., the distance between two service points with the same power code. Thus, a farther error distance means a worse positioning accuracy in a WSN.

Wang, Liu, and Yin refer to OSI 7-Layer to define the QoS parameters for each function layer [WLY06]. Due to their recommendation, in this dissertation we take account of positioning accuracy, which is one kind of data quality. Hence, we consider positioning accuracy belong to application layer QoS parameter.

Yan, He and Stankovic proposed the concept of differentiated surveillance service of sensor networks [YHS03]. They developed an adaptable energy-efficient sensing coverage protocol to support differentiated surveillance by sensor networks.

For providing the target positioning, this work refers to the positioning accuracy as the QoS parameter hereinafter. Moreover, the concept of differentiated QoS is also considered to provide weighted discrimination requirement for each ROI in a sensor field.

This study formulates the proposed problem as a nonlinear integer programming problem, where the objective function is the minimization of the maximum error distance subject to complete coverage and deployment budget constraints. The problem is NP-complete for arbitrary sensor fields. Three heuristics are proposed to determine the location and detection radius for each sensor, such that the positioning accuracy is maximized.

This study differs from prior works in the following ways. First, this study presents a generic framework and a corresponding mathematical model for the sensor placement problem. Second, the positioning accuracy is defined as a QoS parameter in WSN. Third, the sensors detection radius is considered as decision variables in the sensor deployment problem. To the best of our knowledge, no other studies have discussed QoS in WSN about positioning accuracy or differentiating positioning accuracy services up to date.

4.2 Problem Description

4.2.1 The Framework

This study assumes that the terrain of sensor field is predetermined, and that the sensor deployment problem is addressed by the controlled approach. In other words, sensors are placed by a prior planning to satisfy a particular QoS requirement. As shown in Figure 4.1, the sensor field can be represented as a collection of two-dimensional grid points, which are the candidate locations for sensors as well as the service points for the positioning service.

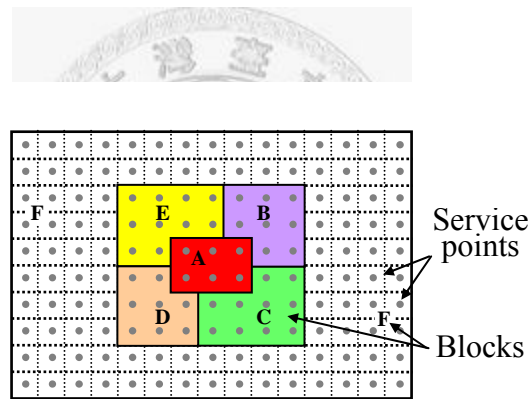


Figure 4.1: A map of museum. (A sensor field with 150 service points)

This study applies the 0/1 detection model for sensors. In this model, the coverage indicator bit of the sensor for a service point is set to 1 if its sensing radius can cover a service point, and 0 otherwise. Then, a power code, which is constructed by all coverage indicator bits of sensors, can be used to represent each service point. A service point with a unique power code is exactly positioning. Otherwise, the error distance of positioning is the maximum distance between those service points with the same power code.

For generality, a terrain of sensor field could have some placement limitations. That is, for all of the positions in the sensor field, the suitability for each placing sensor is unlikely. In most cases, sensors are expensive to place at many locations, and impossible in others, e.g. lakes or wetlands. Additionally, some locations might require surveillance and positioning services, but not be suitable for placing sensors.

Intuitively, we can adopt sensors with large sensing radii which can be used to overcome the placement limitation. However, the use of such sensors affects the positioning accuracy for positioning service. Therefore, this study adopts sensors with adjustable sensing radius and takes the radius of a sensor as one of decision variables for deploying a WSN. This study explores the relationship between the best discrimination resolution of a sensor field and the sensing radius of sensors.

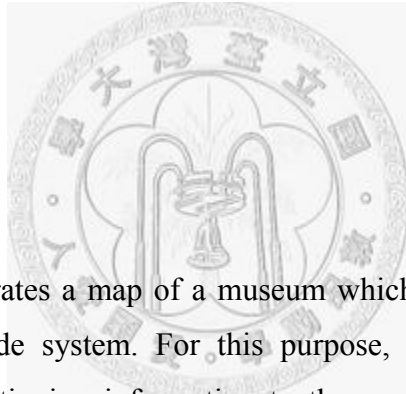
This study provides a differentiated QoS instead of a uniform QoS. The sensor field can be considered as a set of disjoint ROIs that each of them has a different type of QoS. A ROI is an irregular region, which comprises a set of adjacent service points. Three types of QoS are provided for the sensor placement problem as follows:

- **Completely discriminable:** each service point in an ROI can be positioned individually. This is the best QoS provided by a WSN.
- **Discriminable:** a service point can be positioned in an ROI but with a constant error distance. In this type of QoS, a lower error distance indicates a better QoS.
- **Surveillance-only:** all service points can be sensed by sensors, but cannot be discriminated. In this chapter, it is the basic QoS type.

When the resource is limited, the most important ROIs need to have the highest priorities to achieve their QoS requirements. Therefore, except for the

QoS type, each ROI can specify its service priority in the sensor placement phase. The QoS requirement of ROIs with the highest-level priority can be satisfied first if resources are limited. If the resources are still not exhausted, then the requirement of the ROIs with the second-level priority can be satisfied, and so on. The QoS requirement of ROIs with lower-level priority is degraded if the WSN lacks resources. However, all ROIs have to support the surveillance-only service.

The proposed sensor placement framework can be stated briefly as follow. In a sensor field with placement limitations, a WSN is constructed to support the differentiated service for all ROIs in the field. The WSN is constructed by deploying finite sensors at candidate locations, and adjusting the sensing radius of each sensor. The goal of the framework is either to satisfy QoS requirement for all ROIs, or to minimize QoS degradation for each ROI based on its level of priority.



4.2.2 An Example

Figure 4.1 illustrates a map of a museum which needs a WSN to support the security and guide system. For this purpose, the WSN has to provide surveillance and positioning information to the security and guide system in order to provide the location-based service.

The monitoring area of the museum includes the main building and the grass surrounding the main building. Five exhibition areas, denoted by area A, B, C, D, and E, are arranged inside the museum. Area F is the grass field. When a visitor enters the monitoring area, the WSN has to obtain the information rapidly, i.e. the surveillance function. Moreover, while visitors reach exhibit areas, the WSN responds to locate the position of the visitors, i.e. the positioning function.

A WSN that can provide the positioning service for the whole monitoring area of the museum is constructed based on the scenario. Moreover, three priority classes, high, medium, and low, are assigned to each monitoring area to

denote its importance. First, area A is the most important area, so it has a high service priority. Next, the other exhibition areas, B, C, D, and E, have medium service priorities. Third, area F has a low service priority.

According to the above settings, the QoS is separated into four levels, 1–4, as shown in Table 4.1. The best positioning quality, level 4 QoS, is satisfied if the given resource is adequate. Conversely, if the given resource is scarce, the QoS guaranteed for the ROI with lower service priority will be degraded to lower level of QoS, Level 1-3 QoS, according to its service priority. In this scenario, Level 4 QoS is also called Uniform QoS because that all of the service points are discriminated, i.e. they have the same QoS.

Table 4.1: The QoS supported for ROIs in the museum example.

| Level of QoS | QoS supported for ROIs |
|--------------|---|
| 1 | <ul style="list-style-type: none"> ● Completely discriminable: None. ● Discriminable: ROI A. ● Surveillance-only: ROIs B, C, D, E, and F. |
| 2 | <ul style="list-style-type: none"> ● Completely discriminable: ROI A. ● Discriminable: ROIs B, C, D, and E. ● Surveillance-only: ROI F. |
| 3 | <ul style="list-style-type: none"> ● Completely discriminable: ROIs A, B, C, D, and E. ● Discriminable: ROI F. |
| 4 | <ul style="list-style-type: none"> ● Completely discriminable: ROIs A, B, C, D, E, and F. |

4.3 Mathematical Model

This section presents a mathematical model for the proposed sensor placement problem. Since the proposed problem supports the differentiated services as well as the prioritized service, the mathematical model of the problem becomes quite intractable. This study introduces a *discrimination weight* to simplify the model. A discrimination weight is a positive real number, which denotes the priority of discrimination for service points i and j . An ROI with a larger discrimination weight has a higher priority to obtain guaranteed QoS. In this context, the objective of the proposed problem is to minimize the *weighted error distance* for all pairs of service points.

A. Given Parameters and Decision Parameters

The notations used to model the problem are listed as follows.

Given Parameters:

- A : Index set of the service points in the sensor field.
- B : Index set of the sensor's candidate locations, $B \subseteq A$.
- C : Set of the kinds of cost for sensor
- W : Set of the discrimination weight
- R : Set of candidate detection radiuses for sensor
- d_{ij} : Euclidean distance between location i and j ; $i, j \in A$.
- c_k : The cost of sensor located at position k ; $k \in B$, $c_k \in C$.
- c_{min} : The minimum cost of sensors.
- G : The budget limitation for sensors.
- N : The maximum number of sensors, $N = G/c_{min}$.
- w_{ij} : Discrimination weight, $i, j \in A$, $w_{ij} \in W$.
- \bar{K} : A larger number

Decision Variables:

- y_k : 1, if a sensor is allocated at position k , and 0 otherwise, $k \in B$.
- $v_i = (v_{i1}, v_{i2}, \dots, v_{ik})$: A power vector of location i , where v_{ik} is 1 if the target at location i can be detected by the sensor at position k and 0 otherwise, $i \in A, k \in B$.
- r_k : Detection radius of sensor located at k , $k \in B$.

B. Original Model

The original problem (IP4.1) is presented as follows.

$$Z_{IP4.1} = \min_v \max_{\substack{\forall i, j \in A \\ i \neq j}} \frac{w_{ij} d_{ij}}{1 + \bar{K} \sum_{\forall k \in B} (v_{ik} - v_{jk})^2} \quad (IP4.1)$$

subject to:

$$v_{ik} d_{ik} \leq y_k r_k \quad \forall i \in A, k \in B, i \neq k \quad (4.1)$$

$$\frac{d_{ik}}{r_k} > y_k - v_{ik} \quad \forall i \in A, k \in B, i \neq k \quad (4.2)$$

$$v_{kk} = y_k \quad \forall k \in A \cap B \quad (4.3)$$

$$\sum_{\forall k \in B} c_k y_k \leq G \quad (4.4)$$

$$\sum_{\forall k \in B} v_{ik} \geq 1 \quad \forall i \in A \quad (4.5)$$

$$\sum_{\forall k \in B} v_{ik} \leq N \quad \forall i \in A \quad (4.6)$$

$$r_k \in R \quad \forall k \in B \quad (4.7)$$

$$v_{ik} = 0 \text{ or } 1 \quad \forall i \in A, k \in B \quad (4.8)$$

$$y_k = 0 \text{ or } 1 \quad \forall i \in A, k \in B. \quad (4.9)$$

The objective of Problem IP4.1 is to minimize the maximum weighted error distance for any pair of service points. Suppose that $b = \sum_{\forall k \in B} (v_{ik} - v_{jk})^2$ presents the Hamming distance of two power vectors belonging to two service points i and j respectively. If the power vectors are distinct, then the weighted

error distance between service points i and j , i.e. $(w_{ij}d_{ij}/(1+\bar{K}b))$, approaches zero. In contrast, if the power vectors are the same, then the weighted error distance between service points i and j is $w_{ij}d_{ij}$, which is greater than or equal to w_{ij} . Constraint (4.1) requires the power vector (v_{ik}) of a service point which locates on the outside of the sensor coverage to be zero. Constraint (4.2) requires that the power vector (v_{ik}) of service points located on the interior of sensor detection range is 1. Constraint (4.3) requires the coverage to be full for the service point on which sensor is located. Constraint (4.4) requires that the budget to be limited. Constraint (4.5) is the completed coverage requirement. Constraint (4.6) requires the amount of sensors to monitor service point i . Constraint (4.7) requires that the detection radius of sensors belong to set R . Constraints (4.8) and (4.9) are integer constraints.

Subsequently, we discuss how to determine the values of weights and constant \bar{K} , two propositions are obtained and presented as follows.

Proposition 4.1: If the diameter of the sensor field is D , and the discrimination weights are w_1, w_2, \dots, w_h , and $w_1 < w_2 < \dots < w_h$. Then $w_{i+1} > Dw_i$ for any two adjacent weights w_i and w_{i+1} .

Proof:

Some groups of service points all have the same power code. Among these groups, the pair of service points with the highest discrimination weight and the furthest distance has the maximum weighted error distance (the worst positioning accuracy). In this sensor network, the weighted error distance S of any pair of service points, which has discrimination weight w_{i+1} and one unit length apart, should be smaller than Dw_i .

Proposition 4.2: If the diameter of sensor field is D ; the detection range is r , and the discrimination weights are w_1, w_2, \dots, w_n , and $w_1 < w_2 < \dots < w_n$, then the constant \bar{K} must satisfy

constraints as follows.

1. If $2r \geq D$, then $w_l > (w_h \cdot D)/(1 + \bar{K})$,
2. If $2r < D$, then $w_l > \max \{w_h(2r)/(1 + \bar{K}), (w_h \cdot D)/(1 + 2\bar{K})\}$.

Proof:

The minimum value of weighted error distance for indistinguishable service points is w_l , which must be greater than the weighted error distance for each pair of discriminated service points, $(w_i \cdot d_{ij})/(1 + \bar{K} \Sigma(v_i - v_j)^2)$. The maximum value of $(w_i \cdot d_{ij})/(1 + \bar{K} \Sigma(v_i - v_j)^2)$ is first discussed.

For $2r \geq D$

The furthest distance between any pair of discriminated service points is D . The closest distance between two discriminated power codes is 1. Hence, the maximum value of $(w_i \cdot d_{ij})/(1 + \bar{K} \Sigma(v_i - v_j)^2)$ is $(w_i \cdot D)/(1 + \bar{K})$. Therefore, the constraint, $w_l > (w_h \cdot D)/(1 + \bar{K})$, must be satisfied for \bar{K} .

For $2r < D$

The two discriminated service points with furthest distance D cannot possibly be covered with the same sensors while the detection radius of the sensor is less than the diameter of the sensor field. Hence, the distance between their power codes is at least 2, and $(w_h \cdot d_{ij})/(1 + \bar{K} \Sigma(v_i - v_j)^2)$ is $(w_h \cdot D)/(1 + 2\bar{K})$.

Conversely, consider the case of two discriminated service points covered by at least one sensor. The further distance between them is $2r$, and the minimum distance between their power codes is 1. In this case, the value of $(w_h \cdot d_{ij})/(1 + \bar{K} \Sigma(v_i - v_j)^2)$ is $(w_h \cdot 2r)/(1 + \bar{K})$.

Therefore, both $(w_h \cdot D)/(1 + 2\bar{K})$ and $(w_h \cdot 2r)/(1 + \bar{K})$ must be less than w_l . \square

4.4 Lagrangean Relaxation Approach

4.4.1 Equivalent Model

The original model is a nonlinear combinatorial problem, and is therefore hard to solve directly. Hence, an equivalent model is developed. The transformation method is also presented in this section.

An equivalent formulation of problem (IP4.1) is given by (IP4.2) below.

Define

$$S_v = \max_{\substack{\forall i, j \in A \\ i \neq j}} \frac{w_{ij} d_{ij}}{1 + \bar{K} \sum_{\forall k \in B} (v_{ik} - v_{jk})^2} \dots\dots\dots(4.10)$$

, and rewrite the objective function as follows.

$$Z_{IP4.2} = \min_v S_v \tag{IP4.2}$$

subject to:

$$\frac{1}{S_v} \leq \frac{1 + \bar{K} \sum_{\forall k \in B} (v_{ik} - v_{jk})^2}{w_{ij} d_{ij}} \quad \forall i, j \in A, i \neq j \tag{4.11}$$

$$\underline{S} \leq S_v \leq \bar{S} \tag{4.12}$$

\underline{S} is lower bound of S_v ;
 \bar{S} is upper bound of S_v .

$$v_{ik} d_{ik} \leq y_k r_k \quad \forall i \in A, k \in B, i \neq k \tag{4.1}$$

$$\frac{d_{ik}}{r_k} > y_k - v_{ik} \quad \forall i \in A, k \in B, i \neq k \tag{4.2}$$

$$v_{kk} = y_k \quad \forall k \in A \cap B \tag{4.3}$$

$$\sum_{\forall k \in B} c_k y_k \leq G \tag{4.4}$$

$$\sum_{\forall k \in B} v_{ik} \geq 1 \quad \forall i \in A \tag{4.5}$$

$$\sum_{\forall k \in B} v_{ik} \leq N \quad \forall i \in A \quad (4.6)$$

$$r_k \in R \quad \forall k \in B \quad (4.7)$$

$$v_{ik} = 0 \text{ or } 1 \quad \forall i \in A, k \in B \quad (4.8)$$

$$y_k = 0 \text{ or } 1 \quad \forall i \in A, k \in B \quad (4.9)$$

Constraint (4.11) is added to stipulate the upper and lower bound of the maximum weighted error distance S . The theoretic upper and lower bounds are then discussed.

Proposition 4.3: Theoretic upper bound of S_v, \bar{S} , is at $w_h D$, where w_h denotes the highest discrimination weight, and D is the smaller value of the diameter of a sensor field and the maximum detection range of sensors.

Proof:

If the sensor field is not completely discriminable, then a pair of service points on the field that has the farthest distance D and the same power vectors can be found. The value of D is not greater than the diameter of the field and the maximum detection range of the sensors. Hence, \bar{S} is bounded by $w_h D$. \square

Proposition 4.4: Theoretic lower bound of S_v, \underline{S} , is at $w_l / (1 + D_h \bar{K})$ where w_l is the lowest discrimination weight, and D_h represents the maximum Hamming distance of power vectors for all service points. If n is the maximum number of sensors that can cover a service point, then $D_h = \max\{2n, N\}$.

4.4.2 Transformation

Problem (IP4.2) is still very hard to solve, since Constraint (4.11) is nonlinear. Instead of solving Problem (IP4.2) directly, the cutting plane method is applied to transform Constraint (4.11) to a linear Constraint (4.13).

An auxiliary variable t_{ijk} is introduced, where $t_{ijk}=v_{ik}v_{jk}$. Table 4.2 shows the truth table for variables v_{ik} , v_{jk} and t_{ij} . The possible values for the three variables only exist in four integer vertices, $p1$, $p2$, $p3$, and $p4$, of the polyhedron, which are depicted in Figure 4.2. The four planes constructing the polyhedron are presented as following.

$$\begin{aligned}
 v_{ik}-t_{ijk} &\geq 0 && \forall i,j \in A, i \neq j, k \in B \\
 v_{jk}-t_{ijk} &\geq 0 && \forall i,j \in A, i \neq j, k \in B \\
 v_{ik}+v_{jk}-t_{ijk} &\leq 1 && \forall i,j \in A, i \neq j, k \in B \\
 t_{ijk} &\geq 0 && \forall i,j \in A, i \neq j, k \in B
 \end{aligned}$$

Table 4.2: Truth table for variables v_{ik} , v_{jk} , and t_{ijk} .

| v_{ik} | v_{jk} | t_{ijk} |
|----------|----------|-----------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

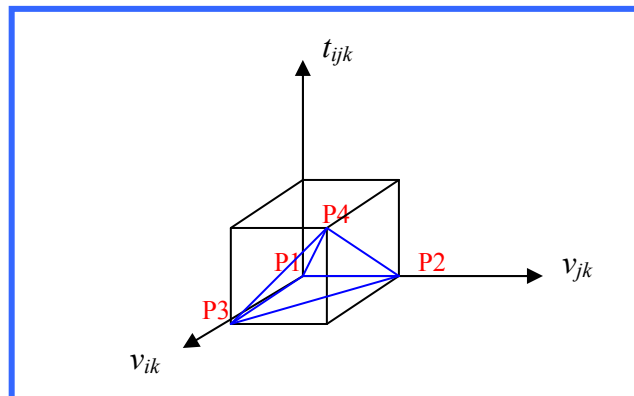


Figure 4.2: Relationship between v_{ik} , v_{jk} , and t_{ijk} .

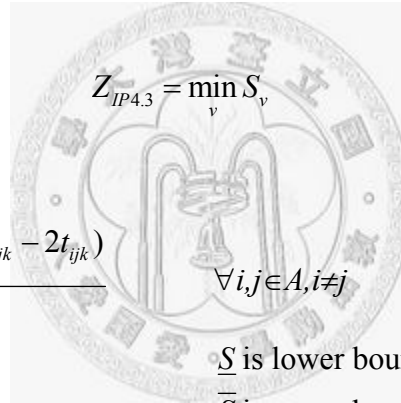
The auxiliary variable t_{ijk} is employed to replace $v_{ik}v_{jk}$. Equation (4.11) can thus be transformed to linear Equation (4.22). The transformation is described as follows.

$$\begin{aligned}(v_{ik} - v_{jk})^2 &= v_{ik}^2 - 2v_{ik}v_{jk} + v_{jk}^2 \\ &= v_{ik} - 2v_{ik}v_{jk} + v_{jk} \quad (\because v_{ik}, v_{jk} = 0 \text{ or } 1)\end{aligned}$$

$$\text{Let } t_{ijk} = v_{ik}v_{jk}$$

$$\Rightarrow (v_{ik} - v_{jk})^2 = v_{ik} + v_{jk} - 2t_{ijk}$$

According to the cutting plane method, Constraints (4.13)~(4.17) must be added to require relationship between v_{ik} , v_{jk} , and t_{ijk} . Constraint (4.13) requires that the number of sensors that can cover both points (i and j) cannot be over the total number of sensors. Hence, the nonlinear combinatorial Problem (IP4.1) is transformed to an equivalent linear combinatorial Problem (IP4.3).



$$Z_{IP4.3} = \min_v S_v \quad (IP4.3)$$

subject to:

$$\frac{1}{S_v} \leq \frac{1 + \bar{K} \sum_{\forall k \in B} (v_{ik} + v_{jk} - 2t_{ijk})}{w_{ij} d_{ij}} \quad \forall i, j \in A, i \neq j \quad (4.11)$$

$$\underline{S} \leq S_v \leq \bar{S} \quad \begin{array}{l} \underline{S} \text{ is lower bound of } S_v; \\ \bar{S} \text{ is upper bound of } S_v. \end{array} \quad (4.12)$$

$$v_{ik} d_{ik} \leq y_k r_k \quad \forall i \in A, k \in B, i \neq k \quad (4.1)$$

$$\frac{d_{ik}}{r_k} > y_k - v_{ik} \quad \forall i \in A, k \in B, i \neq k \quad (4.2)$$

$$v_{kk} = y_k \quad \forall k \in A \cap B \quad (4.3)$$

$$\sum_{\forall k \in B} c_k y_k \leq G \quad (4.4)$$

$$\sum_{\forall k \in B} v_{ik} \geq 1 \quad \forall i \in A \quad (4.5)$$

$$\sum_{\forall k \in B} v_{ik} \leq N \quad \forall i \in A \quad (4.6)$$

$$r_k \in R \quad \forall k \in B \quad (4.7)$$

$$v_{ik}=0 \text{ or } 1 \quad \forall i \in A, k \in B \quad (4.8)$$

$$y_k=0 \text{ or } 1 \quad \forall i \in A, k \in B \quad (4.9)$$

$$\sum_{\forall k \in B} t_{ijk} \leq N \quad \forall i, j \in A, i \neq j \quad (4.13)$$

$$v_{ik} - t_{ijk} \geq 0 \quad \forall i, j \in A, i \neq j, k \in B \quad (4.14)$$

$$v_{jk} - t_{ijk} \geq 0 \quad \forall i, j \in A, i \neq j, k \in B \quad (4.15)$$

$$v_{ik} + v_{jk} - t_{ijk} \leq 1 \quad \forall i, j \in A, i \neq j, k \in B \quad (4.16)$$

$$t_{ijk}=0 \text{ or } 1 \quad \forall i, j \in A, i \neq j, k \in B \quad (4.17)$$

4.4.3 Relaxation

This section presents the algorithm for solving the proposed sensor placement problem. An algorithm based upon Lagrangean relaxation is considered. Lagrangean relaxation is an approach for obtaining lower bounds (for minimization problems) as well as good solutions in integer programming problems. A Lagrangean relaxation is obtained by identifying in the primal problem and a set of complicated constraints whose removal will simplify the solving procedure of the primal problem. Each of the complicated constraints is multiplied by a multiplier and added to objective function. The mechanism is known as dualizing the complicated constraints [Fis81] [Fis85] [Geo74] [HWC74].

A. Relaxation

By Lagrangean relaxation, we dualize Constraints (4.1), (4.2), (4.3), (4.11), (4.14), (4.15), and (4.16) of Problem (IP4.3), as well as get the following Lagrangean relaxation problem.

Problem (LR4.1):

$$\begin{aligned}
Z_D(u^1, u^2, u^3, u^4, u^5, u^6, u^7) = \min_v & \left\{ S_v + \sum_{\forall i \in A} \sum_{\substack{\forall j \in A \\ i \neq j}} u_{ij}^1 \left(\frac{1}{S_v} - \frac{1 + \bar{K} \sum_{\forall k \in B} (v_{ik} + v_{jk} - 2t_{ijk})}{w_{ij} d_{ij}} \right) \right. \\
& + \sum_{\forall i \in A} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{ik}^2 (v_{ik} d_{ik} - y_k r_k) + \sum_{\forall i \in A} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{ik}^3 \left(y_k - v_{ik} - \frac{d_{ik}}{r_k} \right) + \sum_{\forall k \in B} \sum_{\forall k \in B} u_{kk}^4 (v_{ik} - y_k) \\
& + \sum_{\forall i \in A} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\forall k \in B} u_{ijk}^5 (t_{ijk} - v_{ik}) + \sum_{\forall i \in A} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\forall k \in B} u_{ijk}^6 (t_{ijk} - v_{jk}) \\
& \left. + \sum_{\forall i \in A} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\forall k \in B} u_{ijk}^7 (v_{ik} + v_{jk} - t_{ijk} - 1) \right\} \quad (\text{LR4.1})
\end{aligned}$$

subject to:

$$\underline{S} \leq S_v \leq \bar{S}$$

$$\sum_{\forall k \in B} c_k y_k \leq G$$

$$\sum_{\forall k \in B} v_{ik} \geq 1$$

$$\sum_{\forall k \in B} v_{ik} \leq N$$

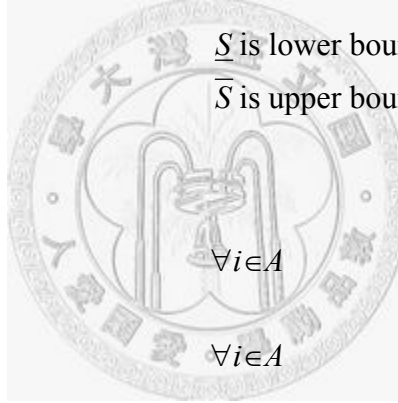
$$r_k \in R$$

$$v_{ik} = 0 \text{ or } 1$$

$$y_k = 0 \text{ or } 1$$

$$\sum_{\forall k \in B} t_{ijk} \leq N$$

$$t_{ijk} = 0 \text{ or } 1$$



\underline{S} is lower bound of S_v ;

\bar{S} is upper bound of S_v .

$$(4.11)$$

$$(4.4)$$

$$(4.5)$$

$$(4.6)$$

$$(4.7)$$

$$(4.8)$$

$$(4.9)$$

$$(4.13)$$

$$(4.17)$$

The multipliers u^1, u^2, \dots, u^7 are the vectors of $\{u_{ij}^1\}, \{u_{ik}^2\}, \dots, \{u_{ijk}^7\}$ respectively. Besides Constraint (4.3) with multiplier $\{u_{kk}^4\}$, the other Constraints are dulized such that the corresponding multipliers, $u^1, u^2, u^3, u^5, u^6,$

and u^7 are nonnegative.

The dual Problem (LR4.1) is rewritten to Equation (LR4.2), where the constant terms are omitted.

$$\begin{aligned}
Z_D(u^1, u^2, u^3, u^4, u^5, u^6, u^7) = \min_v & \left\{ S_v + \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} u_{ij}^1 \frac{1}{S_v} - \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \frac{u_{ij}^1}{w_{ij} d_{ij}} \right. \\
& - \bar{K} \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} u_{ij}^1 \frac{\sum_{k \in B} v_{ik}}{w_{ij} d_{ij}} - \bar{K} \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} u_{ij}^1 \frac{\sum_{k \in B} v_{jk}}{w_{ij} d_{ij}} + 2\bar{K} \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} u_{ij}^1 \frac{\sum_{k \in B} t_{ijk}}{w_{ij} d_{ij}} \\
& + \sum_{\substack{\forall i \in A \\ i \neq k}} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{ik}^2 d_{ik} v_{ik} - \sum_{\substack{\forall i \in A \\ i \neq k}} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{ik}^2 r_k y_k + \sum_{\substack{\forall i \in A \\ i \neq k}} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{ik}^3 y_k - \sum_{\substack{\forall i \in A \\ i \neq k}} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{ik}^3 v_{ik} \\
& - \sum_{\substack{\forall i \in A \\ i \neq k}} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{ik}^3 \frac{d_{ik}}{r_k} + \sum_{\substack{\forall k \in A \\ i \neq k}} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{kk}^4 v_{kk} - \sum_{\substack{\forall k \in B \\ i \neq k}} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{kk}^4 y_k + \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\substack{\forall k \in B \\ i \neq j}} u_{ijk}^5 t_{ijk} \\
& - \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\substack{\forall k \in B \\ i \neq j}} u_{ijk}^5 v_{ik} + \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\substack{\forall k \in B \\ i \neq j}} u_{ijk}^6 t_{ijk} - \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\substack{\forall k \in B \\ i \neq j}} u_{ijk}^6 v_{jk} \\
& + \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\substack{\forall k \in B \\ i \neq j}} u_{ijk}^7 v_{ik} + \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\substack{\forall k \in B \\ i \neq j}} u_{ijk}^7 v_{jk} - \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\substack{\forall k \in B \\ i \neq j}} u_{ijk}^7 t_{ijk} \\
& \left. - \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\substack{\forall k \in B \\ i \neq j}} u_{ijk}^7 \right\} \tag{LR4.2}
\end{aligned}$$

, where the constant term is $\left(- \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \frac{u_{ij}^1}{w_{ij} d_{ij}} - \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\substack{\forall k \in B \\ i \neq j}} u_{ijk}^7 \right)$.

B. Decomposition

According to Lagrangean relaxation approach, Problem (LR4.2) will be

decomposed into four mutually independent and easily solvable subproblems. Each sub-problem only involves one or two decision variables and must be optimal solved. Note that, the constant part is excluded from the objective function in the subproblems but will be considered in the lower bound computation.

Subproblem 1: for S_v

$$Z_{SUB4.1}(u^1) = \min \left(S_v + \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} u_{ij}^1 \frac{1}{S_v} \right) \quad (\text{SUB4.1})$$

subject to:

$$\underline{S} \leq S_v \leq \bar{S} \quad (4.11)$$

To optimal solve the subproblem, the right hand side of Equation (SUB4.1) will be differentiated respected to variable S . Let new equation equals to zero and get the optimal solution of variable S , $S_{opt} = \sqrt{\sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} u_{ij}^1}$.

If $\underline{S} \leq S_{opt} \leq \bar{S}$ then let $Z_{SUB4.1} = 2 \sqrt{\sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} u_{ij}^1}$. Otherwise, \underline{S} and \bar{S} are substituted for S to get $Z_{SUB4.1}^U(u_{ij}^1)$ and $Z_{SUB4.1}^L(u_{ij}^1)$. We can get optimal solution such that $Z_{SUB4.1}(u^1) = \min \left\{ \left(\bar{S} + \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \frac{u_{ij}^1}{\bar{S}} \right), \left(\underline{S} + \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\substack{\forall j \in A \\ i \neq j}} \frac{u_{ij}^1}{\underline{S}} \right) \right\}$.

Subproblem 2: for y_k and r_k

$$Z_{SUB4.2}(u^2, u^3, u^4) = \min \left(- \sum_{\substack{\forall i \in A \\ i \neq k}} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{ik}^2 r_k y_k + \sum_{\substack{\forall i \in A \\ i \neq k}} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{ik}^3 y_k \right. \\ \left. - \sum_{\substack{\forall i \in A \\ i \neq k}} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{ik}^3 \frac{d_{ik}}{r_k} - \sum_{\substack{\forall k \in B \\ i \neq k}} \sum_{\substack{\forall k \in B \\ i \neq k}} u_{kk}^4 y_k \right) \quad (\text{SUB4.2})$$

Equation (SUB4.2) is rewritten to (SUB4.2a), as follows.

$$Z_{SUB4.2}(u^2, u^3, u^4) = \min \sum_{\forall k \in B} \left(\sum_{\substack{\forall i \in A \\ i \neq k}} \left((-u_{ik}^2 r_k + u_{ik}^3) y_k - u_{ik}^3 \frac{d_{ik}}{r_k} \right) - \sum_{\forall k \in B} u_{kk}^4 y_k \right) \quad (\text{SUB4.2a})$$

subject to:

$$\sum_{\forall k \in B} c_k y_k \leq G \quad (4.4)$$

$$r_k \in R \quad \forall k \in B \quad (4.7)$$

$$y_k = 0 \text{ or } 1 \quad \forall i \in A, k \in B. \quad (4.9)$$

Subproblem 2 comprises $|B|$ problems. For each sensor k , we let $b_k(r_k)$ represent the function (SUB 4.2a) while $y_k = 1$.

$$b_k(r_k) = \sum_{\substack{\forall i \in A \\ i \neq k}} \left((-u_{ik}^2 r_k + u_{ik}^3) - u_{ik}^3 \frac{d_{ik}}{r_k} \right) - \sum_{\forall k \in B} u_{kk}^4$$

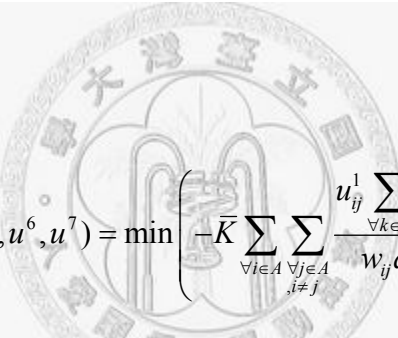
Then, we calculate b_k for each r_k , which is belong to set R , as well as find the best r_k such that b_k is the minimum denoted by $b_{k,min}$. Next, from the set of unallocated sensors, we iteratively choose sensor k with the minimal $b_{k,min}$ to be set. The cost of sensor k must be accumulated. While adding the cost of sensor k will exceed the total deployment cost G , the procedure must be stopped.

Subproblem 3: for v_{ik}

$$Z_{SUB4.3}(u^1, u^2, u^3, u^4, u^5, u^6, u^7) = \min \left(-\bar{K} \sum_{\forall i \in A} \sum_{\substack{\forall j \in A \\ i \neq j}} u_{ij}^1 \frac{\sum_{\forall k \in B} v_{ik}}{w_{ij} d_{ij}} \right)$$

$$\begin{aligned}
& -\bar{K} \sum_{\substack{\forall i \in A \\ , i \neq j}} \sum_{\substack{\forall j \in A \\ , i \neq j}} u_{ij}^1 \frac{\sum_{\forall k \in B} v_{jk}}{w_{ij} d_{ij}} + \sum_{\substack{\forall i \in A \\ , i \neq k}} \sum_{\forall k \in B} u_{ik}^2 d_{ik} v_{ik} - \sum_{\substack{\forall i \in A \\ , i \neq k}} \sum_{\forall k \in B} u_{ik}^3 v_{ik} + \sum_{\forall k \in K} \sum_{\forall k \in B} u_{kk}^4 v_{kk} \\
& - \sum_{\substack{\forall i \in A \\ , i \neq j}} \sum_{\forall j \in A} \sum_{\forall k \in B} u_{ijk}^5 v_{ik} - \sum_{\substack{\forall i \in A \\ , i \neq j}} \sum_{\forall j \in A} \sum_{\forall k \in B} u_{ijk}^6 v_{jk} + \sum_{\substack{\forall i \in A \\ , i \neq j}} \sum_{\forall j \in A} \sum_{\forall k \in B} u_{ijk}^7 v_{ik} \\
& + \sum_{\substack{\forall i \in A \\ , i \neq j}} \sum_{\forall j \in A} \sum_{\forall k \in B} u_{ijk}^7 v_{jk} \Big) \tag{SUB4.3}
\end{aligned}$$

To simplify Equation (SUB4.3), variable v_{jk} should be eliminated from Equation (SUB4.3). For each term with the variable v_{jk} , index j substitutes for i contrariwise. Consequently, the equivalent subproblem (SUB4.3a) replaces Equation (SUB4.3).



$$\begin{aligned}
Z_{SUB4.3}(u^1, u^2, u^3, u^4, u^5, u^6, u^7) &= \min \left(-\bar{K} \sum_{\substack{\forall i \in A \\ , i \neq j}} \sum_{\substack{\forall j \in A \\ , i \neq j}} \frac{u_{ij}^1 \sum_{\forall k \in B} v_{ik}}{w_{ij} d_{ij}} - \bar{K} \sum_{\substack{\forall i \in A \\ , i \neq j}} \sum_{\substack{\forall j \in A \\ , i \neq j}} \frac{u_{ji}^1 \sum_{\forall k \in B} v_{ik}}{w_{ji} d_{ji}} \right. \\
& + \sum_{\substack{\forall i \in A \\ , i \neq k}} \sum_{\forall k \in B} u_{ik}^2 d_{ik} v_{ik} - \sum_{\substack{\forall i \in A \\ , i \neq k}} \sum_{\forall k \in B} u_{ik}^3 v_{ik} + \sum_{\forall i \in B} \sum_{\forall k \in B} u_{ik}^4 v_{ik} - \sum_{\substack{\forall i \in A \\ , i \neq j}} \sum_{\forall j \in A} \sum_{\forall k \in B} u_{ijk}^5 v_{ik} \\
& \left. - \sum_{\substack{\forall i \in A \\ , i \neq j}} \sum_{\forall j \in A} \sum_{\forall k \in B} u_{jik}^6 v_{ik} + \sum_{\substack{\forall i \in A \\ , i \neq j}} \sum_{\forall j \in A} \sum_{\forall k \in B} u_{ijk}^7 v_{ik} + \sum_{\substack{\forall i \in A \\ , i \neq j}} \sum_{\forall j \in A} \sum_{\forall k \in B} u_{jik}^7 v_{ik} \right) \\
&= \min \sum_{\forall i \in A} \left\{ \sum_{\substack{\forall k \in B \\ , i \neq j}} \sum_{\substack{\forall j \in A \\ , i \neq j}} \left(\frac{-\bar{K} u_{ij}^1}{w_{ij} d_{ij}} - \frac{\bar{K} u_{ji}^1}{w_{ji} d_{ji}} - u_{ijk}^5 - u_{jik}^6 + u_{ijk}^7 + u_{jik}^7 \right) \right. \\
& \left. + \sum_{\substack{\forall k \in B \\ , i \neq k}} (u_{ik}^2 d_{ik} - u_{ik}^3) + \sum_{\substack{\forall k \in B \\ , i=k}} u_{ik}^4 \right\} v_{ik} \tag{SUB4.3a}
\end{aligned}$$

subject to:

$$\sum_{\forall k \in B} v_{ik} \geq 1 \quad \forall i \in A \quad (4.5)$$

$$\sum_{\forall k \in B} v_{ik} \leq N \quad \forall i \in A \quad (4.6)$$

$$v_{ik} = 0 \text{ or } 1 \quad \forall i \in A, k \in B \quad (4.8)$$

Subproblem 3 comprises $|A \times B|$ problems. For each service point i , we calculate the coefficient of each variable v_{ik} , and sort them in non-decreasing order. Iteratively, if the minimal one of the coefficient v_{ik} is a positive number, we set the corresponding v_{ik} to be zero. Otherwise, the corresponding v_{ik} is assigned to 1 under the number of sensors constraint. Additionally, for each service point, the coverage constraint must be satisfied also. If no any v_{ik} is 1 for service point i , the v_{ik} with the minimum coefficient will be set.

Subproblem 4: for t_{ijk}

$$Z_{SUB4.4}(u^1, u^5, u^6, u^7) = \min \left(2\bar{K} \sum_{\forall i \in A} \sum_{\substack{\forall j \in A \\ i \neq j}} u_{ij}^1 \frac{\sum_{\forall k \in B} t_{ijk}}{w_{ij} d_{ij}} + \sum_{\forall i \in A} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\forall k \in B} u_{ijk}^5 t_{ijk} \right. \\ \left. + \sum_{\forall j \in A} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\forall k \in B} u_{ijk}^6 t_{ijk} - \sum_{\forall i \in A} \sum_{\substack{\forall j \in A \\ i \neq j}} \sum_{\forall k \in B} u_{ijk}^7 t_{ijk} \right) \quad (SUB4.4)$$

Subproblem (SUB4.4) comprises $|A \times A \times B|$ problems. It is rewritten to Subproblem (SUB4.4a) as follows.

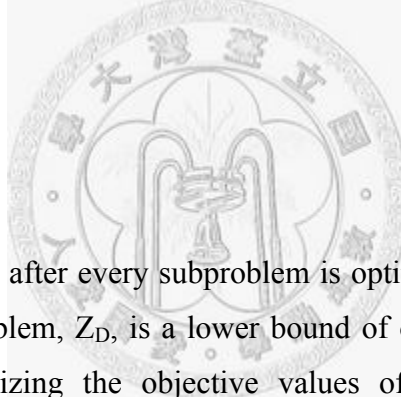
$$Z_{SUB4.4}(u^1, u^5, u^6, u^7) = \min \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\forall j \in A} \sum_{\forall k \in B} \left(\frac{2\bar{K}u_{ij}^1}{w_{ij}d_{ij}} + u_{ijk}^5 + u_{ijk}^6 - u_{ijk}^7 \right) t_{ijk} \quad (\text{SUB4.4a})$$

subject to:

$$\sum_{\forall k \in B} t_{ijk} \leq N \quad \forall i, j \in A, i \neq j \quad (4.13)$$

$$t_{ijk} = 0 \text{ or } 1 \quad \forall i, j \in A, i \neq j, k \in B \quad (4.17)$$

Subproblem (SUB4.4a) can be solved easily. First, the coefficient is calculated for each t_{ijk} . Then we sort the coefficients in non-decreasing order. If the coefficient for t_{ijk} is non-positive, t_{ijk} is assigned to 1. Otherwise, t_{ijk} is zero. However, the number of t_{ijk} which is set one can't exceed the maximal number of sensors.



C. Lower bound

In each iteration, after every subproblem is optimally solved, the objective value of the dual problem, Z_D , is a lower bound of original problem. It can be obtained by summarizing the objective values of all subproblem and the constant part as the following equation.

$$Z_D(u^1, u^2, u^3, u^4, u^5, u^6, u^7) = Z_{SUB4.1} + Z_{SUB4.2} + Z_{SUB4.3} + Z_{SUB4.4} + \left(- \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\forall j \in A} \frac{u_{ij}^1}{w_{ij}d_{ij}} - \sum_{\substack{\forall i \in A \\ i \neq j}} \sum_{\forall j \in A} \sum_{\forall k \in B} u_{ijk}^7 \right)$$

According to the weak Lagrangean duality theorem [Fis81] [Fis85], the optimal objective value of the dual problem (LR), $Z_D(u^1, u^2, u^3, u^4, u^5, u^6, u^7)$, is a lower bound on primal problem (IP4.3). $Z_{IP4.3}$ is subject to $(u^1, u^2, u^3, u^5, u^6, u^7) \geq 0$. Therefore, we can obtain the lower bound by

$$Z_D = \max_{(u^1, u^2, u^3, u^4, u^5, u^6, u^7) \geq 0} Z_D(u^1, u^2, u^3, u^4, u^5, u^6, u^7) \quad (D1)$$

Several methods can be used for solving Equation (D1) to get the highest lower bound. One of the most popular methods is the subgradient method. Let a $(|A|^2 + |A||B| + |A||B| + |A||B| + |A|^2|B| + |A|^2|B| + |A|^2|B|)$ vector g represent a subgradient of $Z_D(u^1, u^2, u^3, u^4, u^5, u^6, u^7)$. We denote $\pi = (u_{ij}^1, u_{ik}^2, u_{ik}^3, u_{ik}^4, u_{ijk}^5, u_{ijk}^6, u_{ijk}^7)$ as the vector of Lagrangean multipliers with respect to relaxed constraints. In iteration m of subgradient optimization procedure, the multiplier π^m is updated by $\pi^{m+1} = \pi^m + \xi^m g^m$.

$$\text{The step size } \xi^m \text{ is determined by } \xi^m = \frac{\lambda(Z_{IP4.3}^* - Z_D(\pi^m))}{\|g^m\|^2}$$

, where $Z_{IP4.3}^*$ represents an upper bound on the primal objective value, obtained by applying a heuristic to (IP4.3), and λ is a scalar satisfying $0 \leq \lambda \leq 2$.

4.4.4 Getting Primal Feasible Solutions

After optimally solving each Lagrangean relaxation problem, a set of decision variables can be found. Since some constraints are relaxed, the solutions of Lagrangean relaxation might be infeasible for the primal problem. Hence, an efficient heuristic algorithm which adjusts the dual solutions to obtain the feasible solutions for the primal problem (IP4.3) must be developed. By increasing the number of iterations, the better primal feasible solution is an upper bound (UB) on the primal problem (IP4.3), while the dual problem provides the lower bound (LB) of the primal problem (IP4.3).

Algorithm 4.1

Step 1: Initialize the decision variables, y_k , v_{ik} , and r_k .

Step 1.1: Before the fifth iteration, initial decision variables y_k are

determined by sub-problem (SUB4.2) on Lagrangean relaxation problem. For each sensor, the five recent history solutions are recorded. After the fifth iteration, we can randomly determine whether decision variables y_k should be one by the placement probability for the sensor k in history record.

Step 1.2: Check Constraints (4.1), (4.2), and (4.3), for each sensor k . For each service point i , let $v_{ik}=0$ if $d_{ik}>$ maximal radius. Add sensor k if $v_{ik}=1$ and $d_{ik}<$ maximal radius, for each service point i .

Step 1.3: Determine the radius r_k if sensor k is allocated. For each sensor k , find the farthest distance between sensor k and the service points with $v_{ik}=1$ to determine the radius r_k .

Step 1.4: The decision variables v_{ik} can be obtained by the decision variables y_k .

Step 2: To satisfy the coverage and cost constraint, the sensors might be added, deleted or changed the radius.

Step 2.1: If the coverage constraint is violated, “Change Radius” or “Add Sensor” procedure will be executed. Randomly select a sensor if the increase of radius for the sensor can improve coverage, the radius will be changed. If the operation of change radius is not suitable, the other operation, “Add Sensor”, can be tried. The sensor that can cover the most uncovered service points will be added with a proper radius until the coverage constraint is satisfied.

Step 2.2: If the budget constraint is violated, “Delete Sensor” procedure can be applied. Remove a sensor away from the sensor field randomly, if the coverage of the sensor field is not changed. The operation is executed until the budget constraint is satisfied.

Step 2.3: Running the previous two steps until both the coverage and budget constraint are satisfied.

Step 3: For each sensor, the operations “Change Position” and “Modify Radius” are tried in order to improve the discrimination resolution.

4.4.5 Computational Results

Three sets of experiments were conducted to evaluate the performance of the proposed algorithm under various settings for the numbers of priority class, amount of resources, placement limitation, topology area, and detection radius. The proposed algorithms were coded in C in the Microsoft Visual C++ 6.0 development environment. All the experiments were performed on a Pentium IV-3.0GHz PC running Microsoft Windows XP Pro. The performance metrics were assessed in terms of the solution quality and computation time.

A. Experiment I

The first experiment was designed to observe the solution qualities of the proposed algorithm. In this experiment, the system parameters, namely placement limitation and topology area were fixed while the amount of resources, detection radius, and numbers of priority class were variant parameters. The sensor field topology was based on the previous museum example, as shown in Figure 4.1. The sensor field was a 10×15 rectangular field, and was divided into seven different ROIs. The QoS and service priority for these ROIs are the same as the example. The candidate sensing-radius of sensor is either fixed or varying between 1 and 8 units of length. The parameters about LR algorithm include: $0 < \lambda \leq 2$, improvement counter is 40, and number of iterations is 1500.

For the differential positioning accuracy, the set of discrimination weights was set to $\{0.1, 5, 100\}$. The discrimination weight between any two service points on ROI A was set to the high weight, $w_h=100$. The discrimination weight

between any two service points, which both on ROIs B, C, D, or E, was set to the medium weight, $w_m=5$. The weights between any two service points, which both are on ROI F was set to the low weight $w_l=0.1$. The weight between any pair of service points on different ROIs was set to the highest weight, $w_h=100$. The diameter of the sensor field was set to $D=16$. The sensor cost of each location, c_k , is one. The parameters w_l , w_m , and w_h were adopted according to Proposition 4.1. Parameter $\bar{K} = 20000$ follows Proposition 4.2.

To contrast the differentiated QoS and uniform QoS, we design another set of experiments for uniform QoS. All of service points in the sensor field, shown in Figure 4.1, require the same discrimination weight. Let $w_{ij}=100$ for all i, j . When objective value is less than 100, the sensor field is completed discriminated. Otherwise, it is discriminated with error distance ($Z_{IP4.3}/100$).

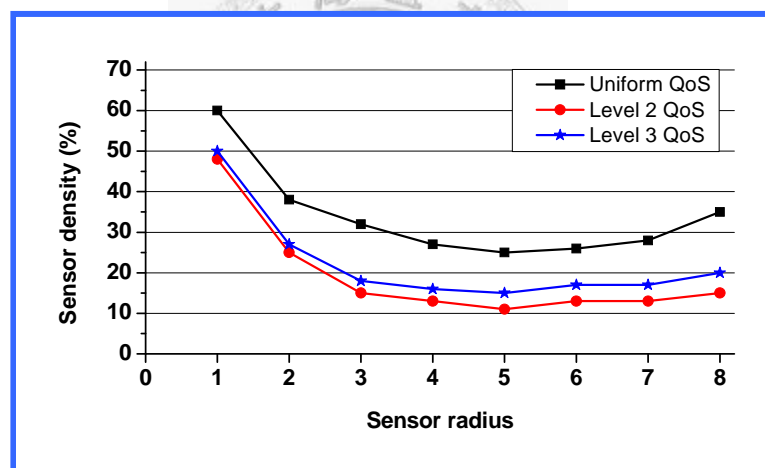


Figure 4.3: The minimum required number of sensor vs. various sensing radius. (Fixed radius)

Figure 4.3 shows the minimum required number of sensor for different levels of QoS requirements. A sensor has a single sensing radius ranging between 1 and 8. The curves marked “Level 2 QoS” and “Level 3 QoS” represent the degree to which Level 2 and Level 3 QoS requirements for ROIs are satisfied, as listed in Table 4.1. The highest curve “Uniform QoS” means

that all service points in Figure 4.1 can be completely discriminated. Figure 4.3 demonstrates that the minimum number of sensors for the sensor network deployment depends strongly on the detection radius of a sensor. Experiment I indicates that using sensors with radius 5 yields the lowest deployment density. In Figure 4.3, we can observe that at the beginning and ending of all curves will trend to up; we give more discussion in Section 4.6.4.

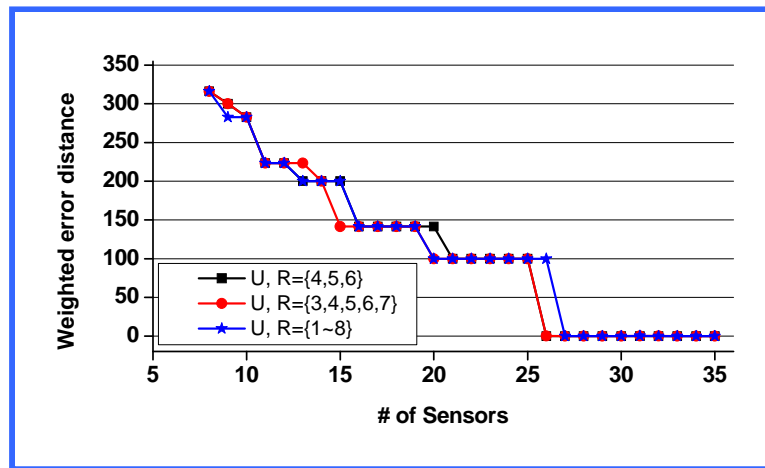


Figure 4.4: The best-found objective values for various set of sensing radius. (Uniform QoS)

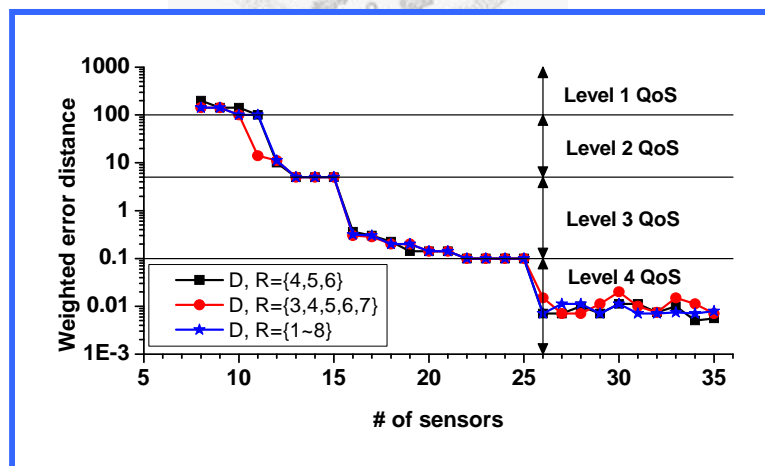


Figure 4.5: The best-found objective values for various set of sensing radius. (Differentiated QoS)

Figure 4.4 and Figure 4.5 show that the best-found objective values of the

uniform QoS and the differentiated QoS respectively. The candidate sets of the sensing radius for sensors were $\{4, 5, 6\}$, $\{3, 4, 5, 6, 7\}$, and $\{1, 2, 3, 4, 5, 6, 7, 8\}$. Table 4.3 lists the ranges of objective values for the differentiated QoS scenario by $Z_{IP4.3}$. These objective values were the weighted error distance and their corresponding quality of positioning service providing by the sensor network. For both the uniform and differentiated QoS, the objective values of the three candidate sets were very close according to Figure 4.4 and Figure 4.5. The candidate radius set $\{3, 4, 5, 6, 7\}$ has a better objective value than others. With these candidate sets, each service point on the field can be completely discriminated when the amount of sensor nodes reaches 26.

Table 4.3: The levels of QoS and their ranges of $Z_{IP4.3}$ in the experiment for differentiated QoS.

| Level of QoS | $Z_{IP4.3}$ (weighted error distance) | Notes |
|--------------|---------------------------------------|----------------------------|
| 1 | $Z_{IP4.3} \geq 100$ | $Z_{IP4.3} \geq w_h$ |
| 2 | $100 > Z_{IP4.3} \geq 5$ | $w_h > Z_{IP4.3} \geq w_m$ |
| 3 | $5 > Z_{IP4.3} \geq 0.1$ | $w_m > Z_{IP4.3} \geq w_l$ |
| 4 | $Z_{IP4.3} < 0.1$ | $Z_{IP4.3} < w_l$ |

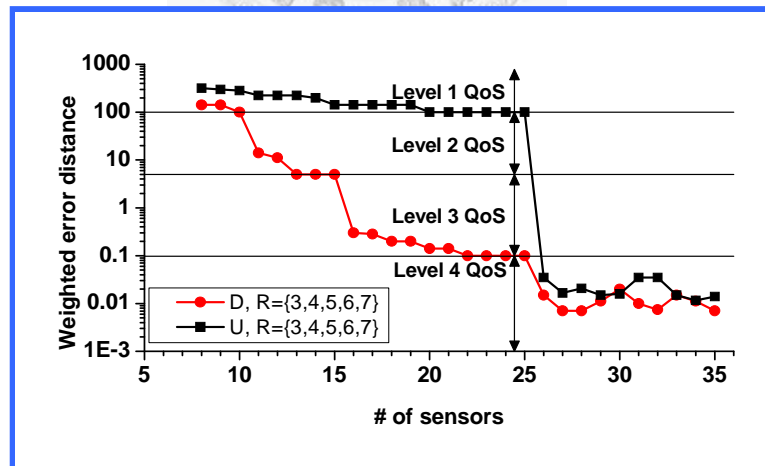


Figure 4.6: Performance comparison between the uniform (U) and differentiated (D) QoS services. (Adjustable radius, $R=\{3, 4, 5, 6, 7\}$)

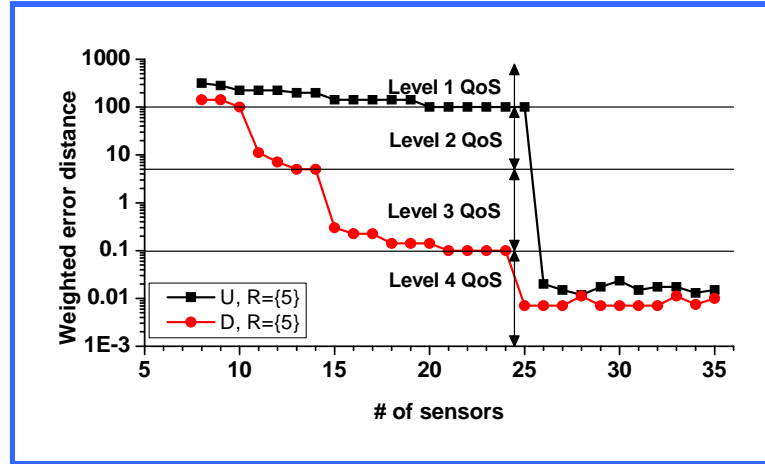


Figure 4.7: Performance comparison between the uniform (U) and differentiated (D) QoS services. (Fixed radius, $R=5$)

Figure 4.6 and Figure 4.7 compare the performance of the uniform and differentiated QoS represented by U and D respectively. The figure only depicts the curves of $R=\{5\}$ and $R=\{3, 4, 5, 6, 7\}$ for the two QoS requirements. These cases achieved the best performance for the scenario of the fixed and various detection radii respectively. Figure 4.6 and Figure 4.7 indicate that the deployment for the network with the differentiated QoS requirement had a better objective value, i.e., the lower weighted error-distance, when the number of deployed sensors was less than 26. That means that service points in ROIs with higher service priority can be discriminated even with a small number of sensors. Moreover, ROIs with lower service priority can obtain a better QoS support (i.e., lower weighted error distance). In Contrast, if a uniform QoS for ROIs is requested, then the service priority for the ROIs cannot be guaranteed effectively. This result confirms the effectiveness of the proposed framework as well as the algorithm.

B. Experiment II

The second experiment was designed to observe the solution qualities of

the proposed algorithm under topography with placement limitations. Placement limitations were added to the topography used in the first experiment: sensors could not be placed in 35 grid points located at the right-upper corner of the museum, as shown in Figure 4.8. Let the deployment cost of the location with placement limitation greater than the deployment budget. Also, the sensor cost in other locations is one. Except for this change, the scenario for this experiment was the same as that in Experiment I.

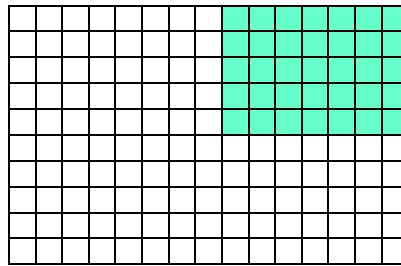


Figure 4.8: The topography with placement limitations.

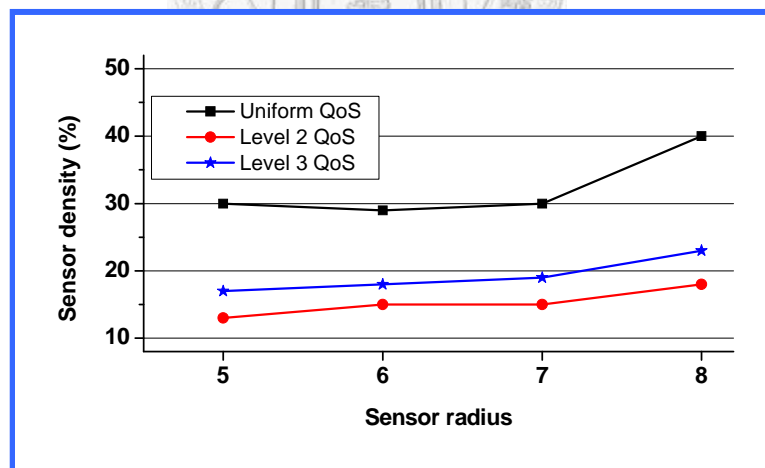


Figure 4.9: The minimum required number of sensor vs. various sensing radius. (Fixed radius, placement limited)

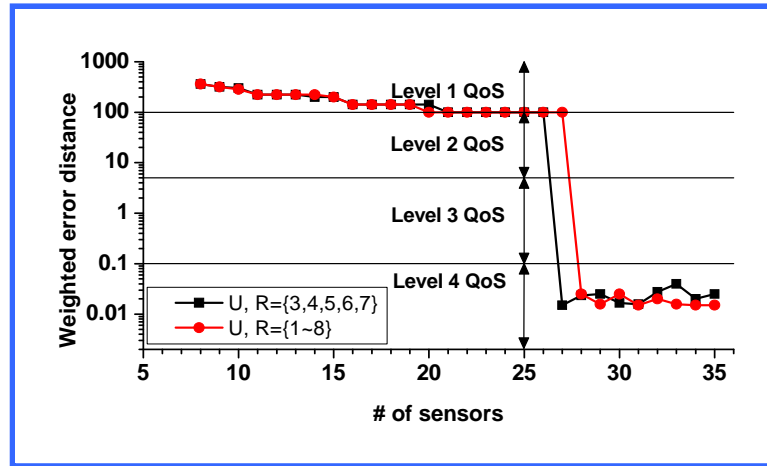


Figure 4.10: The best-found objective values for various set of sensing radius. (Uniform QoS, placement limited)

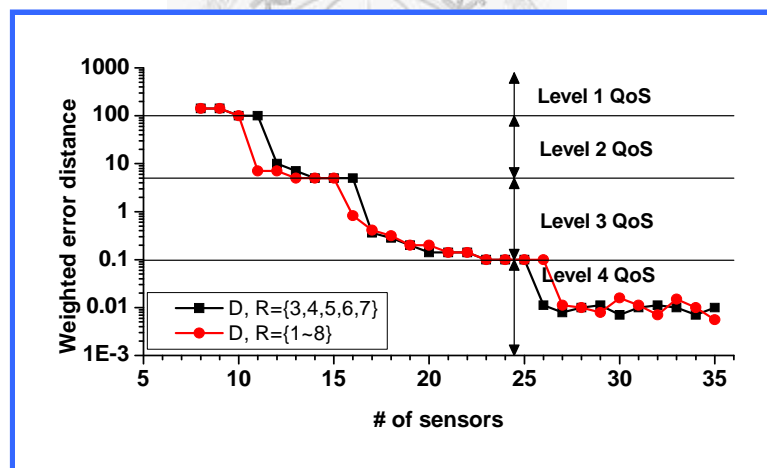


Figure 4.11: The best-found objective values for various set of sensing radius. (Differentiated QoS, placement limited)

From Figure 4.9 to Figure 4.11 provide two observations that are the same as those of experiment I. First, the least number of sensors for different levels of QoS are depended strongly on the selection of sensor radius, when all sensors

have the same radius. Second, different candidate sets of sensing radius obtained almost the same objective value in the case of sensors with various radii. Figure 4.9 indicates that sensors with radius 6 required the minimal number of sensors to achieve uniform QoS (i.e., level 4 QoS). This differs from the result of experiment I, and indicates that both topography and placement limitation dominate the selection of the sensor radius.

Figure 4.12 and Figure 4.13 indicate that even with the placement limitation, the deployment for the network with the differentiated QoS requirement had a lower weighted error-distance whenever the number of deployed sensors is scarce. Moreover, when we adopt the various sensing radius of sensors, we can also observe that both the objective value and the minimum required amount of sensors for each level of QoS are low down. These results confirm the effectiveness of the proposed framework and algorithm in the case of the placement limitation.

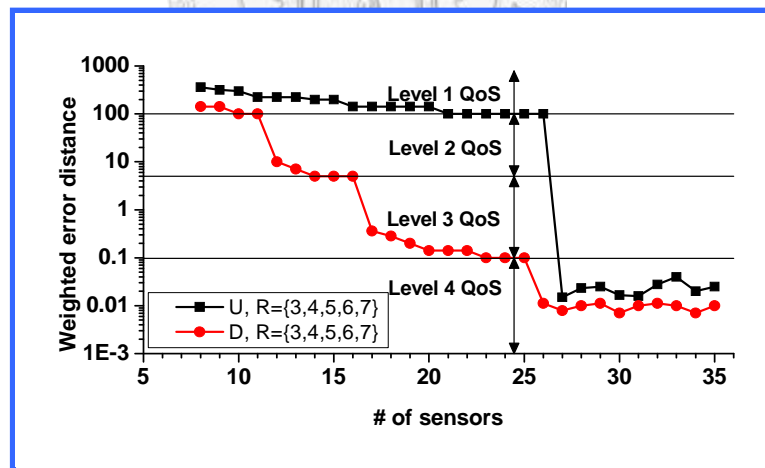


Figure 4.12: Performance comparison between the uniform (U) and differentiated (D) QoS services. (Adjustable radius, placement limited)

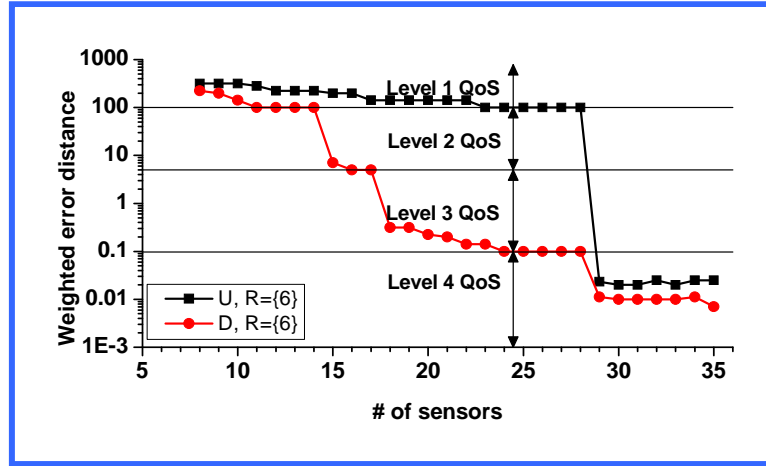


Figure 4.13: Performance comparison between the uniform (U) and differentiated (D) QoS services. (Fixed radius, $R=6$, placement limited)

C. Experiment III

In experiment III, sensor fields 50, 100, 150, and 200 service points were used to evaluate the scalability of the proposed algorithm. The solution space increase exponentially as the sensor field size increased linearly. Therefore, this study observes the variation of computation time and solution quality while the problem size increases.

The parameters about LR algorithm include: $0 < \lambda \leq 2$ and improvement counter is 40. The number of iteration for field size 50, 100, 150, and 200 are 500, 1000, 1500, and 1500, respectively.

Figure 4.14 shows that the minimal requirements of sensor density for various sizes of sensor fields. In the single radius case, the solution spaces of the four cases were 2^{50} , 2^{100} , 2^{150} , and 2^{200} . However, the four curves in Figure 4.14 exhibit the same trend. They indicate that the solution quality of the proposed algorithm is scalable in problem size.

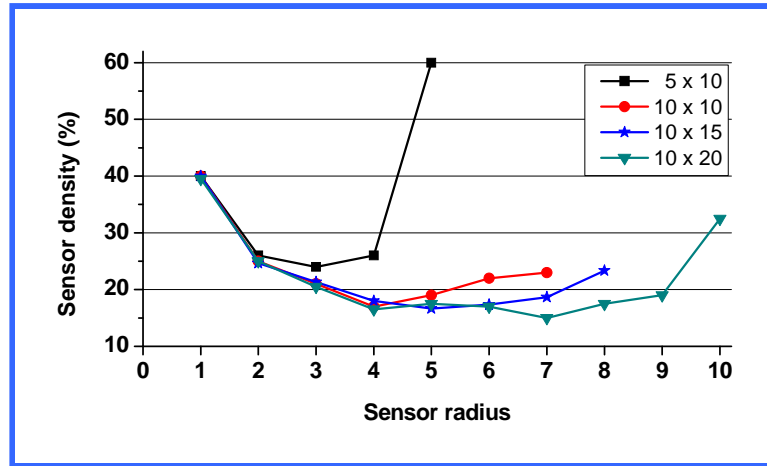


Figure 4.14: The minimum requires sensor density under various sensing radius and sensing area. (LR, Fixed radius)

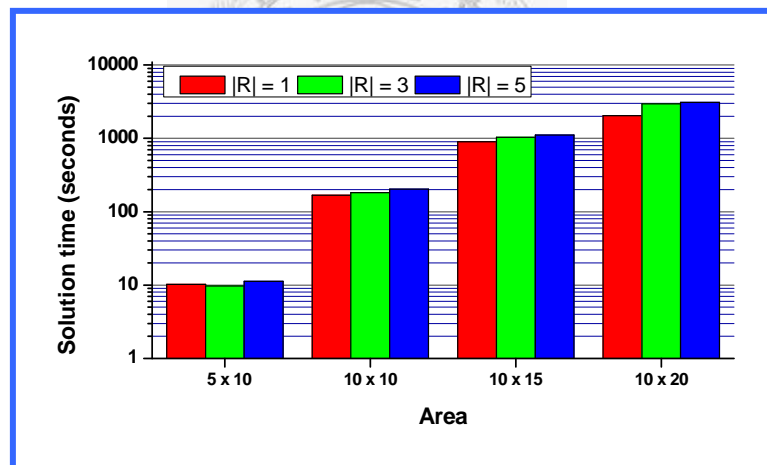


Figure 4.15: The computation time. (LR)

Figure 4.15 presents the computation time of the proposed algorithm, where $|R|$ is the amount of candidate sensing-radius. The solution space of the proposed problem exhibited step growth when $|R|$ increased slightly. However, as shown in Figure 4.15, the computation time did not increase significantly when $|R|$ was increased. These findings clearly indicate that the proposed algorithm is scalable in terms of the candidate sensing-radius. In contrast, the

computation time only increased by a factor of about 17 when the number of service points grew from 50 to 100, and by about 15 times as the number of service point increased from 100 to 200. Results of this experiment indicate that the computation time does not increase exponentially as the solution space grows exponentially. Therefore, the proposed algorithm is also scalable in computation time.



4.5 Simulated Annealing Approach

Simulated annealing (SA) approach is a highly reliable method for solving hard combinatorial optimization problems, *i.e.* it can find the good approximated global optimum solutions for the optimization problems in a large search space. In this section, the original mathematical model, IP4.1, is adopted to develop two SA-based algorithms, noted by SA_1 and SA_2, and to solve the sensor placement problem.

4.5.1 Algorithm SA_1

To simplify the solution procedure, we try to relax the budget and coverage constraints, *i.e.* (4.4) and (4.5), by penalizing objective function $Z_{IP4.1}$ in algorithm SA_1. The penalty function multiplied by original objective is

$$penalty = 1 + p \left(g + \sum_{i \in A} h_i \right)$$

, where constant p is equal to $w_h D \bar{K} N$. D is the diameter of the sensor field. Variable g indicates the exceeding budget. Variable h_i indicates whether service point i is covered. $\sum h_i$ means the number of uncovered service points. Hence, energy E can be defined as follows:

$$E = \max_{\substack{\forall i, j \in A \\ , i \neq j}} \frac{w_{ij} d_{ij}}{1 + \bar{K} \sum_{k \in B} (v_{ik} - v_{jk})^2} \left(1 + p \left(g + \sum_{i \in A} h_i \right) \right).$$

Algorithm 4.2 presents the pseudo code of algorithm SA_1. Initially, sensors are randomly deployed under the budget limitation. The radius of all sensors is set to the maximum one in the candidate radius set. In each loop, the

solution configuration is randomly altered by one of following actions: adding one sensor, reducing one sensor, changing one sensor's location, increasing one sensor's radius, or reducing one sensor's radius. The solution with the minimum energy is saved as the best found solution. The terminate condition of the algorithm is either reaching the frozen temperature t_f or satisfying the desired level of QoS for each block. The energy of best found solution, Z_{min} , can be used to verify whether the latter condition reached. For example, if we want to deploy a completely discriminated sensor network, Z_{min} has to be less than the lowest discrimination weight. When the algorithm stopped, it is a feasible solution to this problem, if no penalty for the best found solution.

Algorithm 4.2: The pseudo code of algorithms SA_1 and SA_2.

```

1. According to the budget, randomly deploy sensors on the sensor field.
2. Calculate  $E$  for the initial guess.
3.  $Z_{old} \leftarrow E$ .
4. Let  $Z_{min} \leftarrow Z_{old}$  and save the current configuration as the best solution.
5. Evaluate  $Z_{min}$ , if desired QoS requirement for each block is satisfied then
   goto step (21).
6.  $t \leftarrow t_0$ ,  $b \leftarrow b_0$ .
7. While  $t > t_f$  do
8.   Repeat  $r$  times
9.     Randomly alter the solution configuration.
10.    Calculate  $E$  for the new configuration.
11.     $Z_{new} \leftarrow E$ .
12.    Evaluate  $Z_{min}$ , if desired QoS requirement for each block is satisfied
     then goto step (21)
13.     $\Delta E \leftarrow Z_{new} - Z_{old}$ .
14.    Generate a random number  $\rho$  uniformly distributed in (0,1).
15.    If  $\Delta E \leq 0$  or  $\rho < e^{(-\Delta E/t)}$  then
16.       $Z_{old} \leftarrow Z_{new}$ 
17.      If  $Z_{old} < Z_{min}$  then  $Z_{min} \leftarrow Z_{old}$ , save current configuration as the
       best solution.
18.    else recover the action in step (9).
19.  End
20.   $b \leftarrow b * \beta$ ,  $t \leftarrow t * \alpha$ 
21. End

```

4.5.2 Algorithm SA_2

Besides relaxing the budget and coverage constraints as algorithm SA_1, we consider the number of service points with the maximal weighted error distance to construct a surrogated energy function E_s :

$$E_s = E * \left(1 + \frac{p'}{|A|} \sum_{i \in A} m_i \right).$$

Where m_i indicates whether the service point i has the maximal weighted error distance with another service point. And, $\sum m_i / |A|$ means the ratio of the service point with the maximal weighted error distance. Constant p' is the minimum gap size between any two adjacent objective values. The value of p' must be computed carefully to avoid the part of penalty over the gap, which is between current original objective and the higher candidate objective values.

The solution procedure of algorithm SA_2 is the same as algorithm SA_1, as listed in Algorithm 4.2, except for using E_s substitutes for E .

4.5.3 Computational Results

Two sets of experiments were conducted to evaluate the performance of the proposed algorithm under various settings for the numbers of priority class, amount of resource, topology area and detection radius. The proposed algorithms were coded in C in MS-VC++ 6.0 development environment. All the experiments were performed on a P4-3.0GHz PC running MS-Windows XP Pro. The performance metrics were assessed in terms of the solution quality and computation time.

The parameters of the cooling schedule are $\alpha=0.7$, $\beta=1.3$. The initial value of b_0 and t_0 are 2000 and 0.001, respectively. As well as the frozen temperature t_f is $t_0/2000$. In addition, sensor cost, c_k , is set to one.

A. Experiment I

The first experiment was designed to observe the solution qualities of the proposed algorithms. The sensor field topology was based on the above museum example, as shown in Figure 4.1.

For the differential positioning accuracy, the set of discrimination weights was set to $\{0.1, 5, 100\}$. The discrimination weight between any two service points on ROI A was set to the high weight, $w_h=100$. The discrimination weight between any two service points, which both on ROIs B, C, D, or E, was set to the medium weight, $w_m=5$. The weights between any two service points, which both are on ROI F was set to the low weight $w_l=0.1$. The weight between any pair of service points on different ROIs was set to the highest weight, $w_h=100$. The diameter of the sensor field was set to $D=16$. Parameter $\bar{K} = 20000$.

Table 4.4: The min. required sensors (S) for level 4 QoS in SA_2.

| | single radius | | | | | | | | adjustable radius | | |
|-----|---------------|----|----|----|----|----|----|----|-------------------|-----|-----|
| R | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 4~6 | 3~7 | 1~8 |
| S | 59 | 36 | 30 | 24 | 24 | 24 | 25 | 29 | 25 | 24 | 25 |

Table 4.4 demonstrates that the minimum number of sensors for the sensor network deployment depends strongly on the detection radius of a sensor, while the sensor is with a single radius. Contrary, while an adjustable radius is used, S is very stable relatively. With these candidate sets, each service point on the field can be completely discriminated when the amount of sensor nodes reaches 25. This result shows that adapting sensors with adjustable radius for the problem is more flexible than using a single radius. This experiment indicates that using sensor with radius 3~7 yield the lowest deployment density.

Figure 4.16 shows the best-found objective value for SA_2. Table 4.3 lists the ranges of $Z_{IP4.1}$. Figure 4.16 indicates that the deployment for the network with the differentiated QoS requirement had a lower weighted error-distance when the number of deployed sensors was less than 24. It means that the

different level of QoS can be satisfied for ROIs according to their service priority even with a small number of sensors. Moreover, the QoS degradation for ROIs with lower service priority is minimized. In contrast, if a uniform QoS for ROIs with lower service priority is requested, then the service priority for the ROIs cannot be guaranteed effectively. This result confirms the effectiveness of the proposed framework as well as algorithm SA_2.

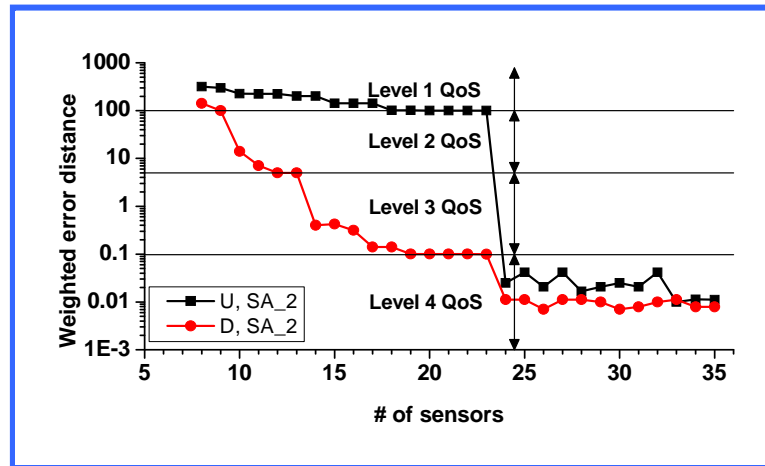


Figure 4.16: Performance comparison between the uniform (U) and differentiated (D) QoS services. (SA_2, $R=\{3, 4, 5, 6, 7\}$)

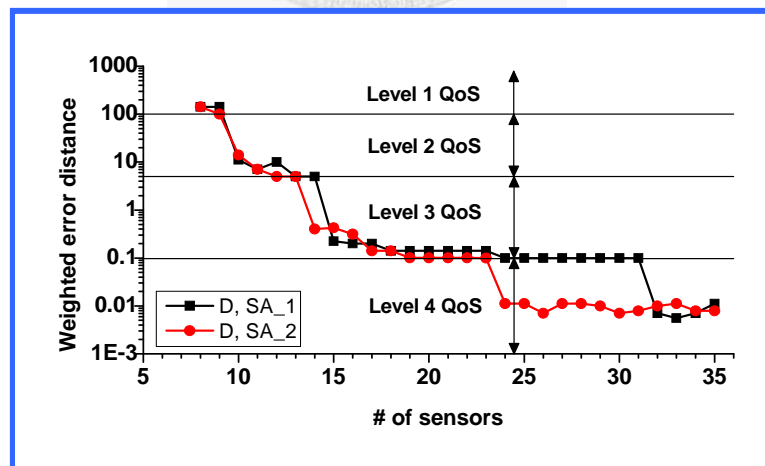


Figure 4.17: Performance comparison between algorithms SA_1 and SA_2. (Differentiated QoS service, $R=\{3, 4, 5, 6, 7\}$)

Figure 4.17 indicates that if algorithm SA_2 is applied, each ROI can get level 4 QoS when the amount of sensor nodes reaches 24. However, it requires 32 sensor nodes if algorithm SA_1 is used. This result shows the SA-based algorithm with a surrogated energy function, SA_2, can convergence more effectively than SA_1, which applies a simple penalized energy function.

B. Experiment II

In experiment II, sensor fields 50, 100, 150, and 200 service points were used to evaluate the scalability of the proposed algorithm.

Figure 4.18 shows that the minimal requirements of sensor density for various sizes of sensor fields. In the single radius case, the solution spaces of the four cases were 2^{50} , 2^{100} , 2^{150} , and 2^{200} . However, the four curves in Figure 4.18 exhibits the same trend and indicate that the solution quality of the proposed algorithm is scalable in problem size.

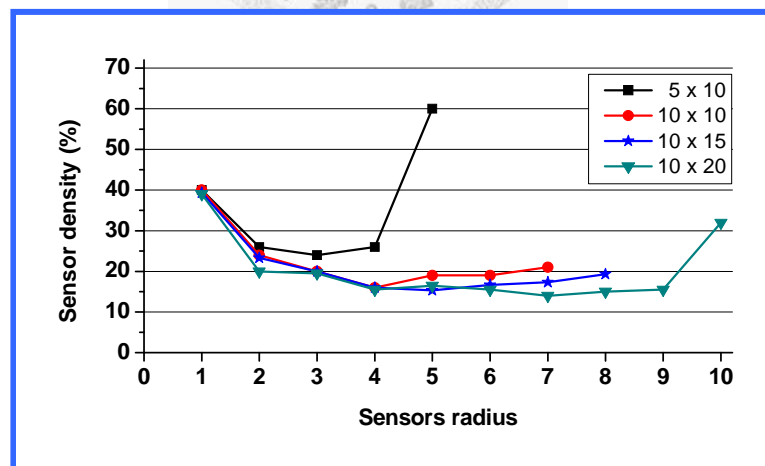


Figure 4.18: The minimum requires sensor density under various sensing radius and sensing area. (SA_2, Fixed radius)

Figure 4.19 shows the computation time of algorithm SA_2. The solution space of the proposed problem exhibited steep growth when candidate radii increased slightly. However, the computation time did not increase significantly. These findings clearly indicate that SA_2 is scalable in terms of the candidate sensing-radius. Moreover, the computation time only increased about 12 times when the number of service points grew from 50 up to 200. Results of this experiment indicate that the computation time does not increase exponentially as the solution space grows exponentially. Therefore, the proposed algorithm is also scalable in computation time.

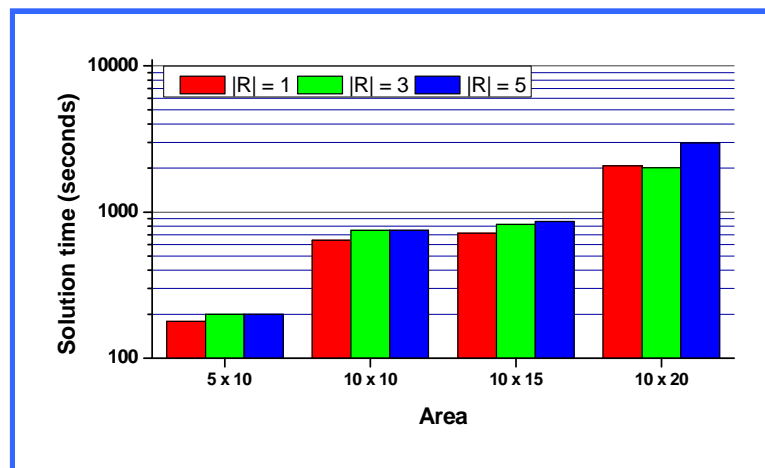


Figure 4.19: The solution time (seconds) of SA_2 in various areas.

4.6 Performance Comparisons

In this section, due to the previous experiments results, the performance and convergence properties of the proposed algorithms, LR algorithm, SA_1, and SA_2, will be compared.

4.6.1 Performance Evaluations

Figure 4.20 depicts the curves of objective values versus the given number of sensors in the case of no placement limitation and $R=\{3, 4, 5, 6, 7\}$. Each ROI can get level 4 QoS when amount of sensor is 24, 26, and 32 by SA_2, LR, and SA_1, respectively. Therefore, for this problem, the SA_2 using surrogate energy function overcomes the other algorithms, LR and SA_1, in solution quality.

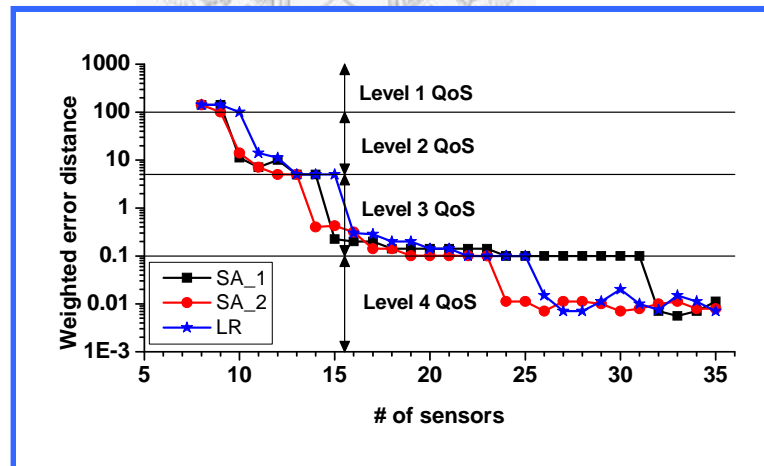


Figure 4.20: Comparison of solution quality between SA_1, SA_2, and LR algorithms. (Area: 10x15, differentiated QoS, $R=\{3, 4, 5, 6, 7\}$)

4.6.2 Convergence Properties Analysis

Several famous searching methods usually are adopted or modified to cope with the unconstrained or constrained nonlinear programming problem, for instance, Golden Section Search, Newton's Method, Steepest Decent Method, Gradient Projection Method, et al. These algorithms have a common heritage of all being *iterative descent algorithms*. We mean that the algorithms iteratively generate a series of points, each point being calculated on the basis of the points preceding it. As well as iteration by iteration, the points corresponding to objective function will decrease. Ideally, the sequence of points generated by the decent algorithm in this way converges in a finite or infinite number of steps to a solution of the original problem. If for arbitrary starting points, the algorithm is guaranteed to converge to a solution, then the algorithm is said to be *globally convergent*. In addition, the analysis of *convergence rate* is another important subject. It can be used to evaluate the effectiveness for iterative descent algorithm [Lue84].

In this dissertation, all proposed problems are formulated as combinatorial optimization problems. Due to the above mentioned searching methods are not applicable to solve these NP-hard problems, we propose Lagrangean relaxation based and simulated annealing based algorithms to cope with these problems. The proposed algorithms do not belong to iterative descent approach. Iteration by iteration, the series of feasible solutions are not monotonic non-increasing. Hence, the convergence rate is replaced with *convergence trend* to observe the effectiveness of the proposed algorithms.

(A) Lagrangean Relaxation Algorithm

The subgradient method is an adaptation of the gradient method in which gradients are replaced by sub-gradients. The subgradient method is easy to program and has worked well on many practical problems, hence, it has become the most popular method for Lagrangean dual problem [Fis81].

In reference papers [HWC74] [Gof77], the convergence properties of the subgradient method are discussed. To ensure convergence, all step sizes t^k have to satisfy the following constraints:

$$t^k \rightarrow 0 \text{ and } \sum_{i=0}^k t^k \rightarrow \infty.$$

In addition, we observe the convergence trend of Lagrangean relaxation algorithm with different values of improvement counter. Based on the map shown in Figure 4.1, without placement limitation, in the case of differentiated QoS requirement and $R = \{3, 4, 5, 6, 7\}$, given number of sensors, 25, we observe the upper bounds on objective function under different values of improvement counter. We use improvement counters, 20, 30, 40, 50, and 60, in our experiments. The experimental results are illustrated in Figure 4.21, only 3 curves (improvement counter are 40, 50, and 60) are depicted. The improvement counter has to be larger than or equal to 50, the proposed LR algorithm will convergent and solutions will be less than 0.1.

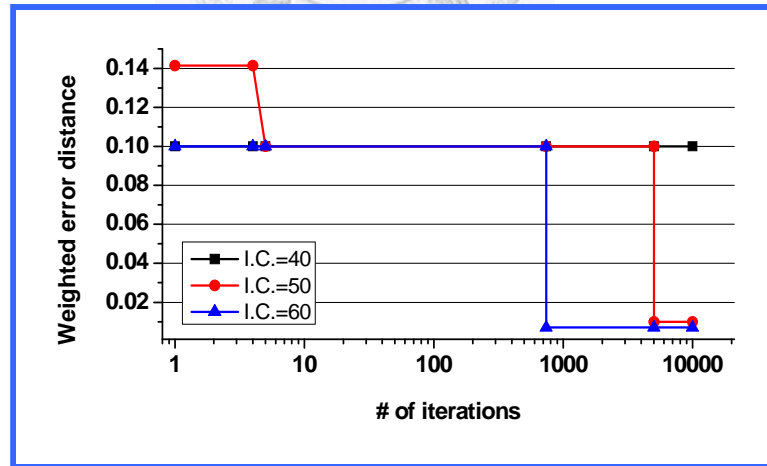


Figure 4.21: Convergence trend of LR algorithm. (Area: 10 by 15, differentiated QoS, $R = \{3, 4, 5, 6, 7\}$, number of sensor is 25, I.C.=Improvement Counter)

Afterwards, the LR algorithm executes using radius sets $\{5\}$, $\{3, 4, 5, 6, 7\}$, and $\{1\sim 8\}$, corresponding to 1, 5, and 8 candidate radii. The test improvement counters are 20, 30, 40, ..., until the feasible solution convergent. We find when number of candidate radii increases, the minimum improvement counters for convergence will increase, shown in Table 4.5. So the improvement counter setting is case by case.

Therefore, we obtain that the upper bound is dependent on the improvement counter setting in terms of problem size.

Table 4.5: The minimal improvement counters for convergence in 150 grid points sensor field.

| | Candidate radius | | |
|------------|------------------|-----------------|---------------|
| | $\{5\}$ | $\{3,4,5,6,7\}$ | $\{1\sim 8\}$ |
| Min. I. C. | 20 | 50 | 80 |

(B) Simulated Annealing Algorithm

The cooling schedule of simulated annealing has to be design carefully for obtaining a good approximated optimal solution. Hence, we design a set of experiment to observe whether the parameters: initial temperature, cooling ratio, and initial number of iteration will affect the convergence properties of SA_2. The context of the experiments for observing convergence trend include 150 service points sensor field without placement limitation, deployed 24 sensors with multiple candidate detection radii, $R=\{3, 4, 5, 6, 7\}$. The SA parameters are setting as follows: $\alpha =0.75$, $\beta =1.3$, $r_0 =2000$, $T_0 =0.000588$, and $T_f=T_0/20000$.

Figure 4.22 shows the convergence trend for different cooling ratio α . Typically, α is about 0.75, the higher α enables a slowly decrease in temperature. The spending time of SA_2 from iteration 1 to 10^4 is very few,

about several seconds. In despite of α being 0.5, 0.6, or 0.7, SA_2 always can converge to objective value 0.1 quickly. But only α is 0.7, SA_2 will converge to less than 0.1. Therefore, the value of cooling ratio α will affect the convergence trend in SA_2.

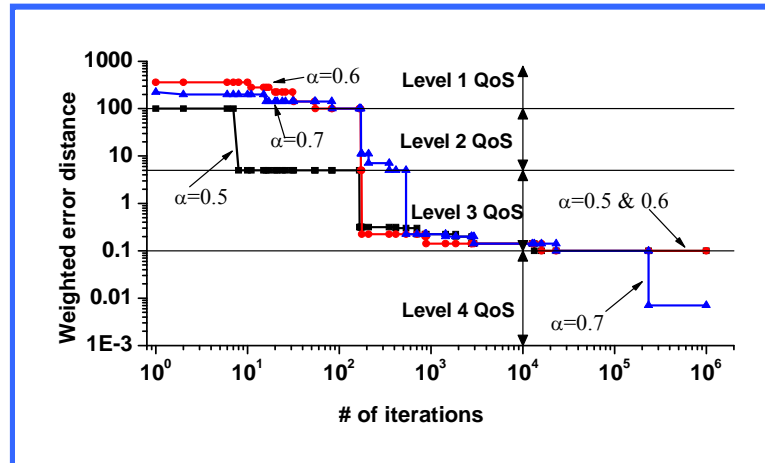


Figure 4.22: Convergence trend of SA_2 for different cooling ratio.

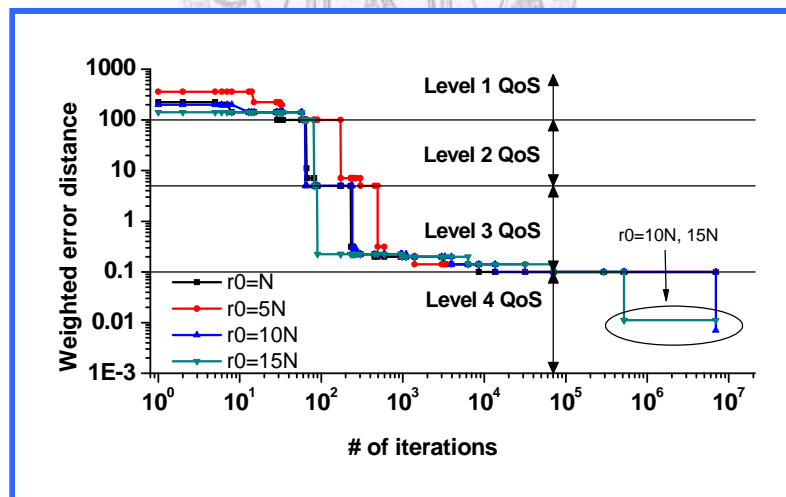


Figure 4.23: Convergence trend of SA_2 for different r_0 . ($N=150$)

Figure 4.23 illustrates the convergence trend for different initial number of iteration, r_0 , which determines number of iterations, as well as physically affects the balance for each temperature level. We let r_0 be multiple of number of grid

point, N . In despite of r_0 being 1, 5, 10, or 15 multiple of N , SA_2 always can converge to objective value 0.1 quickly. But r_0 has to be greater than or equal to $10N$, SA_2 will converge to less than 0.1. Therefore, we can obtain the convergence trend in SA_2 depends on the parameter r_0 .

Finally, we investigate how does the initial temperature T_0 affects the convergence trend of the SA_2 algorithm. The initial temperature T_0 is related to initial transition probability p_0 , $p_0 = \exp(-\Delta E / T_0)$. We let ΔE be the minimum difference between any two adjacent levels of objective value, and set $p_0=0.2, 0.5$, and 0.8 . Then we get three different initial temperatures T_0 , (i.e., 0.001287, 0.000588, and 0.000253), corresponding to three initial transition probabilities p_0 , (i.e., 0.8, 0.5, and 0.2). In Figure 4.24, we can observe that all curves can reach the region of level 4 QoS. This evidence shows that the SA_2 algorithm is insensitive in the variation of T_0 .

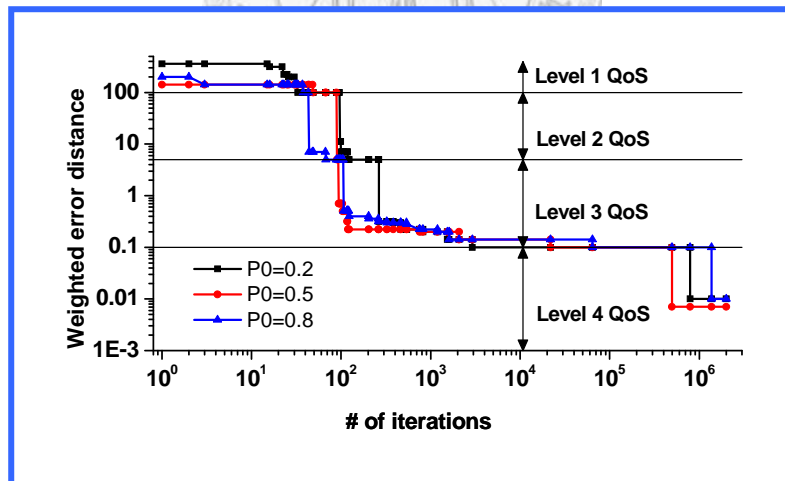


Figure 4.24: Convergence trend of SA_2 for different T_0 (or p_0).

4.6.3 Compare with Other Approaches

4.6.3.1 Compare with CIQ Approach

In Section 2.4.1, we review the papers [CIQ01] [CIQ02], which apply the coding theory to solve the target location problem in sensor networks. In the dissertation, we note the placement method proposed by Chakrabarty et al. as “CIQ approach”. The simpleness and quickness are main advantages of CIQ placement method. However, CIQ approach can not be used for irregular sensor field, and can not use various radii sensors to solve target location problem. In addition, it ignores the sensor field boundary effect. Our approach can address these difficulties.

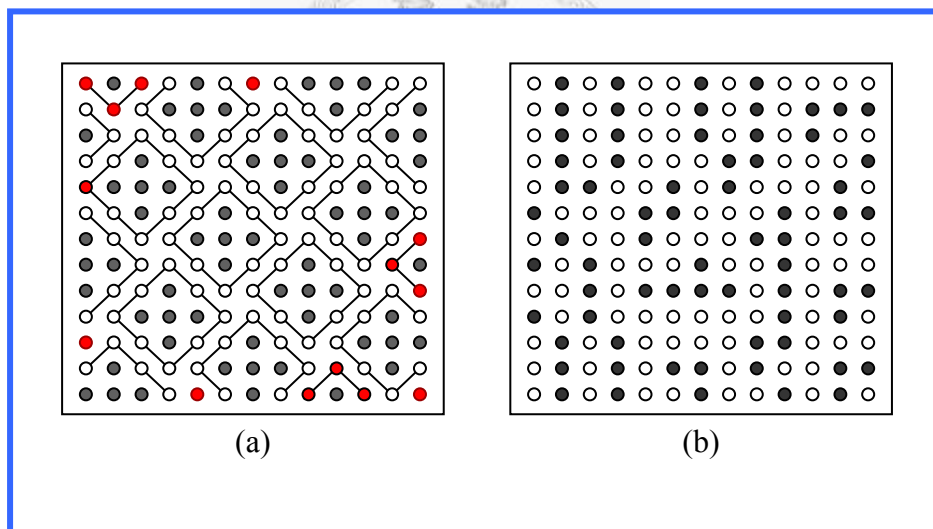


Figure 4.25: The sensor deployment in a 13x13 sensor field. (a) By CIQ approach (79 sensors). (b) By LR approach (68 sensors).

Figure 4.25(a) shows a 13x13 sensor field which is deployed 65 sensors by CIQ method. To solve the boundary problem, we deploy 14 extra sensors (total 79 sensors) for satisfying the completely discrimination constraint. In the same scenario, i.e., 13x13 sensor field and uniform discrimination weight, we use sensor with radius one to deploy a completely discriminated sensor field by

Lagrangean relaxation algorithm. The deployment by LR approach requires only 68 sensors; it is illustrated in Figure 4.25(b). We can claim that our approach outperform CIQ method in terms of deployment cost.

4.6.3.2 Compare with ID-CODE Algorithm

In Section 2.4.1, we have been reviewed the papers [RST04] [RUP03], which apply the identifying code to solve the target location problem in sensor networks. They propose the “ID-CODE” sensor placement algorithm, and design three visiting orders: random, ascending, and descending orders.

However, ID-CODE approach does not consider sensors with adjustable radius for the target location problem. In addition, the algorithm does not also take different discrimination quality into account. Contrary, our approach can address these issues.

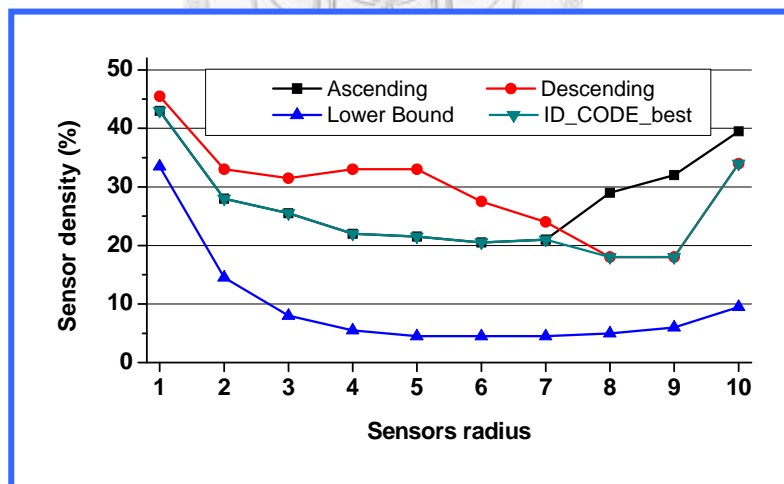


Figure 4.26: Performance of the ID-CODE algorithm in 150 grid points sensor field.

For comparing the proposed algorithms, LR and SA_2 algorithms, with ID-CODE, we design a set of experiments. Due to the results that were obtained by the random order of the ID-CODE approach highly depend on probability. It

will result in either a larger deviation or too much time consumption to obtain a statistical or the best result. Hence, as Figure 4.26 shown, in each experiment, we only adopt the ascending and descending orders of the ID-CODE algorithm. Furthermore, in each simulation scenario, we take the best results, denoted by “ID_CODE_best”, of these two visiting orders as a benchmark of the ID-CODE algorithm. The curve denoted “Lower Bound” is a modified version of lower bound according to Theorem 1(2) and (3) in [KCL98].

From Figure 4.27(a) to (d) illustrate performance comparisons between ID-CODE, SA_2, and LR algorithms under various areas of sensor fields, i.e. 50, 100, 150, and 200 grid-points, respectively. We can observe the SA_2 and LR algorithms outperform ID_CODE_best in terms of deployment cost, i.e., number of sensors.

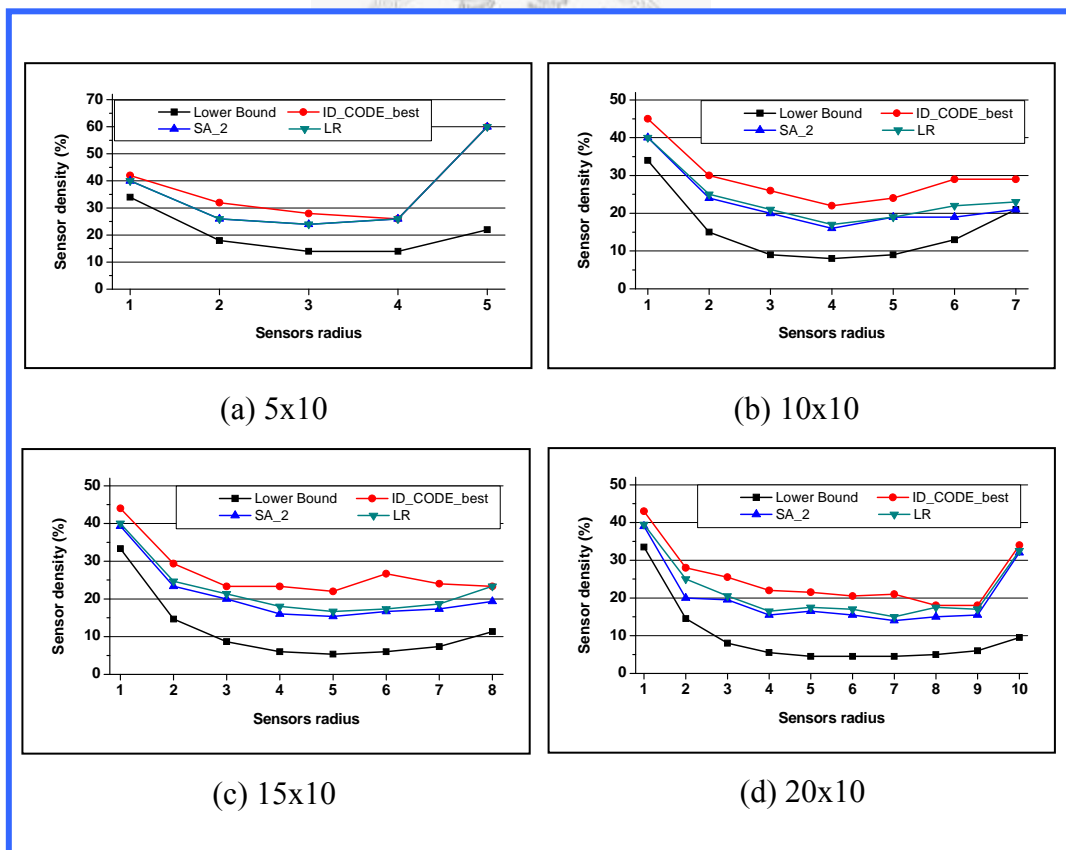


Figure 4.27: Performance comparisons between ID_CODE_best, SA_2, and LR algorithms under various areas of sensor fields.

4.6.4 Results Analysis

From the previous experimental results, we are concerned with an interesting phenomenon. That is, the minimum required number of sensors for the positioning services will increase if the detection radius is larger or smaller, as Figure 4.27 shown. We discuss and investigate from two different directions.

1. By theoretical and mathematical analysis

From information theorem perspective, while sensor detection radius is larger or smaller, the corresponding information (or entropy) of the sensor will decrease. So that, more sensors are required to satisfy the entropy for constructing identifying code in a sensor network. The detail proof is illustrated in Appendix B.

2. By experimental observations

From the experimental results (in Appendix C), we can find that the sensor with smaller radius needs more for supporting target positioning. But when the detection radius closes to the maximum candidate radius for the size of sensor field, the required density will increase according to the boundary effect of the sensor field.

4.7 Concluding Remarks

In this chapter, we propose a generic framework for the sensor placement problem to support the differentiated QoS and prioritized service for different ROIs in a sensor field. The weight is designed to assign the discrimination priority of two service points. Under the budget, coverage, and placement constraints, the goal is to minimize weighted error distance. Besides the sensor's locations, we consider the detection ranges as decision variables and construct the mathematical model for the sensor deployment problem. Next, we develop a Lagrangean Relaxation based heuristic, two SA-based algorithms, SA_1 and SA_2, to scope with the NP-complete problem.

Based on the experiment results, we make a summary as follows:

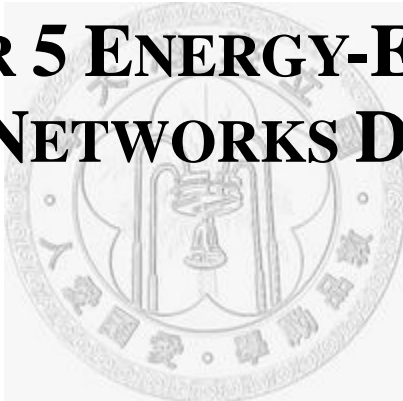
1. The proposed framework can support a better and differentiated QoS than the pervious framework, which can only handle uniform QoS requirements.
2. The performance of the proposed algorithms is almost independent of the radius selection when it adapts sensors with an adjustable sensing-radius. By the various radii, the proposed algorithm can efficiently get well solution quality. Particularly, some locations can't be placed sensors in sensor field, all the proposed algorithms use various radii to cope with the problem effectively.
3. The proposed algorithms are scalable in terms of the number of the radius types and the size of the sensor field.
4. It is independent of the size of sensor field that the sensor density requirement will increase if the radius is shorter or longer.

5. The proposed SA-based algorithm with a surrogated energy function, SA_2, is convergence more effectively than the LR algorithm and SA_1 in solution quality. It is the same that the LR algorithm overcomes SA_1 in solution quality.
6. When size of sensor field is small, the proposed LR algorithm is more efficient than SA_1 and SA_2. When size of sensor field increases, the efficiency of three proposed algorithms come up with no significant difference.
7. The proposed approaches (LR and SA_2) are more effective and flexible than the CIQ and ID-CODE approaches in terms of various sensor radii, terrain as well as other deployment constraints. The main drawbacks of our approaches are more computation time and less scalability.





CHAPTER 5 ENERGY-EFFICIENT SENSOR NETWORKS DESIGN



5.1 Overview

Both sensor deployment and energy conservation are key issues for WSNs [ASC02a]. This work considers the problem of constructing an energy-efficient sensor network for surveillance and target positioning services using the controlled placement approach. The design goals are to achieve target positioning as well as to prolong sensor network lifetime. To support positioning functionality, the sensor field must be completely covered and each unit in the field is discriminable. It requires deploying more sensors than to support surveillance functionality. However, to keep all sensors in active to support the target positioning service is not necessary and waste sensors' energy if intrusion

events occur infrequently. Actually, the surveillance service is enough when there isn't any intruder in the sensor field. Hence, we try to deploy K independent sets of sensors to support positioning service on a sensor field. Each of them, is called a cover, can provide complete coverage of the field. Each set is activated in turn to monitor the field when no any intruder was existence. Once the intrusion event occurs, all sets of sensors are activated and work corporately to locate the intruder. Generally, the power consumption for inactive sensors can be neglected, and the sensor lifetime can be effectively prolonged up to K times.

In this chapter, we formulate the problem as a 0/1 integer programming problem where the objective function is the minimization of the total deployment cost required to complete coverage and discrimination constraints under a given amount of cover K . The problem is a variant of the set K -cover problem and thus is NP-complete [AGP04] [SP01]. Then, the Lagrangian Relaxation (LR) based algorithm and Simulated Annealing (SA) based algorithm are proposed to address the optimization problem [Egl90].

From sensor placement perspective, the energy conservation strategy can be considered in “deployment phase” or “post-deployment phase”. This study belongs to the former, and focuses on energy efficient sensor network deployment.

From papers review, we find that this study differs from prior works in several points. First, we consider both the energy conservation and lifetime extending during the sensor deployment phase for target positioning. Second, we present a mathematical model to describe the optimization problem. Third, the LR-based algorithm and the SA-based algorithm are proposed to solve the problem. Fourth, the relationship between the deployment cost and the maximum extension of system lifetime is investigated. Finally, the performances of the proposed algorithm are evaluated and compared with CPLEX.

5.2 Problem Description

A. Sensor Placement

In this chapter, we use the controlled deployment method to construct WSNs. The sensor field can be represented as a collection of two- or three-dimensional grid points [CIQ02] [CL04] [DC03] [DCI02] [LC05], as illustrated in Figure 5.1. This approach is called grid-based placement. The positioning resolution requirement of application determines the granularity of grid point. In this chapter, the distance between two adjacent grid points is adopted as a length unit. Therefore the sensor field illustrated in Figure 5.1 has 5 by 3 grid points. And six sensors are placed on grid point 4, 6, 7, 9, 10, and 12.

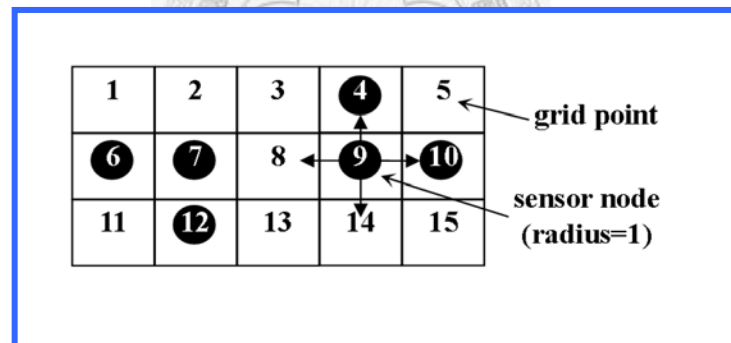


Figure 5.1: A complete coverage/discrimination sensor field.

This study assumes that the sensor detection model is 0/1 model [CIQ02] [CL04] [LC05]. The coverage is assumed to be complete (1) if the distance between the grid point and the sensor is less than the detection radius of the sensor. Otherwise, the coverage is assumed to be incomplete (0). For example, the radius of the sensors illustrated in Figure 5.2 is assigned to one. It is a homogeneous sensor network. Therefore sensor 4 covers grid point 3, 4, 5, and 9, sensor 7 covers grid point 2, 6, 7, 8, and 12, sensor 9 covers grid point 4, 8, 9, 10,

and 14, and so on. If any grid point in a sensor field can be detected by at least one sensor, the field is called completely covered sensor field.

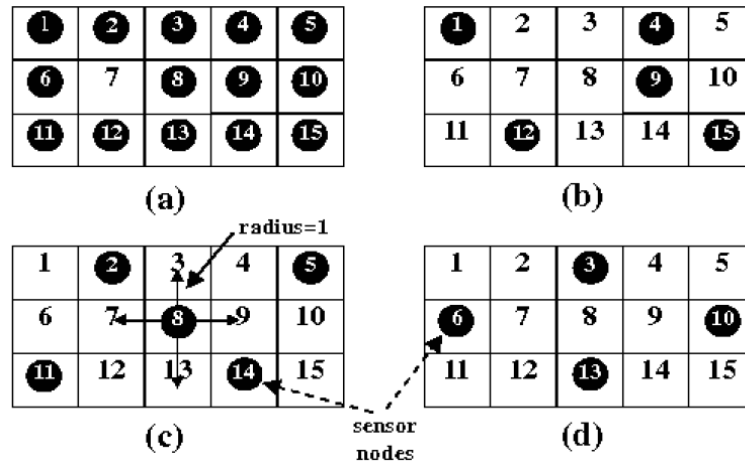


Figure 5.2: A grid-based sensor field with 3 covers: (a) Overall placement. (b) Cover 1. (c) Cover 2. (d) Cover 3.

To locate an intruder, we define a unique power vector for each grid point. The power vector of a grid point is constructed according to the deployment of sensors. If a sensor covers the grid point, constituent of the power vector of the grid point, which is corresponding to the sensor, is set to 1, otherwise 0. For example, as illustrated in Figure 5.2, the power vectors of grid point 1 and 8 are $\langle 0, 1, 0, 0, 0, 0, 0 \rangle$ and $\langle 0, 0, 1, 1, 0, 0 \rangle$ corresponding to sensor 4, 6, 7, 9, 10, and 12, respectively. After all of the sensors are deployed, the power vectors for each grid point in the sensor field are constructed and stored on the database at the back-end of the network. Once an intruder was detected, sensor has to report the information to the sink nodes. According to the received information, the back-end can obtain a power vector to determine the position of the intruder. If each grid point has a unique power vector in a sensor field, the sensor field is called completely discriminated. The sensor field in Figure 5.2 is completely covered/discriminated by the sensor network, which can provide surveillance and target positioning services.

B. Energy-Efficient Sensor Networks

The duplicate placement approach is an intuitive method to extend system lifetime in deployment phase. The algorithm deploys K duplicate sensor networks, which provide surveillance and target positioning services in turn to prolong lifetime by K times. However, the total cost is increased by K times accordingly.

This study attempts to construct the sensor network such that it includes K mutually exclusive sets (number K is given). These sets are called cover [SP01]. Figure 5.3 shows the state transitions of the sensor network. From the network viewpoint, two operation states exist: the surveillance and positioning states. When no any intruder exists in the sensor field, the network operates in the surveillance state. At the period, the K covers of sensors are activated in turn to monitor the whole sensor field. Each sensor may be in sleeping or monitoring states. Once the intrusion event occurs, the network transits to the positioning state. All covers of the sensors are activated and work corporately to locate intruder. At the period, all sensors on the network operate in active state.

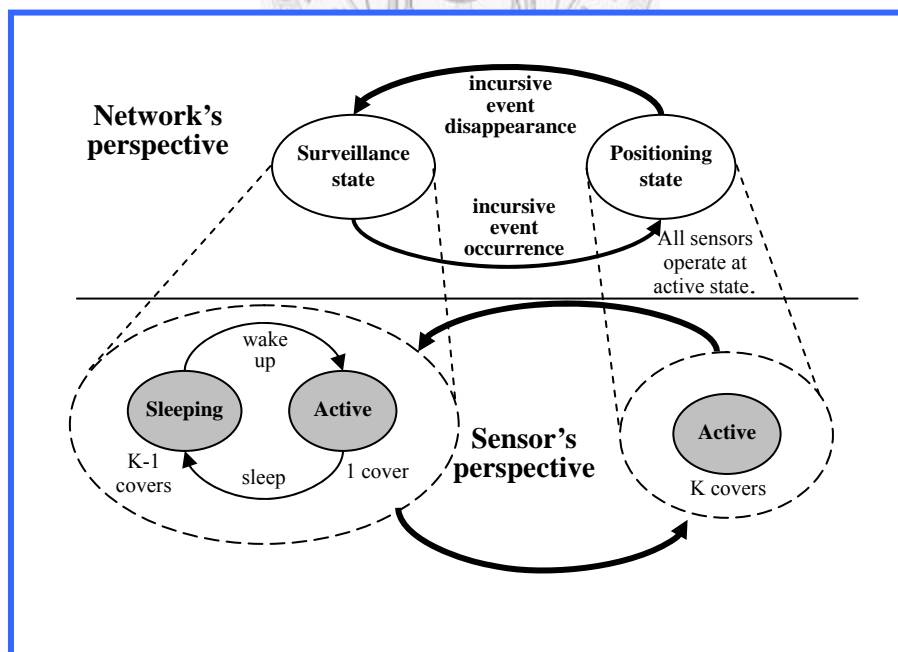


Figure 5.3: The state diagram of the sensor network.

We assume that a sensor has multiple power states, active, monitoring, and sleeping state [SCI01]. There are three main components in sensor nodes: processor, sensor, and radio transceiver. Table 5.1 presents the working modes of the sensor's components corresponding to the three different power states. If there is not any intruder approaching to the sensor field, only one set of sensors operates at monitor state and the others operate at sleeping state. Once intrusion event occurs, all of the sensors transit to active state. In this study, we assume that the intrusion event is infrequently, so only monitor and sleep power states of sensors must be focused.

Table 5.1: The working modes on three states for the sensor node. The radio is dominant power consumer.

| Working modes | | Components of sensor node | | |
|---------------|---------|---------------------------|--------|-------|
| | | Processor | Sensor | Radio |
| Power states | Active | Active | On | Tx/Rx |
| | Monitor | Idle | On | Rx |
| | Sleep | Sleep | Off | Sleep |

The main power consumption for a sensor node contains three domains: sensing, data processing and communication. The communication depletes much more energy than the sensing and processing, so the radio transceiver is dominant power consumer in sensor node. The energy consumption for sensing device and processor can be neglected [EGH99].

Typically, there are four working modes for radios: transmitting (Tx), receiving (Rx), idle, and sleeping. The required power to idle is about the same as the power to receive. Usually, sleep mode power consumption is much less than the transmitting and receiving power consumption (one to four orders of magnitude) [MV04]. So the power consumption of radio transceiver in sleeping sensor is less than the monitor and active states.

Therefore we can assume that power depletion is negligible in the sleep state. However, the network lifetime can be effectively prolonged by K times. For example, if the lifetime of sensor network illustrated in Figure 5.1 is prolonged by three times using duplicate placement approach, the number of sensors will be increased to 18. Our algorithm placed only 14 sensors in the same field and can also prolong sensor network lifetime by three. The whole sensor network is composed of three covers (as illustrated in Figure 5.2), each of them provides complete coverage. Obviously the proposed algorithm provides an economical solution to deploy an energy efficient sensor network.

Afterwards this study discusses the possible number of covers in a sensor network. First the amount of covering grid points of sensor with a specific detection radius has been discussed and the following propositions have been obtained.

Proposition 5.1: Suppose a sensor has detection radius r , $r > 0$, $r \in Integer$, then, the number of covering grids, G_r , for the sensor in an infinite sensor field can be represented as

$$G_r = 2r + 1 + 2 \sum_{\Delta y=1}^r (2 \lfloor \sqrt{r^2 - \Delta y^2} \rfloor + 1)$$

, where r represents the detection radius of the sensor and Δy , $\Delta y > 0$, $\Delta y \in Integer$, is the distance from the sensor to one grid point in y axis, as shown in Figure 5.4.

Proposition 5.2: A grid point can be covered by a set of sensors. The maximum cardinality of the set exactly equals the number of covering grids of a sensor that is allocated in the grid point.

For example, in an infinite field, the radiuses of the sensor are 1, 2, 3, and the numbers of covering grid points are 5, 13, and 29, respectively. Moreover, a grid point can be covered by 5, 13, and 29 sensors at most.

Generally, it is impractical to use an infinite sensor field. This study focuses on the case of the rectangular sensor field with a finite area. For a finite sensor field, the upper bound of the number of covers is determined by critical grid points in the field. A critical grid point is a particular grid point which is covered by a sensor set with smaller cardinality than other sensor sets.

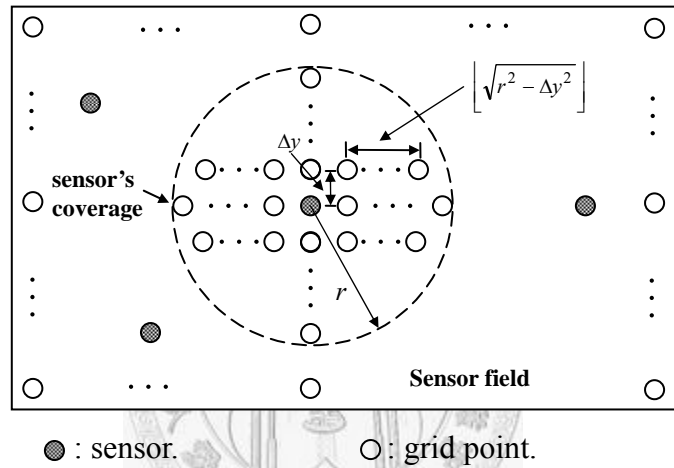


Figure 5.4: Sensor and its coverage. (The distance between any two adjacent grid points is used as one length unit.)

Proposition 5.3: On a rectangular sensor field with a finite area, the critical grid points are located at the corner of the field. Therefore, the upper bound, U_r , is

$$U_r = 2r + 1 + \sum_{\Delta y=1}^r \left\lfloor \sqrt{r^2 - \Delta y^2} \right\rfloor$$

, where r represents the detection radius of the sensor and Δy is the distance from the sensor to a grid point in y -axis.

In a sensor field with 3 by 5 service points, as illustrated in Figure 5.2, the radius of sensor is assumed to be 1, while the *critical grid points* are 1, 5, 11, and 15. According to **Proposition 5.3**, the upper bound of the number of covers, U_r , is 3. In Figure 5.2, grid point 1 can only be covered by sensors placed in grid points 1, 2, and 6. Meanwhile, the corner grid points are all covered by 3 sensors maximally. In the case, the sensor network can be partitioned into a maximum number of covers, 3 covers. Clearly, as shown in Figure 5.2, we can deploy the minimum number of sensors, 14 sensors, for the sensor network with 3 by 5 grid points.

Table 5.2: The theoretic upper bound of the number of covers in 10x10 sensor field.

| Radius (r) | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----------------|---|---|----|----|----|----|----|
| U_r | 3 | 6 | 11 | 17 | 26 | 35 | 45 |

To achieve complete discrimination, the sensor radius must be smaller than a half of the diameter of the sensor field. Therefore, we vary the radius (from 1 to 7) to compute the theoretic upper bound of the number of covers U_r in a 10 by 10 sensor field. Table 5.2 shows the theoretic upper bound of the number of covers. Theoretically, if radius is 7, we can deploy a sensor network including 45 covers such that the lifetime can be extended by 45 times.

The solution space of the problem is $O((K+1)^m)$. When field size m and the number of cover K increase gradually, the solution space increases rapidly. Hence, we develop an effective algorithm to cope with the problem.

5.3 Mathematical Model

The notations used to model the problem are listed as follows.

Given Parameters:

A : Index set of the service points in the sensor field.

B : Index set of sensors' candidate locations, $B \subseteq A$.

C : Index set of sensor cost.

K : The number of covers required for the sensor network.

a_{ij} : Indicator function which is 1 if service point i can be covered by sensor j and 0 otherwise.

c_j : Cost function of sensor j , $c_j \in C$.

Decision Variables:

x_{jk} : 1 if sensor j is allocated on cover k of the sensor network.

y_j : Sensor allocation decision variable which is 1 if sensor j is allocated in the sensor network.

Problem (IP5.1):

$$Z_{IP5.1} = \min \sum_{\forall j \in B} \sum_{\forall k \in K} c_j x_{jk} \quad (IP5.1)$$

subject to:

$$\sum_{\forall j \in B} a_{ij} x_{jk} \geq 1, \forall i \in A, k \in K \quad (5.1)$$

$$\sum_{\forall k \in K} x_{jk} \leq 1, \forall j \in B \quad (5.2)$$

$$y_j = \sum_{\forall k \in K} x_{jk}, \forall j \in B \quad (5.3)$$

$$\sum_{\forall j \in B} (a_{ij} - a_{lj})^2 y_j \geq 1, \forall i, l \in A, i \neq l \quad (5.4)$$

$$x_{jk}, y_j = 0 \text{ or } 1, \forall j \in B, k \in K \quad (5.5)$$

Physical meanings of the objective function and constraints are briefly described as follows. Problem (IP5.1) presents that the objective is to minimize the total cost of sensors. Constraint (5.1) requires that each service point must be covered in every cover of the sensor network. Constraint (5.2) and (5.3) ensure that each sensor only belongs to one cover of the sensor network. The discrimination constraint is $\sum_{\forall j \in B} (a_{ij} - a_{lj})^2 y_j \geq 1$ that requires the Hamming distance between each pair of service points in the sensor network must be greater than one. And the discrimination constraint can be rewritten as constraint (5.4). Constraint (5.5) requires integer property of the decision variables with respect to x_{jk} and y_j .



5.4 Lagrangean Relaxation Approach

5.4.1 Relaxation

This section presents the algorithm for solving the proposed sensor placement problem. An approach based upon Lagrangean relaxation is considered. Lagrangean relaxation is a method for obtaining lower bounds (for minimization problems) as well as good primal solutions in integer programming problems. A Lagrangean relaxation is obtained by identifying in the primal problem a set of complicated constraints whose removal will simplify the solution of the primal problem. Each of the complicated constraints is multiplied by a multiplier and added to the objective function. This mechanism is known as dualizing the complicating constraints [Fis81] [Fis85] [Geo74] [HWC74].

Using the Lagrangean relaxation, this investigation chooses to dualize Constraints (5.1), (5.3), and (5.4), and establishes the following Lagrangean relaxation problem.

Problem (LR5.1)

$$Z_D(u^1, u^2, u^3) = \min \left\{ \sum_{\forall j \in B} \sum_{\forall k \in K} c_j x_{jk} + \sum_{\forall i \in A} \sum_{\forall k \in K} u_{ik}^1 (1 - \sum_{\forall j \in B} a_{ij} x_{jk}) + \sum_{\forall j \in B} u_j^2 (y_j - \sum_{\forall k \in K} x_{jk}) \right. \\ \left. + \sum_{\forall i \in A} \sum_{\substack{\forall \ell \in A, \\ \ell \neq i}} u_{i\ell}^3 (1 - \sum_{\forall j \in B} (a_{ij} - a_{i\ell})^2 y_j) \right\} \quad (\text{LR5.1})$$

subject to:

$$\sum_{\forall k \in K} x_{jk} \leq 1 \quad \forall j \in B \quad (5.2)$$

$$x_{jk} = 0 \text{ or } 1 \quad \forall j \in B, k \in K \quad (5.5)$$

$$y_j = 0 \text{ or } 1 \quad \forall j \in B \quad (5.6)$$

The multipliers u^1 , u^2 , and u^3 are the vectors of $\{u_{ik}^1\}$, $\{u_j^2\}$, and $\{u_{il}^3\}$, respectively. Notably, the constraints (5.1) and (5.4) are dualized such that the corresponding multipliers u^1 and u^3 are nonnegative.

(LR5.1) can be decomposed into two independent and easily solvable subproblems, where only the x_{jk} decision variables are involved in the first subproblem and only the y_j decision variables are involved in the second subproblem. Note that, the constant terms, $\sum_{\forall i \in A} \sum_{\forall k \in K} u_{ik}^1$ and $\sum_{\forall i \in A} \sum_{\substack{\forall l \in A, \\ l \neq i}} u_{il}^3$, were

omitted from the objective function in the subproblems.

Subproblem 5.1: for x_{jk}

$$Z_{SUB5.1}(u^1, u^2) = \min \left\{ \sum_{\forall j \in B} \sum_{\forall k \in K} \left((c_j - u_j^2) - \sum_{\forall i \in A} u_{ik}^1 a_{ij} \right) x_{jk} \right\} \quad (SUB5.1)$$

subject to:

$$\sum_{\forall k \in K} x_{jk} \leq 1 \quad \forall j \in B \quad (5.2)$$

$$x_{jk} = 0 \text{ or } 1 \quad \forall j \in B, k \in K \quad (5.5)$$

Subproblem 5.2: for y_j

$$Z_{SUB5.2}(u^2, u^3) = \min \sum_{\forall j \in B} (u_j^2 - \sum_{\substack{\forall i \in A \\ \forall l \in A, \\ l \neq i}} u_{il}^3 (a_{ij} - a_{lj})) y_j \quad (SUB5.2)$$

subject to:

$$y_j = 0 \text{ or } 1 \quad \forall j \in B \quad (5.6)$$

Subproblem 5.1 comprises $|B|$ (one for each sensor) problems. To simplify descriptions of the procedures for solving Subproblem 5.1, p_{jk} is used to represent the following function:

$$p_{jk} = (c_j - u_j^2) - \sum_{\forall i \in A} u_{ik}^1 a_{ij}$$

For each sensor, first we assume sensor j is allocated and p_{jk} is calculated for each cover k . Then the minimal p_{jk} of sensor j in all cover ($\min p_{jk}$) is determined and the corresponding cover number k' can be obtained. If the minimal p_{jk} in all cover is negative, we assign $x_{jk'}$ to 1. It means that sensor j is allocated and belongs to cover k' . Otherwise, all x_{jk} are assigned to 0.

Subproblem 5.2 also comprises $|B|$ problems. Let q_j be the coefficient of y_j in (SUB5.2).

$$q_j = u_j^2 - \sum_{\forall i \in A} \sum_{\substack{\forall l \in A, \\ l \neq i}} u_{il}^3 (a_{ij} - a_{lj})^2$$

For each sensor j , if q_j is negative, we assign y_j to 1. Otherwise, let y_j be zero.

For any $(u^1, u^3) \geq 0$, using the weak Lagrangean duality theorem, the optimal objective function value of (LR5.1), $Z_{D1}(u^1, u^2, u^3)$, is a lower bound on $Z_{IP5.1}$. The dual problem then is

$$Z_{D5.1} = \max_{(u^1, u^3) \geq 0} Z_{D1}(u^1, u^2, u^3). \quad (D5.1)$$

(D5.1) is solved to find the highest lower bound. Several methods exist for

solving the dual problem (D5.1). One of the most popular methods is the subgradient method. Let a $(|A| \times |K| + |B| + |A| \times |A|)$ vector g represent a subgradient of $Z_{D5.1}(u^1, u^2, u^3)$. In iteration m of the subgradient optimization procedure, the multiplier vector π is updated by

$$\pi^{m+1} = \pi^m + t^m g^m.$$

The step size t^m is determined by

$$t^m = \frac{\lambda(Z_{IP5.1}^* - Z_D(\pi^m))}{\|g^m\|^2},$$

where $Z_{IP5.1}^*$ represents an upper bound on the primal objective function value, obtained by applying a heuristic to (IP5.1), and λ is a scalar satisfying $0 \leq \lambda \leq 2$.

5.4.2 Getting Primal Feasible Solutions

After optimally solving the Lagrangean dual problem, a set of decision variables can be found in each round. Since some of the constraints are relaxed, the solutions are infeasible for the primal problem. However, efficient heuristic algorithms must be developed to adjust the optimal dual solutions. A set of feasible solutions of the primal problem (IP5.1) then can be obtained. With increasing number of iterations, the better primal feasible solution is an upper bound (UB) of the problem (IP5.1), while the Lagrangean dual problem provides the lower bound (LB) of the problem (IP5.1).

In this section, we propose a heuristic for obtaining primal feasible solutions. The algorithm is shown as follows.

Step 1: Check constraint (5.3) for each sensor. If $y_j=1$ and

$$\sum_{\forall k \in K} x_{jk} = 0 \text{ for sensor } j, \text{ sensor } j \text{ is added to cover } k' \text{ such}$$

that $p_{ik'}$ is the minimum of p_{ik} for all covers on sensor j .

If $y_j=0$ and $\sum_{\forall k \in K} x_{jk} = 1$ for sensor j , then let y_j be one, and the sensor j is allocated.

Step 2: For each cover k of the sensor network, check coverage constraint (5.1). If the coverage constraint is violated, then *sensor-addition-procedure* are repeated, or *sensor-exchanged-procedure* are repeated until the coverage constraint is satisfied. Then, we try to *drop* sensors, which are redundant in terms of coverage constraint.

Step 3: Check the discrimination constraint (5.4) for the whole sensor network. If constraint (5.4) is violated, a lot of sensors are *added* to achieve the completely discriminated sensor network. Afterwards, this algorithm attempts to *drop* some sensors that are redundant for coverage and discrimination constraints.

5.4.3 Computational Results

To evaluate the performance of the proposed algorithm, we conduct a serial of experiments. The performance is assessed in terms of lifetime of sensor, deployment cost, and computation time.

5.4.3.1 Scenario

The proposed algorithm is coded in C under a Microsoft® Visual C++ 6.0 development environment. All the experiments are performed on a Pentium IV-1.4GHz PC running Microsoft Window XP. The algorithm was tested on a 10 by 10 sensor area. To achieve complete discrimination, the sensor radius must be smaller than a half of the diameter of the sensor area. The distance between two adjacent grid points defines the length unit. Hence, seven sets of experiments are conducted, which consider sensor radius r ranging from 1 to 7.

According to **Proposition 5.3**, each set of experiments is investigated under a given K cover, which ranges between 1 and the theoretic upper bound on maximum number of covers, U_r , $1 \leq r \leq 7$, as listed in Table 5.3. The parameters about LR algorithm include: $0 < \lambda \leq 2$, improvement counter is 35, and the amount of iteration is 1000.

5.4.3.2 Results

A. Maximum Increasing Lifetime of Sensor

We first investigate whether the theoretic upper bound of covers, U_r , can be found. Based on our assumptions, the maximum lifetime of sensor is almost proportional to the number of covers that can be found. The experimental results are listed in Table 5.3. In the first five cases, the sensor radius ranges from 1 to 5, and the proposed algorithm can always obtain the solution under the given upper bound of cover. Moreover, little difference exists between the situations where the sensor radius is 6 and 7. The degradation of the solution quality is less than 4.4%. From this perspective, the proposed algorithm is very effective for maximizing the lifetime of sensor.

Table 5.3: Comparison of U_r between the theoretical and the best found values.

| Radius (r) | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|------------------------|---|---|----|----|----|-----|-----|
| U_r (theoretic) | 3 | 6 | 11 | 17 | 26 | 35 | 45 |
| U_r (the best found) | 3 | 6 | 11 | 17 | 26 | 34 | 43 |
| Degradation (%) | 0 | 0 | 0 | 0 | 0 | 2.9 | 4.4 |

B. Deployment Cost

This study shows the best found for the minimum deployment cost by the proposed algorithm. Since all sensors have the same deployment cost, the

overall deployment cost can be simplified as the number of deployed sensors. This section instead uses a normalized term, sensor density, as a performance metric. Sensor density can be defined as follows:

$$\text{Sensor density (\%)} = \left(\frac{1}{n} \sum_{\forall j \in B} \sum_{\forall k \in K} x_{jk} \right) \times 100\%$$

Table 5.4 lists the selected results of experiments, which shows the sensor density requirements with specific sensor radius and the number of cover. For example, the sensor radius is 1 in the first experiment, and the number of covers is 1, 2 and 3. From average sensor density perspective, the average sensor density per cover is higher while the number of covers is few, as shown in Figure 5.5 and Figure 5.6. But while the cover quantity increases, the average sensor density per cover decreases progressively and achieves stability. Therefore, the proposed sensor placement algorithm is extremely effective for minimizing the sensor density increase in extending lifetime.

Table 5.4: Selected sensor densities obtained in the experiment.

| # of covers | Sensor radius | | | | | | |
|-------------|---------------|------|------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 0.40 | 0.28 | 0.25 | 0.19 | 0.22 | 0.25 | 0.25 |
| 3 | 0.8 | 0.41 | 0.29 | 0.22 | 0.22 | 0.25 | 0.25 |
| 6 | | 0.8 | 0.43 | 0.31 | 0.25 | 0.26 | 0.26 |
| 11 | | | 0.78 | 0.51 | 0.39 | 0.31 | 0.33 |
| 17 | | | | 0.79 | 0.61 | 0.43 | 0.4 |
| 26 | | | | | 0.97 | 0.65 | 0.53 |
| 34 | | | | | | 0.97 | 0.75 |
| 43 | | | | | | | 0.97 |

Moreover, from the energy efficiency and deployment cost perspectives, the proposed algorithm demonstrates a significant improvement compared with duplicate deployment approach. This study uses the required number of sensors for one cover as a base, then the times of lifetime extension and cost increase of duplicate deployment approach compare with the proposed approach's, as listed in Table 5.5. Obviously, the times of cost increase for the proposed approach is

lower than that for duplicate deployment approach. For sensor radius 7, the required number of sensors is as low as 8.4% of duplicate deployment approach.

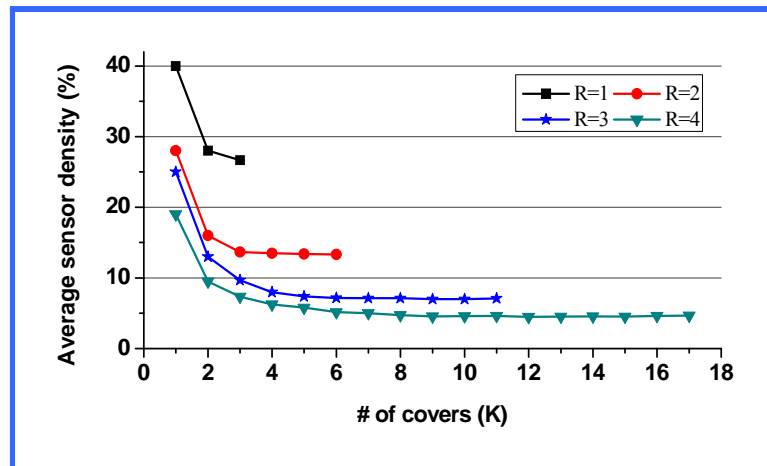


Figure 5.5: Proportion of the lifetime extending times to average sensor density per cover. (LR, R=1~4)

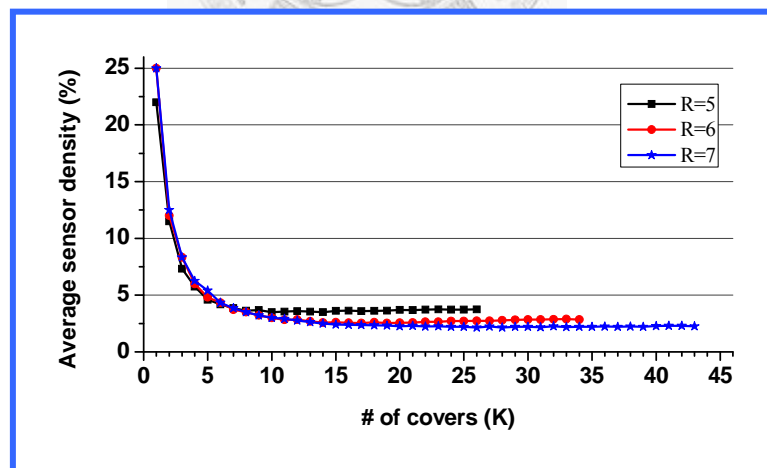


Figure 5.6: Proportion of the lifetime extending times to average sensor density per cover. (LR, R=5~7)

Table 5.5: Performance comparison between the duplicate deployment and the proposed sensor placement approach.

| Radius | The duplicate deployment | | The proposed approach | |
|--------|--------------------------|----------------|-----------------------|----------------|
| | #Duplication | Increased cost | #Cover | Increased cost |
| 1 | 3 | 3 | 3 | 2.00 |
| 2 | 6 | 6 | 6 | 2.71 |
| 3 | 11 | 11 | 11 | 3.04 |
| 4 | 17 | 17 | 17 | 4.16 |
| 5 | 26 | 26 | 26 | 4.41 |
| 6 | 34 | 34 | 34 | 3.88 |
| 7 | 43 | 43 | 43 | 3.88 |

C. Computation Time

The study observes the computation time for the proposed algorithm. Table 5.6 lists the execution time of each set of experiments. The solution time of the algorithm is below 100 seconds in all cases. The efficiency of the algorithm thus can be confirmed.

Table 5.6: The execution time of each set experiments.

| Sensor radius | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|------------------------|----|----|----|----|----|----|----|
| Solution time (Second) | 43 | 61 | 85 | 38 | 25 | 91 | 51 |

D. Density vs. Different Radius

Next, we observe the experimental results that the number of covers is one, as illustrated in Figure 5.7. The sensor radius varies from 1 to 7, and the sensor densities first decrease and then increase. In radius 4, the sensor density requirement is the lowest in all cases. It is reasonable for sensors with smaller radiuses to have smaller covered areas, and thus more sensors are required to cover the whole sensor field. Meanwhile, a larger sensor radius requires more sensors to satisfy the discrimination constraint.

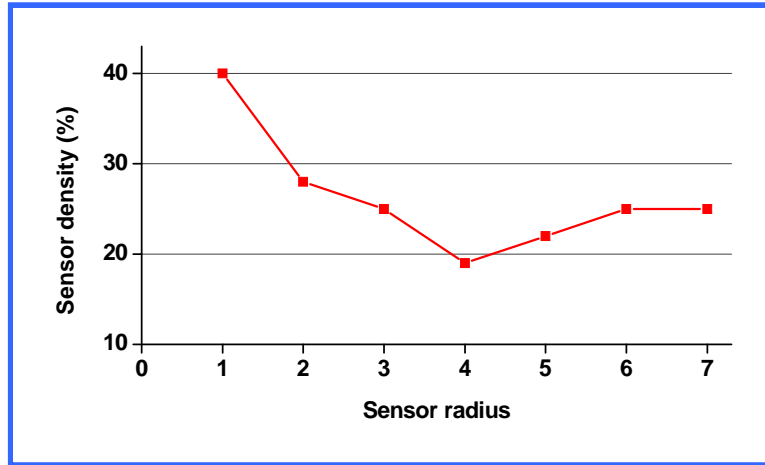


Figure 5.7: Variant of the sensor radius and the corresponding density requirement. (LR, K=1)

E. Scalability

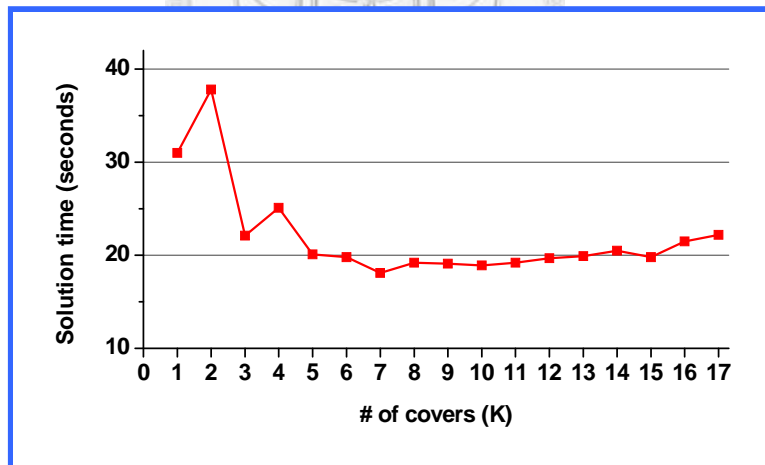


Figure 5.8: The solution time for 10x10 sensor area. (R=4)

Finally, we investigate the scalability of the proposed algorithm in terms of the solution time by two experiments. First, we evaluate the solution time under various amount of covers, K . Figure 5.8 shows the solution times for the 10 by

10 sensor field. The sensor radius is 4. In this experiment, the solution space ranges between $O(2^{100})$ and $O(18^{100})$. The results indicate that the solution times are very stable when K value increases. Actually, in all cases, the solution times are below 40 seconds.

The second experiment explores the solution time of the proposed algorithm under various sensor fields. The solution space extends from $O(2^{100})$ to $O(4^{200})$. The results show that the solution time increases very slowly, as shown in Figure 5.9, when the solution space extends greatly. The maximum solution time in this experiment is only 542 seconds. These experiments indicate the proposed algorithm has excellent solution time and highly scalable.

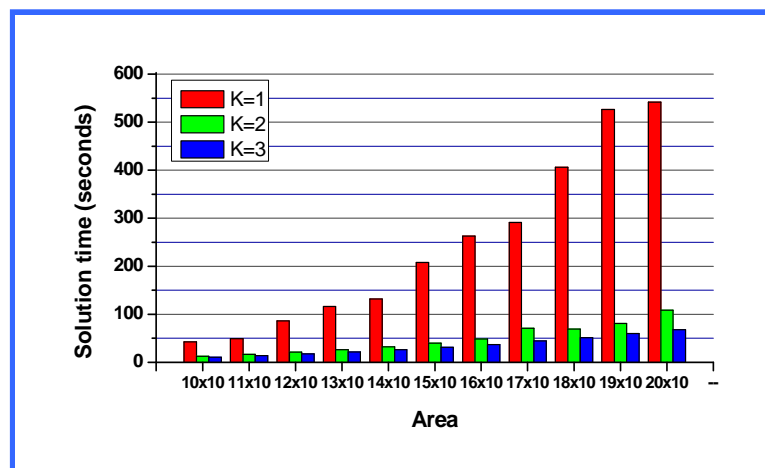


Figure 5.9: The solution time for various sensor areas. (LR, R=1)

5.5 Simulated Annealing Approach

5.5.1 Algorithm

Simulated annealing (SA) is a highly reliable method for solving hard combinatorial optimization problems [KGM83]. The concept of SA is applied to derive an efficient method for solving the problem approximately. To simplify the solution procedure, we try to relax the coverage and discrimination constraints (i.e., constraints (5.1) and (5.4)) by penalizing objective function $Z_{IP5.1}$.

The penalty for coverage constraint (constraint (5.1)) is

$$1 + p \sum_{\forall k \in K} \sum_{\forall i \in A} g_{ik}$$

, where $p, p \geq |B|$, is a constant. Variable $g_{ik}, \forall i \in A, k \in K$, indicates whether grid point i was covered by sensors in cover k . $g_{ik} = 1$, if $a_{ij}x_{jk} = 0, \forall j \in B$. Otherwise, $g_{ik} = 0$.

The penalty for constraint (4) is $1 + p^2(1 - d_{\min})$. Where d_{\min} represents the minimum Hamming distance between each pair of service points. If $\min_{\forall i, \ell \in A, i \neq \ell} \sum_{j=1}^m (a_{ij} - a_{j\ell})^2 y_j \geq 1, d_{\min} = 1$. Otherwise, $d_{\min} = 0$. Hence, energy E can be defined as

$$E = (1 + p \sum_{\forall k \in K} \sum_{\forall i \in A} g_{ik})(1 + p^2(1 - d_{\min})) \sum_{\forall j \in B} \sum_{\forall k \in K} c_j x_{jk}.$$

Table 5.7 shows a pseudo code of the algorithm. We will use the following symbols regarding any feasible solution x :

$$S_0 = \{j \mid \sum_{\forall k \in K} x_{jk} = 0, \forall j \in B\}, S_{1k} = \{j \mid x_{jk} = 1, \forall j \in B\}, \text{ and } S_1 = \bigcup_{\forall k \in K} s_{1k}.$$

Table 5.7: The SA pseudo code for sensor placement.

| | |
|-----|--|
| 1. | Let $x_{jk} \leftarrow 1, \forall j \in A$, if $(j \bmod K) + 1 = k$. |
| 2. | Calculate S_0 , S_1 , and $s_{1k}, \forall k \in K$. |
| 3. | $z_{old} \leftarrow E$. |
| 4. | Let $z_{min} \leftarrow z_{old}$ and save the current configuration as the solution. |
| 5. | $t \leftarrow t_0, b \leftarrow b_0$ |
| 6. | While $t > t_f$ do |
| 7. | Repeat b times |
| 8. | If $S_0 \neq \emptyset$ then choose $s_{new} \in S_0$ randomly. |
| 9. | Choose $s_{old} \in S_1$ randomly. |
| 10. | Choose covers c_1 and $c_2, c_2 \neq c_1$, in $[1, K]$ randomly. |
| 11. | Choose an action $ac, ac \in \{1, 2, 3\}$, randomly. |
| 12. | If $ac \neq 1$ and $S_0 \neq \emptyset$ then $S_1 \leftarrow S_1 \cup s_{new}$, $s_{1k} \mid_{k=c_2} \leftarrow s_{1k} \mid_{k=c_2} \cup s_{new}$, $S_0 \leftarrow S_0 - s_{new}$. |
| 13. | If $ac \neq 2$ and $S_1 \neq \emptyset$ then $s_{1k} \mid_{k=c_1} \leftarrow s_{1k} \mid_{k=c_1} - s_{old}$, $s_{1k} \mid_{k=c_2} \leftarrow s_{1k} \mid_{k=c_2} \cup s_{old}$. |
| 14. | $z_{new} \leftarrow E, \Delta E \leftarrow z_{new} - z_{old}$. |
| 15. | Generate random variable γ uniformly distributed in $(0, 1)$. |
| 16. | If $\Delta E \leq 0$ or $\gamma < e^{(-\Delta E/t)}$ then |
| 17. | $z_{old} \leftarrow z_{new}$ |
| 18. | If $z_{new} < z_{min}$ then $z_{min} \leftarrow z_{new}$ and save the current configuration as the solution. |
| 19. | else recover the change for S_0, S_1 , and $s_{1k}, k \in K$, that were made in steps (12) and (13). |
| 20. | End |
| 21. | $b \leftarrow b * \beta$; If $z_{old} < n$ then $t \leftarrow t * \alpha_1$ else $t \leftarrow t * \alpha_2$. |
| 22. | End. |

Initially, we assume the sensors are deployed at all grid points. In each loop, one of three actions (i.e., remove, add, and exchange) will be chosen randomly. Each action attempts to change the deployment status of one sensor. The solution with the minimum energy, z_{min} , is saved as the best found solution. While frozen temperature, t_f , is reached, the algorithm stops. If $z_{min} \leq m$, the

best found solution is a feasible solution to this problem.

The SA-based algorithm is a stochastic process. Computing time for the algorithm is relative to the selection of the cooling schedule, the desired solution quality, and the optimization problem itself. Hence, as the size of the problem increases, the time complexity of the problem will become increasingly irrelevant. In the proposed problem, the evaluation for discrimination constraint (i.e., constraint (5.4)) dominates the solution speed of the algorithm. To check whether the constraint is satisfied, we have to calculate the Hamming distance of power vectors for each pair of grid points in a sensor field. Therefore, the time complexity for the evaluation is $O(n^3)$, where n is number of grid points in the field.

5.5.2 Computational Results

To evaluate the performance of the proposed algorithm, we conduct a serial of experiments. The performance is assessed in terms of solution quality, efficiency, and scalability. We also make the same experiments by a well-known optimization software package, ILOG CPLEX 9.0. Afterward, we compare the performance of the SA algorithm with that of CPLEX.

5.5.2.1 Scenario

We assume that all sensors have the same deployment cost, the overall deployment cost can be simplified as the amount of deployed sensors. This section uses a normalized term, sensor density, as a performance metric to evaluate the solution quality of the algorithm. The average sensor density (%) is

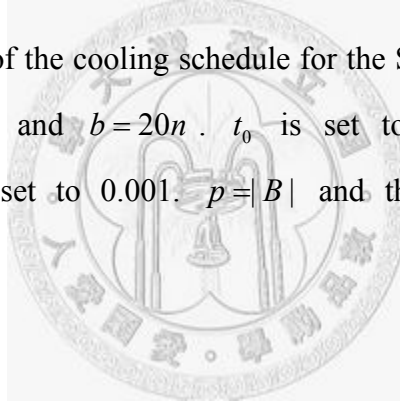
$$\text{Average sensor density (\%)} = \left(\frac{1}{Kn} \sum_{\forall j \in B} \sum_{\forall k \in K} x_{jk} \right) \times 100\%$$

. The solution time is used to evaluate efficiency of the algorithm. We observe the variation of solution quality and performance under the situation that sensor field is increased gradually to assess scalability of the SA algorithm. ILOG

CPLEX 9.0 is also used to solve the problem. Afterward, we use the solution by CPLEX to be benchmark and evaluate the performance of the SA algorithm.

We develop two sets of experiments to evaluate the performance of the proposed algorithm under various system parameters: field size, sensor radius, and number of covers. First, the algorithm was tested on a 10 by 10 service area. The sensor radius r ranges between 1 and 7. According to **Proposition 5.3**, each set of experiments is investigated under a given K cover which ranges between 1 and the theoretic upper bound on maximum number of covers, U_r , as listed in Table 5.8. The main purpose of the experiments is to examine the solution quality and efficiency of the algorithm. In second sets of experiments, the radius r is set to 1 and 7, the number of covers k is from 1 to U_r and the field size is varied from 5 by 5 to 15 by 15. The purpose of the experiment is to examine whether the algorithm has scalability while the solution space increasing.

The parameters of the cooling schedule for the SA algorithm are $\alpha_1 = 0.5$, $\alpha_2 = 0.75$, $\beta = 1.3$, and $b = 20n$. t_0 is set to $0.5G_r |B|$. The frozen temperature, t_f , is set to 0.001. $p = |B|$ and the cost of sensor $c_j = 1$, $\forall j \in B$.



5.5.2.2 Results

A. Solution Quality

We first investigate whether the theoretic upper bound of covers, U_r , can be found. Based on our assumptions, the maximum lifetime of sensor is almost proportional to the number of covers that can be found. The experimental results are listed in Table 5.8. In all cases, the SA algorithm can always obtain the solution under the theoretic upper bound of cover. From this perspective, the proposed algorithm is very effective for maximizing the network lifetime.

Table 5.8: Comparison of U_r between the theoretical and the best found values.

| Radius (r) | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|------------------------|---|---|----|----|----|----|----|
| U_r (theoretic) | 3 | 6 | 11 | 17 | 26 | 35 | 45 |
| U_r (the best found) | 3 | 6 | 11 | 17 | 26 | 35 | 45 |
| Degradation (%) | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 5.9 lists the selected results of experiments, which shows the sensor density requirements with specific sensor radius and the number of cover. Moreover, from the energy efficiency and deployment cost perspectives, the proposed algorithm demonstrates a significant improvement compared with duplicate deployment approach. This study uses the required number of sensors for one cover as a base, then the times of lifetime extension and cost increase of duplicate deployment approach compare with the proposed approach's, as listed in Table 5.10. Obviously, the times of cost increase for the SA algorithm is lower than for duplicate deployment approach. For sensor radius 7, the required number of sensors is as low as 10.58% of duplicate deployment approach.

Table 5.9: Selected sensor densities obtained in experiment.

| # of covers | Sensor radius | | | | | | |
|-------------|---------------|------|------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 0.39 | 0.23 | 0.20 | 0.16 | 0.19 | 0.20 | 0.21 |
| 3 | 0.79 | 0.37 | 0.25 | 0.19 | 0.19 | 0.20 | 0.21 |
| 6 | | 0.71 | 0.37 | 0.28 | 0.21 | 0.21 | 0.21 |
| 11 | | | 0.73 | 0.47 | 0.36 | 0.27 | 0.26 |
| 17 | | | | 0.81 | 0.60 | 0.38 | 0.35 |
| 26 | | | | | 0.97 | 0.64 | 0.52 |
| 35 | | | | | | 1.00 | 0.70 |
| 45 | | | | | | | 1.00 |

From average sensor-density perspective, the average sensor density per cover is higher while the number of covers is few, as shown in Figure 5.10 and Figure 5.11. While the cover quantity increases the average sensor-density per

cover decreases progressively and achieves stability. Therefore, the proposed sensor placement algorithm is extremely effective for minimizing the sensor density increase in extending lifetime.

Table 5.10: Performance comparison between the duplicate deployment and the SA-based sensor placement algorithm.

| Radius | The duplicate deployment | | The SA algorithm | |
|--------|--------------------------|----------------|------------------|----------------|
| | #Duplication | Increased cost | #Cover | Increased cost |
| 1 | 3 | 3 | 3 | 2.03 |
| 2 | 6 | 6 | 6 | 3.09 |
| 3 | 11 | 11 | 11 | 3.65 |
| 4 | 17 | 17 | 17 | 4.94 |
| 5 | 26 | 26 | 26 | 5.16 |
| 6 | 35 | 35 | 35 | 5.00 |
| 7 | 45 | 45 | 45 | 4.76 |

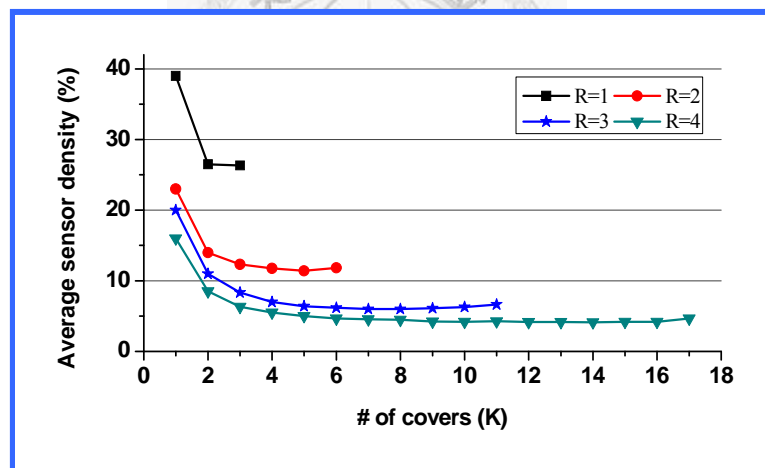


Figure 5.10: Average deployment density for 10x10 sensor field. (SA, R=1 ~ 4)

In 10 by 10 sensor field, the problem is also solved by CPLEX under different number of covers. In some case, the solution time of CPLEX is very long which can exceed several hours. So, we stop the computation of CPLEX if it exceeds 10000 seconds. Figure 5.12 to 5.15 show the sensor density solving by SA algorithm and CPLEX. The difference between densities by two

approaches is very little. In some case, SA algorithm is better than CPLEX, and vice versa. These results indicate both SA algorithm and CPLEX has the same solution quality in terms of the sensor density.

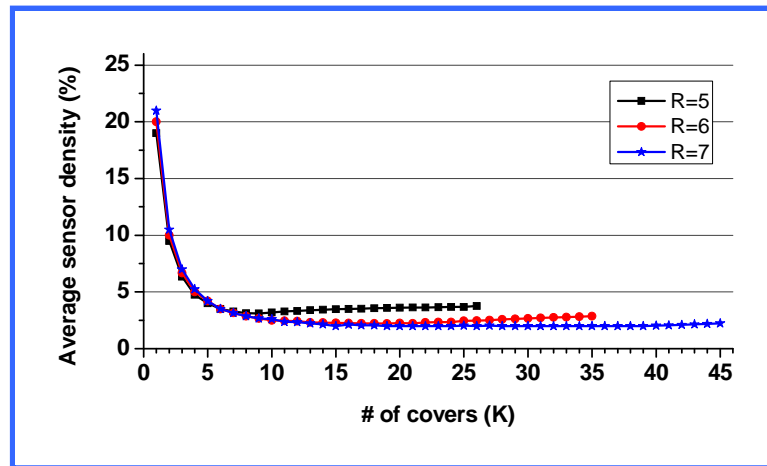


Figure 5.11: Average deployment density for 10x10 sensor field. (SA, R=5 ~ 7)

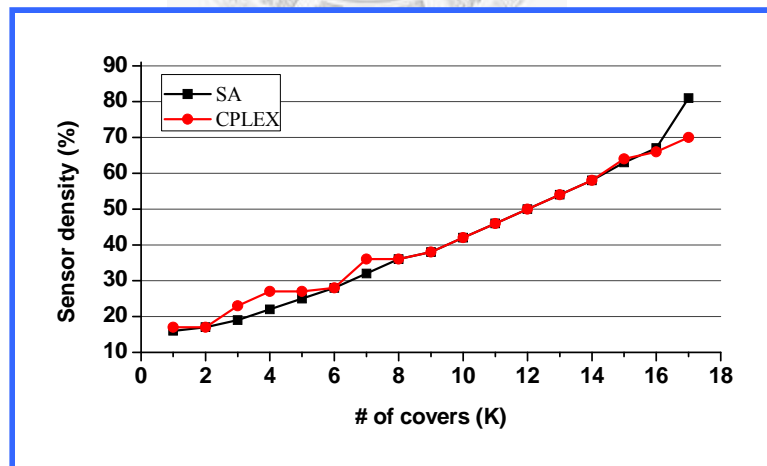


Figure 5.12: The requirement sensor density compares between the SA algorithm and CPLEX. (R=4)

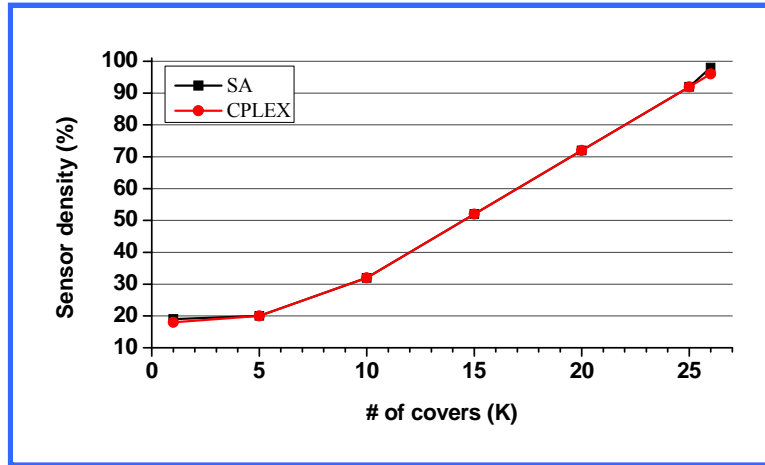


Figure 5.13: The requirement sensor density compares between the SA algorithm and CPLEX. (R=5)

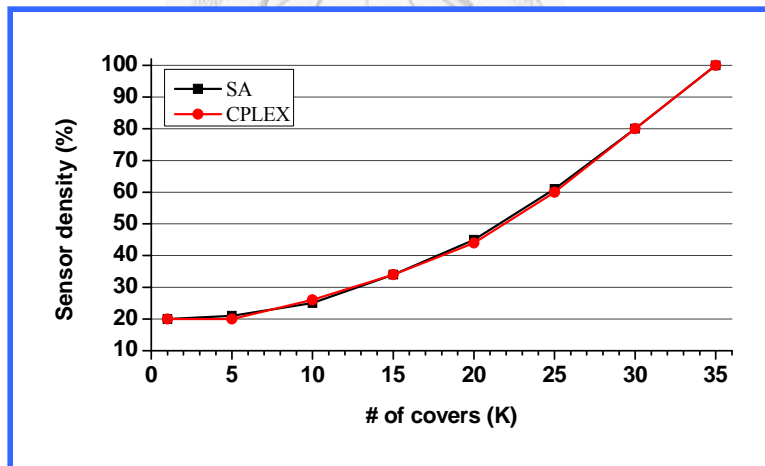


Figure 5.14: The requirement sensor density compares between the SA algorithm and CPLEX. (R=6)

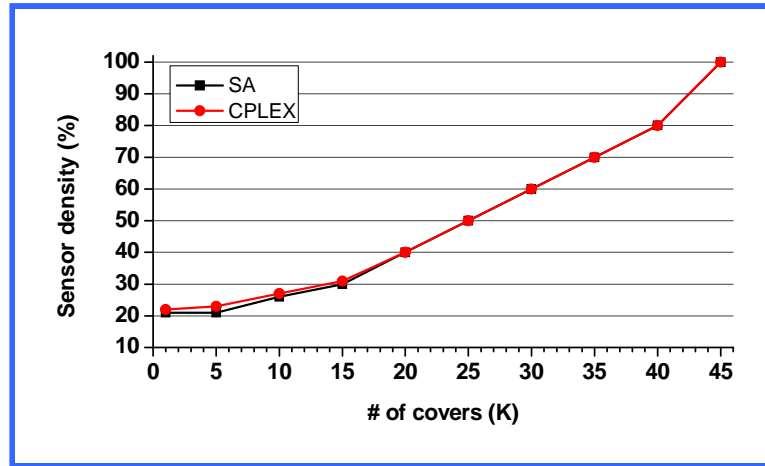


Figure 5.15: The requirement sensor density compares between the SA algorithm and CPLEX. (R=7)

B. Solution Time

We investigate the solution time of the SA algorithm with various radius and k value in a 10 by 10 field. Table 5.11 lists the minimum, maximum, and average solution times of each set of experiments, which has same radius and different k . The solution space in this experiment ranges from $O(2^{100})$ to $O(46^{100})$. The average solution time of our algorithm is only several hundreds of seconds. The maximum solution time is no more than 1600 seconds. We can also find out that the maximum difference among all cases is only about 50 times. Therefore, the proposed algorithm is efficient in terms of solution time.

Table 5.11: The solution time of the SA algorithm.

| Solution time (Seconds) | Sensor radius | | | | | | |
|----------------------------|---------------|-----|-----|-----|-----|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Average | 94 | 267 | 227 | 265 | 298 | 478 | 587 |
| Maximum | 141 | 593 | 537 | 463 | 676 | 1557 | 1437 |
| Minimum | 44 | 71 | 55 | 36 | 91 | 87 | 149 |

C. Scalability

In the second set experiments, we set sensor radius to be 1 and attend to investigate the sensor density solved by SA algorithm and CPLEX with different size of fields. Because of the computing time of CPLEX is so long, even more than several hours. We stop it while execution time exceeds 10,000 seconds and adopt the feasible solution at that moment. Figure 5.16 to Figure 5.18 show the sensor density under different field size with the number of covers, k , is 1, 2, and 3, respectively. From the sensor density perspective, the proposed algorithm is better than CPLX. The proposed algorithm always can get lower sensor density than CPLX. In addition, CPLEX can not always obtain feasible solution in 10000 seconds. Figure 5.18 shows CPLEX does not obtain feasible solutions when sensor field is 12 by 12 or 15 by 15 with K is 3.

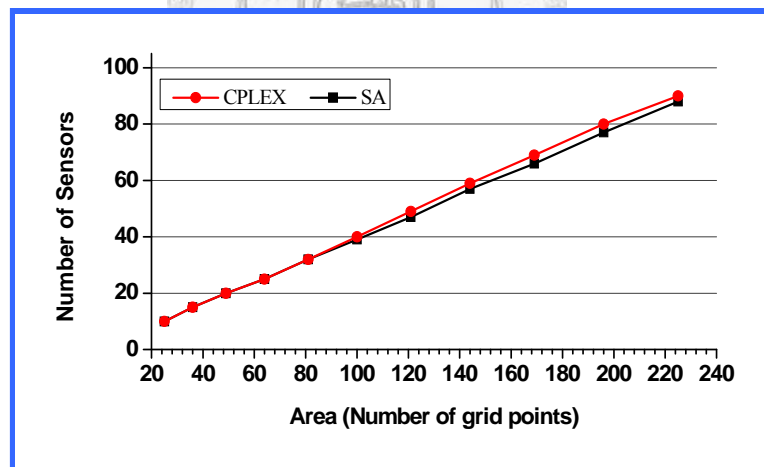


Figure 5.16: The required sensor density for various sensor fields. ($K=1$, $R=1$)

The solution time about the second set experiments is illustrated in Figure 5.19. When sensor field increases from 5×5 to 15×15 , the solution space extends from 2^{25} to 2^{225} . However, the solution time of the proposed algorithm increases

slowly when the solution space extends greatly. Contrarily, the solution time of CPLEX increases very rapidly when sensor field increases. Therefore, the proposed algorithm is highly scalable in terms of number of cover and field size for sensor density and solution time.

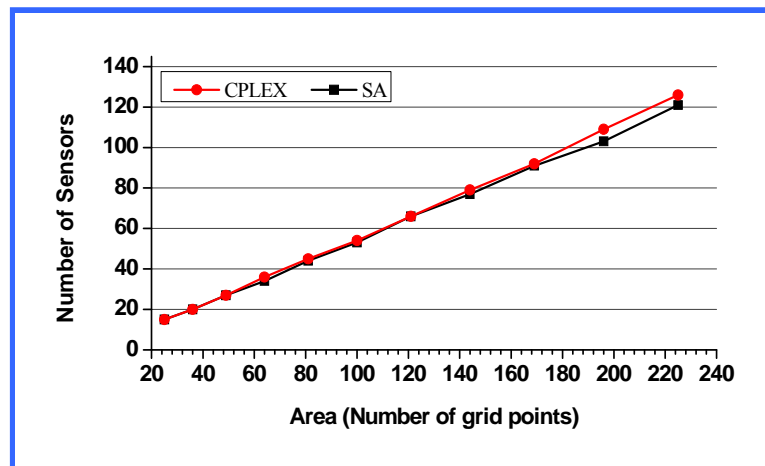


Figure 5.17: The required sensor density for various sensor fields. (K=2, R=1)

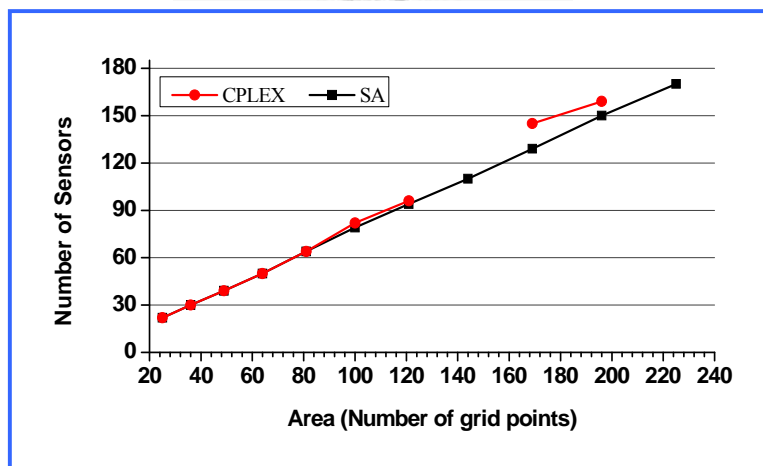


Figure 5.18: The required sensor density for various sensor fields. (K=3, R=1)

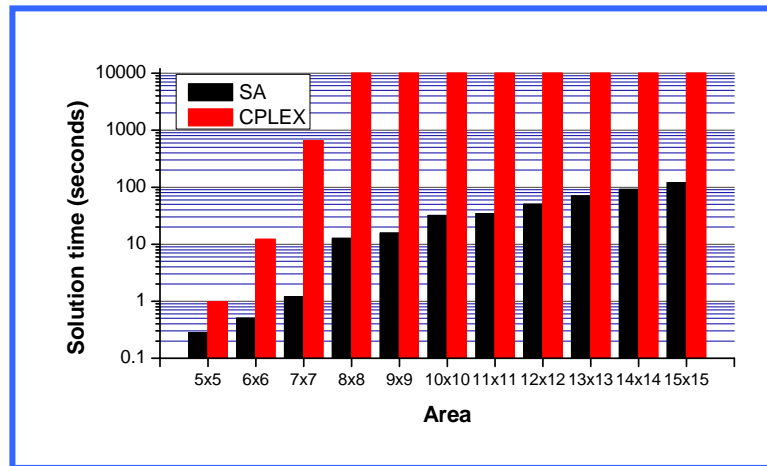


Figure 5.19: The solution time for various sensor fields (K=1).



5.6 Concluding Remarks

In this chapter, we study energy efficient sensor placement problem for surveillance and positioning service. By Lagrangean relaxation method and Simulated Annealing meta-heuristic, we develop two algorithms to address this problem. As the best of our knowledge, the proposed algorithms are truly novel as it has not been discussed in previous research.

At first, this study formulates the problem as a 0/1 integer programming problem, and then proposes a Lagrangean relaxation (LR) based heuristic and a simulated annealing (SA) algorithm for solving the optimization problem. And CPLEX also is used to solve the problem.

The proposed Simulated Annealing approach can almost prolong the working life of a sensor up to its theoretical upper bound without surveillance quality degradation. In the worst case, the proposed LR approach has 4.4% degradation.

About solution time, the proposed LR approach outperforms the SA approach. CPLEX is the worst in solution time. Both the proposed LR and SA approach are scalable in terms of solution quality and solution time, but CPLEX isn't.

The required average sensor density of one cover is effectively minimized; the maximal deployment cost is just 10.58% of that of using the duplicate sensor placement approach. Furthermore, using the same deployment density for a single-cover sensor network, we can deploy an energy-efficient sensor network such that its' lifetime can be extended up to 3 times.

The computational results indicate that the sensor placement approach is effective and the proposed algorithm is highly efficient, effective as well as scalable. Obviously, this study contributes to deploying a sensor network for target positioning with maximum lifetime.



CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1 Summary

In this dissertation, we focus on the sensor deployment problems to support environment surveillance and target positioning services from various perspectives.

First, we addressed the homogeneous sensor placement problem for environment surveillance and target positioning. The problem was formulated as a min-max mathematical optimization model. The maximum error distance was used to measure the positioning accuracy of a WSN. Then, a simulated-annealing-based algorithm has been proposed to solve the optimization problem. The experimental results reveal that the proposed algorithm not only can efficiently obtain a high-quality solution but also is effective, scalable, and robust. Afterward, we considered adjusting the sensing

radius for individual sensor in a randomly deployed sensor network and proposed a simulated annealing (SA) algorithm to cope with the problem as well. The experimental results indicate, sensors with adjustable sensing radius can actually improve the quality of positioning service when locations are given.

Next, the dissertation focuses on supporting differentiated quality of positioning service for WSNs. The work dealt with the differentiated QoS requirements for each region of interesting (ROI). The goal of the problem is either to guarantee QoS requirement of all ROIs or to minimize the QoS degradation for ROIs based on its' level of priority. We have developed three heuristics, including one Lagrangean-Relaxation-based (LR-based) algorithm and two SA-based algorithms, to address this problem. The experimental results show that when the given resource is scarce, the sensor deployment approach with differentiated QoS requirements can obtain better QoS solution than that with the uniform QoS requirement. Furthermore, for a sensor field with placement limitations, using sensors with adjustable radius can obtain higher level of QoS than adopting fixed radius sensors.

The third topic focuses on the energy efficiency issue. K independent sets of sensors (K covers) monitor a sensor field in turn and locate the target together. The problem was formulated as a mathematical optimization model, where the sensor cost was objective. We have developed a LR-based heuristic and a SA-based algorithm to solve the problem. The experimental results indicate that the proposed strategy is very effective for energy conservation. And the deployment cost is just 9% of that using the duplicate sensor deployment approach when radius and number of cover increase. Furthermore, using the same deployment density for a single-cover sensor network, we can construct an energy-efficient sensor network such that its lifetime can be extended up to at least 3 times.

The contributions of this dissertation are summarized as follows.

1. We first introduced the positioning ability as QoS parameter in WSNs designs from application perspective, also proposed the error distance to

measure the positioning ability.

2. A generic framework for sensor placement problem has been proposed to support the differentiated QoS and prioritized service in WSNs.
3. The mathematical optimization models were proposed to define the problems clearly.
4. Base on Lagrangean Relaxation and Simulated Annealing methods, we developed many heuristics to solve these optimization problems.
5. The results of this dissertation could be used as references or guidelines for sensor network builders and researchers.



6.2 Future Work

Even though we have dealt with a series of sensor deployment problems for surveillance and positioning services, there are still many open issues to be investigated worthily. We point out three challenging issues to be tackled in the future.

1. Fault Tolerant Sensor Networks Design for Surveillance and Positioning Services

We assume the survivability requirement for each ROI in sensor field is different. We will try to design the sensor network, which has to remain complete coverage, when number of sensors k_1 fail, as well as to maintain a specific positioning quality when number of sensors k_2 fail. And the redundant sensors will be used to enlarge network lifetime.

2. Scalable Sensor Networks Design for Target Location Services

The large scale sensor network design is an important and difficult problem. We will attempt to adopt the divide-and-conquer heuristic to reduce problem size and solution time. The trade off between solution time and deployment cost can be investigated.

3. Mobile Sensor Networks Design

Recently the issue about mobile sensors is attended intensively. We will assume the sensor network mixed with a lot of mobile sensors and some stationary sensors. How to optimize the coverage and quality of service of the sensor network under the moving energy constraint is one of key challenges.

We will strictly formulate the above three future issues as mathematical optimization models. The Lagrangean relaxation method and some meta-heuristics will be used to address these optimization problems.

REFERENCES

- [AGP04] Z. Abrams, A. Goel, and S. Plotkin, "Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks," in *Proc. 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, April 2004, pp. 424-432.
- [AS03] S. Adlakha and M. Srivastava, "Critical Density Thresholds for Coverage in Wireless Sensor Networks," *IEEE Wireless Communication and Networking*, vol. 3, March 2003, pp. 1615-1620.
- [ASC02a] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: a Survey," *Computer Networks*, vol. 38, March 2002, pp. 393-422.
- [ASC02b] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, August 2002, pp. 102-114.
- [BG77] M. L. Bazarra and J. J. Goode, "The Traveling Salesman Problem: A Duality Approach," *Math. Programming*, vol. 13, 1977, pp. 221-237.
- [BHE00] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low-Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications*, October 2000.
- [BKX06] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying Wireless Sensors to Achieve both Coverage and Connectivity," in *Proc. 7th International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc'06)*, 2006.
- [BP00] P. Bahl and V. N. Padmanabhan, "RADAR: an In-Building RF-based User Location and Tracking System," in *Proc. IEEE INFOCOM 2000*, vol. 2, March 2000, pp. 775-784.
- [CCK06] Y. H. Cho, S. P. Choi, W. Y. Kim, and E. C. Choi, "Development of Sensor Network Nodes for Ultrasonic Sensor-Driven Position System Inside Buildings," in *Proc. IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, vol. 2, 2006, pp. 232-236.
- [CCR01] P. Castro, P. Chiu, T. Rremenek, and R. R. Muntz, "A Probabilistic Room Location Service for Wireless Networked Environments," in *Proc. ACM UbiComp 2001*.
- [CCZ05] Y. Chen, C. N. Chuah, and Q. Zhao, "Sensor Placement for Maximizing Lifetime per Unit Cost in Wireless Sensor Networks," in *Proc. IEEE Conference on Military Communications (MILCOM2005)*, vol. 2, October 2005, pp. 1097-1102.
- [CD05] M. Cardei and D.-Z. Du, "Improving Wireless Sensor Network Lifetime through Power Aware Organization," *ACM Wireless Networks*, vol. 11, Issue 3, May 2005, pp. 333-340.

- [CHL03] I. Charon , O. Hudry, and A. Lobstein, “Minimizing the Size of an Identifying or Locating-Dominating Code in a Graph is NP-hard,” *Theoretical Computer Science*, vol. 290, 2003, pp. 2109 – 2120.
- [CHS04] W. P. Chen, J. C. Hou, and L. Sha, “Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks,” *IEEE Trans. on Mobile Computing*, vol. 3, Issue 3, July 2004, pp. 258-271.
- [CIQ01] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, “Coding Theory Framework for Target Location in Distributed Sensor Networks,” in *Proc. IEEE International Conference on Information Technology: Coding and Computing*, April 2001, pp. 130-134.
- [CIQ02] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, “Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks,” *IEEE Trans. on Computers*, vol. 51, no. 12, Dec. 2002, pp. 1148-1153.
- [CKN07] Z. Chaczko, R. Klempous, J. Nikodem, and M. Nikodem, “Methods of Sensors Localization in Wireless Sensor Networks,” in *Proc. 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07)*, 2007.
- [CL04] P. L. Chiu and Frank Y. S. Lin, “A Simulated Annealing Algorithm to Support the Sensor Placement for Target Location,” in *Proc. IEEE CCECE*, May, 2004.
- [CMC02] M. Cardei, D. Maccallum, X. Cheng, M. Min, X. Jia, D. Li, and D.-Z. Du, “Wireless Sensor Networks with Energy Efficient Organization,” *Journal of Interconnection Networks*, vol 3, No 3-4, 2002, pp. 213-229.
- [CT06] T. M. Cover and J. A. Thomas, “Elements of Information Theory,” 2nd ed., published by John Wiley and Sons, Inc., New Jersey, 2006.
- [CTL05] M. Cardei, M. T. Thai, Y. Li, and W. Wu, “Energy-Efficient Target Coverage in Wireless Sensor Networks,” in *Proc. IEEE INFOCOM 2005*, vol. 3, March 2005, pp.1976-1984.
- [CW06] M. Cardei and J. Wu, “Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks,” *Computer Communications*, vol. 29, 2006, pp. 413-420.
- [CWL05] M. Cardei, J. Wu, M. Lu, and M. O. Pervaiz, “Maximum Network Lifetime in Wireless Sensor Networks with Adjustable Sensing Ranges,” in *Proc. WiMob 2005*, vol. 3, pp. 438-445.
- [DC03] S. S. Dhillon and K. Chakrabarty, “Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks,” in *Proc. IEEE WCNC*, vol. 3, March 2003, pp. 1609-1614.
- [DCI02] S. S. Dhillon, K. Chakrabarty, and S.S. Iyengar, “Sensor Placement for Grid Coverage under Imprecise Detections,” in *Proc. 15th International Conference on Information Fusion*, vol. 2, July 2002, pp. 1581-1587.
- [DCT05] M. Ding, D. Chen, A. Thaeler, and X. Cheng, “Fault-Tolerant Target Detection in Sensor Networks,” in *Proc. IEEE Conference on Wireless Communications and Networking*, vol. 4, March 2005, pp. 2362–2368.

- [DVZ06] A. Dhawan, C. T. Vu, A. Zelikovsky, Y. Li, and S. K. Prasad, "Maximum Lifetime of Sensor Networks with Adjustable Sensing Range," in *Proc. 7th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, (SNPD 2006)*, June 2006, pp. 285 – 289.
- [EGH99] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," in *Proc. MOBICOM*, 1999, pp. 263-270.
- [Egl90] R. W. Eglese, "Simulated Annealing: A tool for Operational Research," *European Journal of Operational Research*, vol. 46, 1990.
- [Etc77] J. Etcheberry, "The Set-Covering Problem: A New Implicit Enumeration Algorithm," *Operations Res.*, vol. 25, 1977, pp. 760-772.
- [Fis73] M. L. Fisher, "Optimal Solution of Scheduling Problems Using Lagrange Multipliers: Part I," *Operations Res.*, vol. 21, pp. 1114-1127, 1973.
- [Fis76] M. L. Fisher, "A Dual Algorithm for One-Machine Scheduling Problem," *Math. Programming*, vol. 11, 1976, pp. 229-251.
- [Fis81] M. L. Fisher, "The Lagrangean Relaxation Method for Solving Integer Programming Problem," *Management Science*, vol. 27, no. 1, January 1981, pp. 1-18.
- [Fis85] M. L. Fisher, "An Applications Oriented Guide to Lagrangian Relaxation," *Interfaces*, vol. 15, no. 2, March-April 1985, pp. 10-21.
- [FS74] M. L. Fisher and J. F. Shapiro, "Constructive Duality in Integer Programming," *SIAM J. Appl. Math.*, vol. 27, pp. 31-52, 1974.
- [GCB06] D. Ganesan, R. Cristescu, and B. Beferull-Lozano, "Power-Efficient Sensor Placement and Transmission Structure for Data Gathering under Distortion Constraints," *ACM Trans. on Sensor Networks*, vol. 2, issue 2, May 2006.
- [Geo74] M. Geoffrion, "Lagrangean Relaxation and Its Uses in Integer Programming," *Math. Programming Study*, vol. 2, 1974, pp. 82-114.
- [GJ79] M. R. Garey and D. S. Johnson, "Computers and intractability: A guide to the theory of NP-Completeness," *Freeman*, New York, 1979.
- [Gof77] J. L. Goffin, "On the Convergence Rates of Subgradient Optimization Methods," *Mathematical Programming*, vol. 13, 1977, pp. 329-347.
- [GT02] M. T. Goodrich and R. Tamassia, "Algorithm Design – Foundations, Analysis, and Internet Examples," *John Wiley & Sons, Inc.*, ISBN: 0-471-38365-1, 2002, p. 611.
- [HA07] M. Hefeeda and H. Ahmadi, "Probabilistic Coverage in Wireless Sensor Networks," *Technical Report: TR 2006-21*, Simon Fraser University, Surrey, Canada, March 2007.

- [HB01] J. Hightower and G. Borriello, "A Survey and Taxonomy of Location Systems for Ubiquitous Computing," *Technical Report UW-CSE 01-08-03*, University of Washington, Computer Science and Engineering, August 24, 2001.
- [HWC74] M. Held, P. Wolfe, and H. D. Crowder, "Validation of Subgradient Optimization," *Math. Programming*, vol. 6, 1974, pp. 62-88.
- [IMP05] A. Iranli, M. Maleki, and M. Pedram, "Energy Efficient Strategies for Deployment of a Two-Level Wireless Sensor Network," in *Proc. International Symposium on Low Power Electronics and Design (ISLPED'05)*, August 2005, pp. 233 – 238.
- [KCL98] M. G. Karpovsky, K. Chakrabarty, and L. B. Levitin, "On a New Class of Codes for Identifying Vertices in Graphs," *IEEE Transactions on Information Theory*, vol. 44, no. 2, 1998, pp. 599-611.
- [KGG06] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-Optimal Sensor Placements: Maximizing Information while Minimizing Communication Cost," in *Proc. 1st International Conference on Information Processing in Sensor Networks (IPSN2006)*, April 2006, pp. 2-10.
- [KGM83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4596, May 1983, pp. 671-680.
- [KKL06] Y. G. Kim, H. K. Kim, S. G. Lee, and K. D. Lee, "Ubiquitous Home Security Robot based on Sensor Network," in *Proc. IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT '06)*, December 2006, pp. 700–704.
- [LAR01] Q. Li, J. Aslam, and D. Rus, "Hierarchical Power-aware Routing in Sensor Networks," in *Proc. DIMACS Workshop on Pervasive Networking*, May 2001.
- [LBA02] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "RAP: A Real Time Communication Architecture for Large-Scale Wireless Sensor Networks," in *Proc. IEEE RTAS 2002*.
- [LC05] Frank Y. S. Lin and P. L. Chiu, "A Near-optimal Sensor Placement Algorithm to Achieve Complete Coverage/Discrimination in Sensor Networks," *IEEE Communications Letters*, vol. 9, no. 1, January 2005.
- [LC06] M. H. Lu and T. Chen, "CMUseum: A Location-aware Wireless Video Streaming System," in *Proc. IEEE International Conference on Multimedia and Expo*, July 2006, pp. 2129 – 213.
- [LP06] L. Lazos and R. Poovendran, "HiRLoc: High-Resolution Robust Localization for Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, February 2006.
- [LRS05] M. Leoncini, G. Resta, and P. Santi, "Analysis of a Wireless Sensor Dropping Problem in Wide-Area Environmental Monitoring," in *Proc. 4th IEEE/ACM Symposium on Information Processing in Sensor Networks (IPSN'05)*, April 2005, pp. 239-245.

- [LT04] B. Liu and D. Towsley, "A Study of the Coverage of Large-scale Sensor Networks," in *Proc. IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04)*, Fort Lauderdale, Florida, October 2004, pp. 475-483.
- [Lue84] D. G. Luenberger, "Linear and Nonlinear Programming," 2nd edition, Addison-Wesley Publishing Company, 1984.
- [LXP05] S. Li, C. Xu, W. Pan, and Y. Pan, "Sensor Deployment Optimization for Detecting Maneuvering Targets," in *Proc. 7th International Conference on Information Fusion (FUSION)*, 2005, pp. 1629–1635.
- [MKP01] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage Problems in Wireless Ad-Hoc Sensor Networks," in *Proc. IEEE INFOCOM 2001*, pp. 1380–1387.
- [MP03] S. Megerian and M. Potkonjak, "Low power 0/1 coverage and scheduling techniques in sensor networks," *UCLA Technical Reports 030001*, January 2003.
- [MP05] M. Maleki and M. Pedram, "QoM and Lifetime-Constrained Random Deployment of Sensor Networks for Minimum Energy Consumption," in *Proc. 4th International Conference on Information Processing in Sensor Networks (IPSN '05)*, April 2005.
- [MRK05] V. P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff, "A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint," *IEEE Trans. on Mobile Computing*, vol. 4, no. 1, January/February, 2005, pp. 4-15.
- [MV04] M. J. Miller and N. H. Vaidya, "Minimizing Energy Consumption in Sensor Networks using a Wakeup Radio," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 4, March 2004, pp.2335–2340.
- [NKJ05] A. Nadeem, S.S. Kanhere, and S. Jha, "Probanilistic Coverage in Wireless Sensor Networks", in *Proc. IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)*, 2005.
- [NN04] R. E. Neapolitan and K. Naimipour, "Foundations of Algorithms using C++ Pseudocode," 3rd edition, Jones and Bartlett Publishers, 2004.
- [PCB00] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *Proc. ACM MOBICOM 2000*.
- [PK00] G. J. Pottie and W.J. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, vol.43, no. 5, 2000, pp. 551-558.
- [PME00] A.Porret, T. Melly, C.C. Enz, and E.A. Vittoz, "A Low-power Low-voltage Transceiver Architecture Suitable for Wireless Distributed Sensors Network," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 1, 2000, pp. 56-59.
- [QIC01] H. Qi, S. S. Iyengar, and K. Chakrabarty, "Distributed Sensor Networks-a Review of Recent Research," *Journal of the Franklin Institute*, vol. 338,

no. 6, September 2001, pp. 655-668.

- [Rap96] T. S. Rappaport, "Wireless Communications: Principles and Practice," New Jersey: Prentice Hall, 1996.
- [RSP02] V. Raghunathan, C. Schurgers, S. Park, and B. Srivastava, "Energy Aware Wireless Microsensor Networks," *IEEE Signal Processing Magazine*, pp. 40-50, March, 2002.
- [RST04] S. Ray, D. Starobinski, A. Trachtenberg, and R. Ungrangsi, "Robust Location Detection With Sensor Networks," *IEEE Journal on Selected Area in Communications*, vol. 22, no. 6, August 2004.
- [RUP03] S. Ray, R. Ungrangsi, F. D. Pellegrini, A. Trachtenberg, and D. Starobinski, "Robust Location Detection In Emergency Sensor Networks," in *Proc. IEEE INFOCOM 2003*.
- [SCI01] E. Shih, S.H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks," in *Proc. 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, July 2001, pp. 272-286.
- [SP01] S. Slijepcevic and M. Potkonjak, "Power Efficient Organization of Wireless Sensor Networks," in *Proc. IEEE International Conference on Communications (ICC'01)*, vol. 2, June 2001, pp. 472-476.
- [SR02] R.C. Shah and J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, March 2002.
- [SS05] Y. Shang and H. Shi, "Coverage and Energy Tradeoff in Density Control on Sensor Networks," in *Proc. 11th International Conference on Parallel and Distributed Systems*, vol. 1, July 2005, pp. 564-570.
- [SSS03] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," in *Proc. IEEE INFOCOM 2003*.
- [VGD06] C. T. Vu, S. Gao, W. P. Deshmukh, and Y. Li, "Distributed Energy-Efficient Scheduling Approach for K-Coverage in Sensor Networks," in *Proc. MILCOM 2006*.
- [VVB04] L. F. M. Vieira, M. A. M. Vieira, L. R. Beatriz, A. A. F. Loureiro, D. C. Silva, and A. O. Fernandes, "Efficient Incremental Sensor Network Deployment Algorithm," in *Proc. Brazilian Symposium on Computer Networks*, 2004.
- [Wan06] B. Wang, "A Survey on Coverage Problems in Wireless Sensor Networks", *Tech. Rep.*, National University of Singapore, 2006.
- [WLY06] Y. Wang, X. Liu, and J. Yin, "Requirements of Quality of Service in Wireless Sensor Network," in *Proc. International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICN/ICONS/MCL 2006)*, April 2006.

- [WWS05] B. Wang, W. Wang, V. Srinivasan, and K. C. Chua, "Information Coverage for Wireless Sensor Networks," *IEEE Communications Letters*, vol. 9, no. 11, 2005, pp. 967–969.
- [WX07] C. Wang and L. Xiao, "Sensor Localization under Limited Measurement Capabilities," *IEEE Network*, vol. 21, Issue 3, May-June 2007, pp.16 – 23.
- [WXZ03] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proc. 1st ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, November 2003.
- [WY04] J. Wu and S. Yang, "Coverage and Connectivity in Sensor Networks with Adjustable Ranges," in *Proc. International Workshop on Mobile and Wireless Networking (MWN)*, 2004.
- [Xu03] N. Xu, "A Survey of Sensor Network Applications," Survey Paper for CS694a, Computer Science Department, University of Southern California, 2003. <http://enl.usc.edu/~ningxu/papers/survey.pdf>
- [XWH05] K. Xu, Q. Wang, H. Hassanein, and G. Takahara, "Optimal Wireless Sensor Networks (WSNs) Deployment: Minimum Cost with Lifetime Constraint," in *Proc. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'05)*, August 2005, pp. 454–461.
- [YHE02] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proc. 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, June 2002.
- [YHS03] T. Yan, T. He, and J. A. Stankovic, "Differentiated Surveillance for Sensor Networks," in *Proc. 1st ACM Conference on Embedded Networked Sensor Systems (Sensys03)*, November 2003.
- [ZC03a] Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization based on Virtual Forces," in *Proc. IEEE INFOCOM 2003*, vol. 2, pp. 1293-1302.
- [ZC03b] Y. Zou and K. Chakrabarty, "Target Localization based on Energy Considerations in Distributed Sensor Networks," *Ad Hoc Networks*, no. 1, 2003, pp. 261-272.
- [ZDG04] Z. Zhou, S. Das, and H. Gupta, "Connected k-Coverage Problem in Sensor Networks," in *Proc. ICCCN 2004*.



PUBLICATIONS

(A) Journal papers:

1. F.Y.S. Lin and **P.L. Chiu**, "A Near-optimal Sensor Placement Algorithm to Achieve Complete Coverage/Discrimination in Sensor Networks," *IEEE Communications Letters*, vol. 9, no. 1, January 2005, pp. 43-45.
2. **P.L. Chiu** and F.Y.S. Lin, "Sensor Placement Algorithms Supporting Differentiated Positioning Services," (in preparation).

(B) Conference papers:

1. **P.L. Chiu** and Frank Y.S. Lin, "A Simulated Annealing Algorithm to Support the Sensor Placement for Target Location," in *Proc. IEEE CCECE 2004*, pp. 867-870.
2. F.Y.S. Lin and **P.L. Chiu**, "A Simulated Annealing Algorithm for Energy Efficient Sensor Network Design," in *3rd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, April 2005. (Acceptance ratio= $44/141=31\%$)
3. F.Y.S. Lin and **P.L. Chiu**, "Energy-Efficient Sensor Network Design Subject to Complete Coverage and Discrimination Constraints," in *Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON)*, September 2005, pp. 586 - 593. (Acceptance ratio= $55/202=27.2\%$)

APPENDIX A: SET-COVER AND SET K-COVER PROBLEMS

Set-Cover Problem

The **set covering problem** is a classical problem in computer science and complexity theory. Given several sets, its may have some elements in common. You must select a minimum number of these sets so that the sets you have picked contain all the elements that are contained in any of the sets in the input.

More formally, problem Set-Cover takes a collection of m sets S_1, S_2, \dots, S_m and an integer parameter k as input, asks whether there are is a sub-collection of k sets $S_{i_1}, S_{i_2}, \dots, S_{i_k}$, such that

$$\bigcup_{i=1}^m S_i = \bigcup_{j=1}^k S_{i_j}.$$

That is, the union of the sub-collection of k sets includes every element in the union of the original m sets [GT02].

In the set covering optimization problem, the input is a collection of m sets, and the task is to find a set covering which uses the fewest sets. The decision version of set covering is NP-complete, and the optimization version of set cover is NP-hard.

Set K-Cover Problem

The **Set K-Cover problem** is a known problem in combinatorial algorithms and is defined as follow [SP01]:

INSTANCE: Collection C of subsets of a set A , positive integer K .

QUESTION: Does C contain K disjoint covers for A , i.e. covers C_1, C_2, \dots, C_K , where $C_i \subseteq C$ such that every element of A belongs to at least one member of each of C_i ?

The Set K -Cover problem has been proved that is NP-complete problem using polynomial time transformation from the minimum cover problem [GJ79].



APPENDIX B: THEOREM A.1

From Figure 4.3, Figure 4.14, Figure 4.18, Figure 4.26, and Figure 4.27, we can observe that at the beginning and ending of all curves will trend to up, and the middle of these curves is lower than that. That means the required sensor density is high when we deploy sensors with a smaller or larger radius. In this Appendix, we will discuss the relationship between the required number of deployed sensors and sensor radius from a theoretic viewpoint. Table A.1 lists the notations used in this Appendix.

Table A.1: Lists of Notations used in Appendix B.

| Notations | Descriptions |
|----------------------------------|---|
| N_A | The number of grid points in a sensor field. |
| C_j | The number of grid points which can be covered by sensor with detection radius r_j . |
| N^{C_j} | The number of sensors with detection radius r_j that required to uniquely identify every grid points. |
| X_k | The random variable of outcome for sensor k , $X_k=1$, when sensor k can cover a given grid point, otherwise, $X_k=0$, $k \in N^{C_j}$. |
| $(X_1, X_2, \dots, X_{N^{C_j}})$ | The power code of outcome of the all sensors. |
| Y | The random variable of outcome for grid point identification, which is 0 if no grid point is identified, and it is j if j -th grid point is identified. |

First, we review backgrounds of the information theory that will be used in Theorem A.1 [CT06]. The mutual information $I(X; Y)$ is the reduction in the

uncertainty of X due to the knowledge of Y . It can be expressed as follows.

$$I(X; Y) = H(X) - H(X|Y).$$

The mutual information of a random variable with itself is the entropy of the random variable. This is the reason that entropy is sometimes referred to as *self-information*. Hence, we have the following lemma.

Lemma A.1: (*Mutual information and entropy*) [CT06]

$$I(X; Y) = H(X) - H(X|Y)$$

$$I(X; Y) = H(Y) - H(Y|X)$$

$$I(X; Y) = H(X) + H(Y) - H(X, Y)$$

$$I(X; Y) = I(Y; X)$$

$$I(X; X) = H(X).$$

□

The relationship between $H(X)$, $H(Y)$, $H(X, Y)$, $H(X|Y)$, $H(Y|X)$, and $I(X; Y)$ is expressed in a Venn diagram, as shown in . Notice that the mutual information $I(X; Y)$ corresponds to the intersection of the information in X with the information in Y .

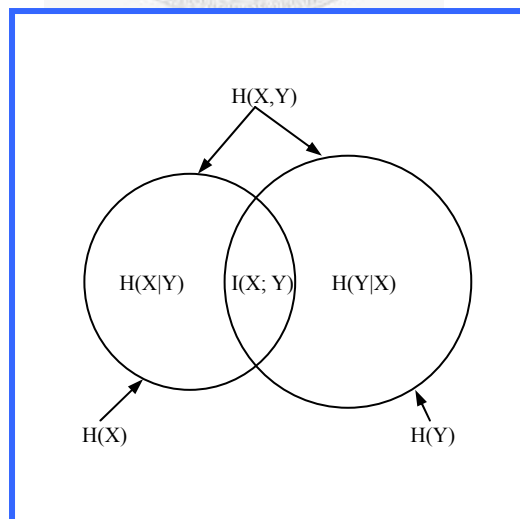


Figure A.1: Relationship between entropy and mutual information.

We now show that the entropy of a collection of random variables is the sum of the conditional entropies.

Lemma A.2: (*Chain rule for entropy*) [CT06]

Let X_1, X_2, \dots, X_n be drawn according to $p(x_1, x_2, \dots, x_n)$. Then

$$H(X_1, X_2, \dots, X_n) = \sum_{i=1}^n H(X_i | X_{i-1}, X_{i-2}, \dots, X_1). \quad \square$$

Lemma A.3: (*Conditioning reduces entropy*) (*Information can't hurt*) [CT06]

$$H(X|Y) \leq H(X)$$

with equality if and only if X and Y are independent. □

Now, the relationship between the required number of deployed sensors and sensor radius can be expressed as Theorem A.1.

Theorem A.1:

For a sensor field with N_A grid points, let $C_1 < C_2 < \dots < C_j < \dots < C_R$. If the boundary effect of sensor field is ignored, then

$$(1) \quad h\left(\frac{C_j}{N_A + 1}\right) \cdot N^{C_j} \geq \log(N_A + 1)$$

, where $h(x) = -x \log_2 x - (1-x) \log_2 (1-x)$ is the binary entropy function.

$$(2) \quad \text{Let } C_1 < C_2 < \dots < C_j \leq \frac{N_A + 1}{2} \quad \text{and} \quad \frac{N_A + 1}{2} \leq C_{j+1} < \dots < C_R, \quad \text{then}$$

$$N^{C_1} > N^{C_2} > \dots > N^{C_j} \quad \text{and} \quad N^{C_{j+1}} < N^{C_{j+2}} < \dots < N^{C_R}.$$

Proof:

Part (1):

It is a special case of [KCL98, Theorem 1.2]. We denote by X_k ($k=1, 2, \dots$,

N^{C_j}) the result of the (identification) test performed by the i -th sensor. Each X_k is a binary random variable, $X_k = \{0, 1\}$. Denote by Y the random variable which is equal to 0 when no grid point is to be identified and j ($j = 1, 2, \dots, N_A$) if the j -th grid point is to be identified, as Figure A.2 shown. We assume that all N_A+1 cases are equiprobable. Thus the entropy $H(Y) = \log_2(N_A+1)$. Figure A.3 illustrates an example of ideal identifying code for 7 grid points and 3 codewords.

| | | Outcomes of Y | | | | | |
|-------|---------------|-----------------|-----|-----|-----|-------|----------|
| | | 0 | 1 | 2 | ... | N_A | |
| X_i | X_1 | 0 | ... | ... | ... | ... | C_j 's |
| | X_2 | 0 | 0 | 1 | ... | ... | C_j 's |
| | ... | 0 | ... | ... | ... | ... | C_j 's |
| | $X_{N^{C_j}}$ | 0 | ... | ... | ... | ... | C_j 's |

Figure A.2: Random variables X_k (sensor k) and Y (grid point identification).

| | | Grid point | | | | | | | |
|--------|-------|------------|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Sensor | X_1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| | X_2 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| | X_3 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

Power vector of grid point 2 (points to the value 0 in row X_1 , column 2)

Coverage of X_3 (points to the row X_3)

Figure A.3: An example of ideal identifying code.

Now, denote the *mutual information* by $I(X_1, X_2, \dots, X_{N^{C_j}}; Y)$, which means the reduction in uncertainty of $X_1, X_2, \dots, X_{N^{C_j}}$ due to the knowledge

of Y . According to **Lemma A.1**, we denote mutual information as Equation (A.1) and illustrate in Figure A.4.

$$\begin{aligned} I(X_1, X_2, \dots, X_{N^{c_j}}; Y) \\ = H(Y) - H(Y/X_1, X_2, \dots, X_{N^{c_j}}). \end{aligned} \quad (\text{A.1})$$

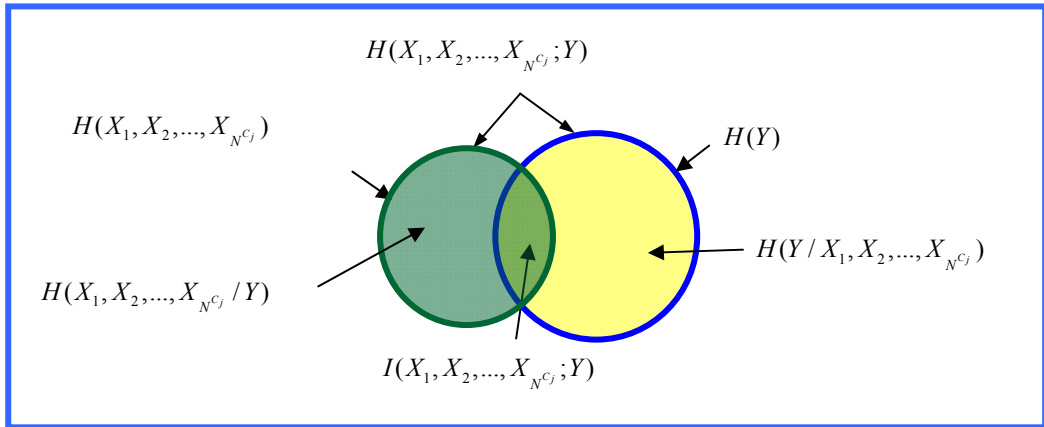


Figure A.4: Mutual information and entropy in Equation (1).

If a lot of grid points cannot be uniquely identified, then $H(Y/X_1, X_2, \dots, X_{N^{c_j}}) \neq 0$, as Figure A.5 shown. In this dissertation, the grid point in sensor field is uniquely determined by the given power code, $(X_1, X_2, \dots, X_{N^{c_j}})$, so that $H(Y/X_1, X_2, \dots, X_{N^{c_j}})$ is 0 and shows in Figure A.6. We rewrite Equation (A.1) as the follows:

$$\begin{aligned} I(X_1, X_2, \dots, X_{N^{c_j}}; Y) \\ = H(Y) - H(Y/X_1, X_2, \dots, X_{N^{c_j}}) \\ = H(Y) \\ = \log_2(N_A + 1). \end{aligned} \quad (\text{A.2})$$

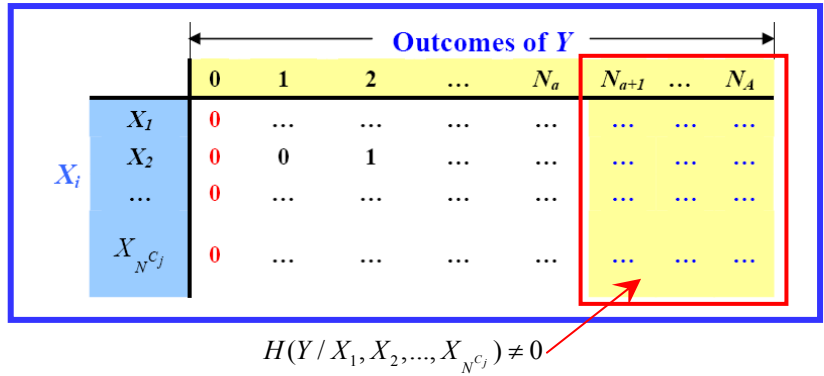


Figure A.5: A lot of grid points cannot be uniquely identified.

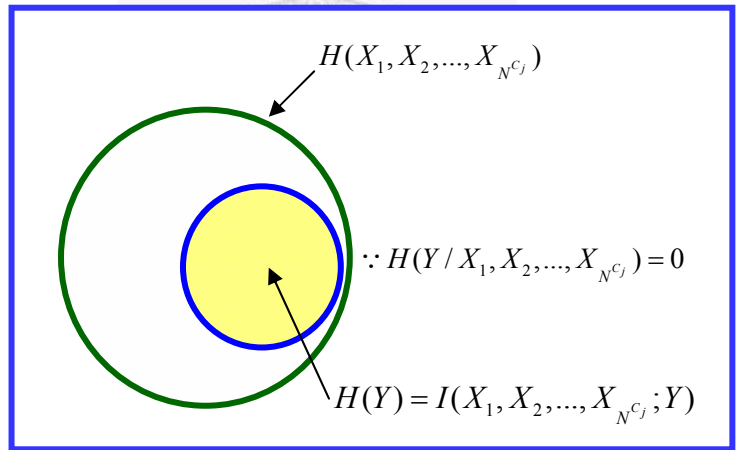


Figure A.6: Mutual Information and entropy for Equation (2).

Afterwards, due to Lemma A.1 again, we get Equation (A.3).

$$\begin{aligned}
 & I(X_1, X_2, \dots, X_{N^{c_j}}; Y) \\
 &= H(X_1, X_2, \dots, X_{N^{c_j}}) - H(X_1, X_2, \dots, X_{N^{c_j}} | Y)
 \end{aligned} \tag{A.3}$$

Because entropy is larger than or equal to zero, we can obtain Inequality

(A.4) and illustrate in Figure A.7.

$$\begin{aligned}
 & H(X_1, X_2, \dots, X_{N^{c_j}}) \\
 &= I(X_1, X_2, \dots, X_{N^{c_j}}; Y) + H(X_1, X_2, \dots, X_{N^{c_j}} | Y) \\
 &\geq I(X_1, X_2, \dots, X_{N^{c_j}}; Y) \\
 &\geq \log_2(N_A + 1)
 \end{aligned} \tag{A.4}$$

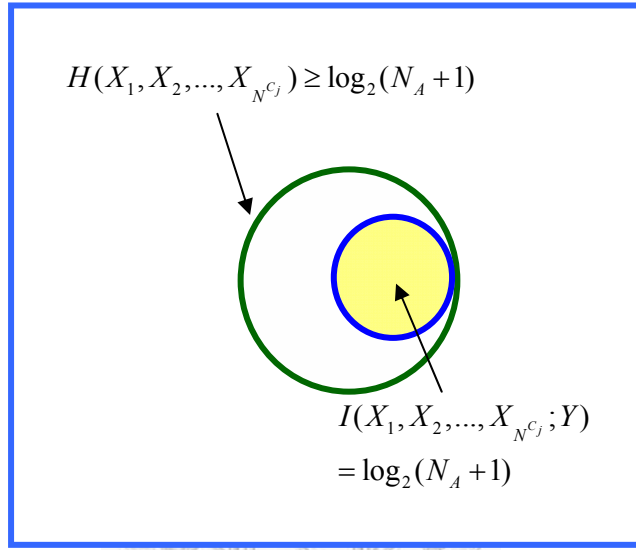


Figure A.7: Mutual information and entropy of Inequality (4).

Then based on chain rule for entropy (Lemma A.2), we can obtain Equation (A.5).

$$\begin{aligned}
 & H(X_1, X_2, \dots, X_{N^{c_j}}) \\
 &= H(X_1) + H(X_2 | X_1) + \dots + H(X_{N^{c_j}} | X_{N^{c_j}-1}, \dots, X_1) \\
 &= \sum_{k=1}^{N^{c_j}} H(X_k | X_{k+1}, \dots, X_1)
 \end{aligned} \tag{A.5}$$

From the Lemma A.3, $H(X|Y) \leq H(X)$, we obtain Inequality (A.6) and illustrates in Figure A.8.

$$\begin{aligned} & H(X_1, X_2, \dots, X_{N^{C_j}}) \\ &= \sum_{k=1}^{N^{C_j}} H(X_k | X_{k-1}, \dots, X_1) \leq \sum_{k=1}^{N^{C_j}} H(X_k) \end{aligned} \quad (\text{A.6})$$

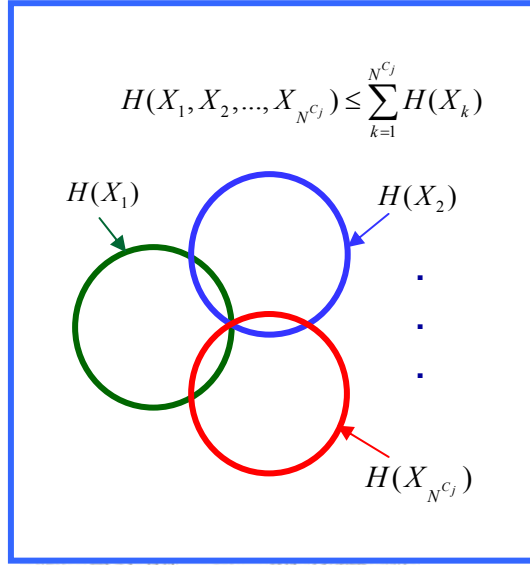


Figure A.8: The conditioning reduces entropy of Inequality (6).

However, the probability $\Pr\{X_k = 1\} = \frac{C_j}{N_A + 1}$, which is determined by the ratio of the coverage to the amount of grid points for sensor k . Hence, $H(X_k) = h\left(\frac{C_j}{N_A + 1}\right)$. Therefore, Inequality (6) can be rewritten as Inequality (A.7).

$$\begin{aligned} & H(X_1, X_2, \dots, X_{N^{C_j}}) \\ & \leq \sum_{k=1}^{N^{C_j}} H(X_k) = h\left(\frac{C_j}{N_A + 1}\right) \cdot N^{C_j} \end{aligned} \quad (\text{A.7})$$

Hence, due to Inequalities (A.4) and (A.7), we can obtain that the number of sensors N^{C_j} for identifying each grid point completely has to satisfy the following inequality.

$$h\left(\frac{C_j}{N_A+1}\right) \cdot N^{C_j} \geq \log_2(N_A+1).$$

Part (2):

When the entropy $h\left(\frac{C_j}{N_A+1}\right)$ increases, the minimum number of sensor for discrimination will decrease. The value of $\frac{C_j}{N_A+1}$ is between 0 and 1. As well as the maximum entropy approximates to 1, while $\frac{C_j}{N_A+1} \approx \frac{1}{2}$, *i.e.* the sensor coverage ratio is about 0.5 of number of grid points. In this case, the number of sensor for discrimination can be minimized. Contrarily, while $\frac{C_j}{N_A+1}$ approaches to 0 or 1, the entropy will decrease, even approach to 0. It means that the sensor coverage ratio is either smaller or larger, the entropy will decrease and the number of sensor required for discrimination will increase. \square

Example A.1: In a sensor field with 150 grid points, sensors with 1, 2, 3, 4, 5, 6, and 7 detection radii are adopted to construct a completely discriminated sensor field. The minimum required sensors for each radius are listed in . The trend of the minimum required sensors for each radius is drawn as .

Table A.2: The minimum required sensors for various radii.

| r | coverage C_j | Coverage ratio $C_j/(N_A+1)$ | Entropy of each sensor $H(C_j/N_A+1)$ | $\log_2(151)/$ $H(C_j/N_A+1)$ | Min. number of sensors min N^{C_j} |
|-----|-------------------|---------------------------------|---|----------------------------------|--|
| 1 | 5 | 0.033113 | 0.209769 | 34.50655 | 35 |
| 2 | 13 | 0.086093 | 0.423292 | 17.10026 | 18 |
| 3 | 29 | 0.192053 | 0.705746 | 10.25639 | 11 |
| 4 | 49 | 0.324503 | 0.909212 | 7.961185 | 8 |
| 5 | 81 | 0.536424 | 0.996169 | 7.266242 | 8 |
| 6 | 113 | 0.748344 | 0.813892 | 8.893569 | 9 |
| 7 | 149 | 0.986755 | 0.101609 | 71.23783 | 72 |

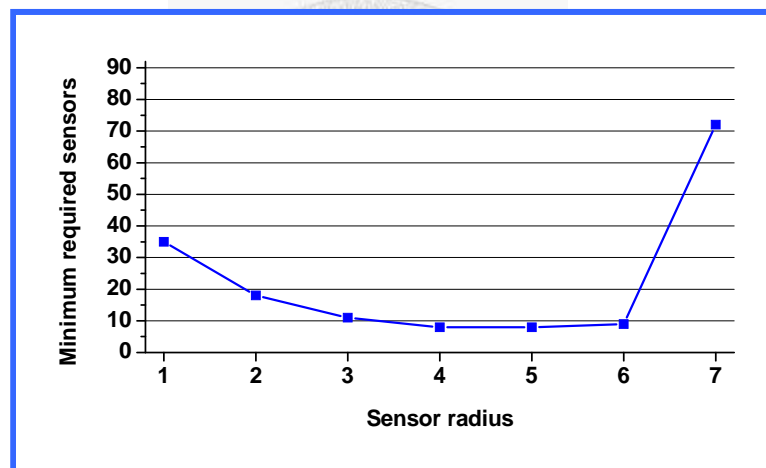


Figure A.9: The minimum required sensors vs. sensor radius in a 10x15 sensor field.

APPENDIX C:

Deploying minimum sensors to support target positioning service, why the minimum number of deploying sensors will increase while the detection radius is larger or smaller? We conduct a set of experiments to investigate this problem.

In a rectangular sensor field, the width is divided into 15 grid points and the height increases from one grid point to 15 grid points. The uniform quality of positioning service is required, and the set of candidate detection radius depends on the size of the sensor field. In Table A.3, by our algorithm (SA_2), we obtain the minimum density of sensors required to deploy a sensor network for supporting target positioning in the various detection radius and height of sensor field. The blank in Table A.3 means the radius does not be used in the sensor field with the height. From Table A.3, we can obtain the following remarks.

(1) Observing any detection radius (each column):

While the height of sensor field increases, the boundary effect for sensors moderates. As well as the trend of the sensor density requirements is descending, then moderate, and even moving up and down. In addition, the height of sensor field increases, we find the density of the sensor density is descending and even lower than the density of sensors with smaller radius. For example, when the heights are 1 to 4, the densities of sensors with radius 3 are higher than sensors with radius 2, but when the height (H) is more than 4, the densities of sensors with radius 3 are lower than sensors with radius 2.

(2) Observing the detection radius 1 to 5 (columns 1 to 5):

Although, the height of sensor field increases, the boundary effect for sensors moderates. Observing these relatively small radii, the required densities

decrease while detection radii increase. For example, the required minimum sensor density is about 38.6% for radius 1, 23% for radius 2, 18.2% for radius 3, and 13.3% for radius 5. We deduct that the required sensor density for deploying a sensor network to support target positioning will decrease when sensor's radius increase under the boundary effect being insignificant.

Table A.3: The minimum sensor density (%) for various sizes of sensor field. (The width of sensor field is 15.)

| H | Detection radius | | | | | | | | | |
|----|------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 53.33 | 60.00 | 53.33 | 60.00 | 73.33 | 86.67 | 93.33 | | | |
| 2 | 46.67 | 40.00 | 46.67 | 40.00 | 50.00 | 60.00 | 70.00 | | | |
| 3 | 40.00 | 28.89 | 44.44 | 35.56 | 44.44 | 53.33 | 62.22 | | | |
| 4 | 41.67 | 26.67 | 30.00 | 33.33 | 50.00 | 60.00 | 70.00 | | | |
| 5 | 40.00 | 25.33 | 24.00 | 24.00 | 40.00 | 48.00 | 56.00 | | | |
| 6 | 40.00 | 25.56 | 21.11 | 20.00 | 30.00 | 36.67 | 43.33 | | | |
| 7 | 40.00 | 25.71 | 19.05 | 20.00 | 20.00 | 20.00 | 23.81 | 41.90 | | |
| 8 | 39.17 | 24.17 | 19.17 | 18.33 | 18.33 | 19.17 | 19.17 | 35.00 | | |
| 9 | 39.26 | 23.70 | 19.26 | 17.78 | 16.30 | 17.04 | 18.52 | 20.74 | | |
| 10 | 39.33 | 23.33 | 20.00 | 16.00 | 15.33 | 16.67 | 17.33 | 19.33 | | |
| 11 | 39.39 | 23.64 | 20.00 | 16.36 | 15.15 | 15.15 | 16.36 | 17.58 | 19.39 | |
| 12 | 39.44 | 23.33 | 18.89 | 17.22 | 14.44 | 14.44 | 15.00 | 17.22 | 17.78 | |
| 13 | 39.49 | 23.08 | 18.97 | 16.92 | 13.33 | 13.85 | 14.36 | 15.90 | 16.41 | |
| 14 | 39.05 | 23.33 | 19.05 | 16.67 | 13.81 | 13.33 | 13.81 | 14.76 | 15.24 | |
| 15 | 38.67 | 23.11 | 18.22 | 16.44 | 14.22 | 12.89 | 12.89 | 14.67 | 15.11 | 17.33 |

(3) Observing the detection radius 6 to 10 (columns 6 to 10):

We can find the following two points which support the boundary effect is more significant for the sensors with larger detection radius.

- i. The maximum candidate radius is increasing when the height of sensor field increases. For example, the sensors with radius 8 can't

be used when the height is less than 7, the sensors with radius 9 can't be used when the height is less than 11.

- ii. In a sensor field, the detection radius is more close to the maximum candidate detection radius, the boundary effect is more significant, and the required deploying density is more.



APPENDIX D: LIST OF NOTATIONS

| Notation | Description |
|------------|---|
| A | Index set of the service points in the sensor field. |
| B | Index set of the sensor's candidate locations, $B \subseteq A$. |
| C | Index set of sensor cost. |
| D | The diameter of a sensor field. |
| $E(x)$ | The energy function of solution structure x . (for SA algorithm) |
| ΔE | The energy difference between new and current solution structures. (for SA algorithm) |
| G | Budget limitation. |
| G_r | The number of covering grids of a sensor with detection radius r . |
| K | Index of covers required for the sensor network. |
| \bar{K} | A large number. |
| N | The maximum number of sensors, $N = G/c_{min}$. |
| R | Set of candidate detection radiuses for sensor. |
| T_0 | The initial temperature. (for SA algorithm) |
| T_f | The frozen temperature. (for SA algorithm) |
| U_r | The upper bound on number of covers in a rectangular sensor field where deploys sensors with detection radius r . |
| W | Set of the discrimination weight. |
| a_{ij} | Indicator function which is 1 if service point i can be covered by sensor j and 0 otherwise. |
| c_k | The cost of sensor located at position k ; $k \in B$, $c_k \in C$. |
| c_{min} | The minimum cost of sensors. |
| d_{ij} | Euclidean distance between location i and j ; $i, j \in A$. |
| $r(T)$ | The number of iterations on temperature T . (for SA algorithm) |
| r_k | Detection radius of sensor located at k , $k \in B$. |

| Notation | Description |
|-----------|--|
| t_0 | The initial temperature. (for SA algorithm) |
| t_f | The frozen temperature. (for SA algorithm) |
| w_h | The discrimination weight for the highest-level QoS in a sensor field. |
| w_i | The discrimination weight for the i -th level QoS in a sensor field. |
| w_{ij} | Discrimination weight, $i, j \in A$, $w_{ij} \in W$. |
| x_{jk} | 1 if sensor j is allocated on cover k of the sensor network. |
| y_k | 1, if a sensor is allocated at location k and 0 otherwise, $k \in B$. |
| α | The cooling ratio, $\alpha < 1$. (for SA algorithm) |
| β | The cooling speed control parameter, $\beta > 1$. (for SA algorithm) |
| λ | A scalar for the LR-based algorithm, $0 \leq \lambda \leq 2$. |

