

Distributed Target Engagement in Large-scale Mobile Sensor Networks

by

Samaneh Hosseini Semnani

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2015

© Samaneh Hosseini Semnani 2015

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Sensor networks comprise an emerging field of study that is expected to touch many aspects of our life. Research in this area was originally motivated by military applications. Afterward sensor networks have demonstrated tremendous promise in many other applications such as infrastructure security, environment and habitat monitoring, industrial sensing, traffic control, and surveillance applications. One key challenge in large-scale sensor networks is the efficient use of the network's resources to collect information about objects in a given Volume of Interest (VOI) . Multi-sensor Multi-target tracking in surveillance applications is an example where the success of the network to track targets in a given volume of interest, efficiently and effectively, hinges significantly on the network's ability to allocate the right set of sensors to the right set of targets so as to achieve optimal performance. This task can be even more complicated if the surveillance application is such that the sensors and targets are expected to be mobile. To ensure timely tracking of targets in a given volume of interest, the surveillance sensor network needs to maintain engagement with all targets in this volume. Thus the network must be able to perform the following real-time tasks: 1) sensor-to-target allocation; 2) target tracking; 3) sensor mobility control and coordination. In this research I propose a combination of the Semi-Flocking algorithm, as a multi-target motion control and coordination approach, and a hierarchical Distributed Constraint Optimization Problem (DCOP) modelling algorithm, as an allocation approach, to tackle target engagement problem in large-scale mobile multi-target multi-sensor surveillance systems.

Sensor-to-target allocation is an NP-hard problem. Thus, for sensor networks to succeed in such application, an efficient approach that can tackle this NP-hard problem in real-time is disparately needed. This research work proposes a novel approach to tackle this issue by modelling the problem as a Hierarchical DCOP. Although DCOPs has been proven to be both general and efficient they tend to be computationally expensive, and often intractable for large-scale problems. To address this challenge, this research proposes to divide the sensor-to-target allocation problem into smaller sub-DCOPs with shared constraints, eliminating significant computational and communication costs. Furthermore, a non-binary variable modelling is presented to reduce the number of inter-agent constraints.

Target tracking and sensor mobility control and coordination are the other main challenges in these networks. Biologically inspired approaches have recently gained significant attention as a tool to address this issue. These approaches are exemplified by the two well-known algorithms, namely, the Flocking algorithm and the Anti-Flocking algorithm. Generally speaking, although these two biologically inspired algorithms have demonstrated promising performance, they expose deficiencies when it comes to their ability to maintain simultaneous reliable dynamic area coverage and target coverage. To address this challenge, Semi-Flocking, a biologically inspired algorithm that benefits from key characteristics of both the Flocking and Anti-Flocking algorithms, is proposed. The Semi-Flocking algorithm approaches the problem by assigning a small flock of sensors to each target, while at the same time leaving some sensors free to explore the environment. Also, this thesis presents an extension of the Semi-Flocking in which it is combined with a constrained clustering approach to provide better coverage over maneuverable targets. To have a reliable target tracking, another extension of Semi-Flocking algorithm is presented which is a coupled distributed estimation and motion control algorithm. In this extension the Semi-Flocking algorithm is employed for the purpose of a multi-target motion control, and Kalman-Consensus Filter (KCF) for the purpose of motion estimation. Finally, this research will show that the proposed Hierarchical DCOP algorithm can be elegantly combined with the Semi-Flocking algorithm and its extensions to create a coupled control and allocation approach. Several experimental analysis conducted in this research illustrate how the operation of the proposed algorithms outperforms other approaches in terms of incurred computational and communication costs, area coverage, target coverage for both linear and maneuverable targets, target detection time, number of undetected targets and target coverage in noise conditions sensor network. Also it is illustrated that this algorithmic combination can successfully engage multiple sensors to multiple mobile targets such that the number of uncovered targets is minimized and the sensors' mean utilization factor is maximized.

Acknowledgements

First and above all, I praise God, the almighty, for providing me this opportunity and granting me the capability to proceed successfully. Next, I would like to express my sincere gratitude to my supervisor, Dr. Otman Basir, for his encouragement and guidance throughout my research. He has provided me with infinite support through difficulties in the research and helped me to become an independent open-minded critical thinker and creative researcher.

I would like to acknowledge my advisory committee members, Dr. Peter van Beek, Dr. Mohamed Kamel and Dr. Catherine Gebotys, for their interest and advice during my PhD research program. I would like to thank my external committee member Dr. Aboelmagd Noureldin for his participation in my defense and for his invaluable comments. I am so thankful for the financial support I received during my graduate studies including the University of Waterloo Graduate Research Scholarship, Provost Doctoral Entrance Award for Women, Doctoral Thesis Writing Award and University of Waterloo Special Graduate Scholarship. I extend my special thanks to the many individuals and research groups that contributed to the experimental work during my research at the University of Waterloo. I truly acknowledge Dr. Reza Olfati-Saber and Zhaoxin Wan for spending the time to answer my technical questions. I have been fortunate to work with many brilliant people at the Pattern Analysis and Machine Intelligence (PAMI) research laboratory. My acknowledgements go to all of them for creating an excellent environment for creativity and innovation. A tribute goes to Dr. Arash Tabibiazar, Dr. Aya Sayedelahl and Dr. Alakhamis for their fruitful discussion and support. My deepest appreciation and warm thanks to Mary McPherson and Patricia Gaudet for their valuable comments to make the thesis come out in its current form. Finally, I would like to thank my friends and family for their continued support and belief in me. A special thanks to my parents, Abbas Hosseini and Farzaneh Tabrizi. They bore me, raised me, supported me, and loved me, unconditionally.

Most importantly, I would like to dedicate this thesis to my husband Ali Sheikholeslam, who never stopped supporting me and believing in my ability to pursue my ambitions. Finally I am thankful to the happiness of my life, my daughter, Zahra, for all her smiles, hugs, kisses and laughs.

Dedication

To my parents, for their never-ending love and support

To my husband, Ali, for his remarkable patience and unwavering love

To my daughter, Zahra, for making my life much happier and warmer

Table of Contents

List of Tables	xiv
List of Figures	xv
Nomenclature	xxi
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	3
1.3 Contributions	4
1.4 Organization	6
2 Background and Literature Review	7
2.1 Introduction	7
2.2 Sensor Management	9
2.3 Target Engagement	11
2.3.1 Target-to-sensor Allocation	13
2.3.2 Mobility Control and Coordination	13
2.3.3 Challenges of Target Engagement	14

2.4	State-of-the-Art in Target Engagement	15
2.4.1	Target-to-sensor Allocation Approaches	16
2.4.1.1	Neural Network-based Approaches	17
2.4.1.2	Negotiation-based Approaches	19
2.4.1.3	Classical Optimization Approaches	20
2.4.1.3.1	LP-based Approaches	20
2.4.1.3.2	DCOP-based Approaches	22
2.4.1.3.2.1	Distributed Constraint Satisfaction and Optimization	23
2.4.1.3.2.2	DCSP Definition:	23
2.4.1.3.2.3	DCOP Definition:	24
2.4.1.3.2.4	Dynamic DCOP:	25
2.4.1.3.2.5	DCSP/DCOP for Target-to-sensor Allocation	25
2.4.1.3.2.6	Sensor Network a Test-bed for DCSP/DCOP Algorithms	26
2.4.1.3.2.7	Target-to-sensor Allocation to DCSP/DCOP Modelling Approaches	28
2.4.1.4	Heuristic Optimization Approaches	30
2.4.1.5	Learning-based Approaches	31
2.4.1.6	Probabilistic Approaches	32
2.4.1.7	Organization-based Approaches	32
2.4.2	Motion Control and Coordination Approaches	34
2.4.2.1	Flocking-based Approached	35
2.4.2.1.1	Flocking Algorithm	36
2.4.2.1.2	Anti-Flocking Algorithm	40
2.5	Summary	41

3	Problem Formulation and Solution Strategy	43
3.1	Target Engagement	43
3.2	Solution Strategy	50
3.2.1	DCOP Modelling for Target-to-sensor Allocation	50
3.2.1.1	Scalability of the DCOP Model	52
3.2.1.2	Dynamicity of the DCOP Model	52
3.2.2	Flocking-based Algorithm for Control and Coordination	54
3.2.3	Airport Security as a Test Case	55
3.3	Summary	57
4	Target to Sensor Allocation: A Hierarchical Dynamic Distributed Constraint Optimization Approach	59
4.1	Introduction	60
4.2	Target-to-Sensor Allocation Problem: the Modelling	61
4.3	Hierarchical target-to-Sensor: Non-Binary Variable DCOP Representation	63
4.3.1	Hierarchical Architecture	64
4.3.2	Non-Binary Variable Modelling	66
4.3.3	Combination of Hierarchical Architecture and Non-Binary DCOP Modelling	69
4.3.4	Formal Representation of Hierarchical Non-Binary DCOP Modelling	71
4.4	Complete versus incomplete DCOP algorithms	74
4.5	Java Implementation and Results	75
4.5.1	Test Setup	75
4.5.1.1	ADOPT and DBA algorithms	77
4.5.2	Test Results	78
4.6	Summary	90

5	Semi-Flocking Algorithm for Motion Control of Mobile Sensors in Large-Scale Surveillance Systems	92
5.1	Introduction	93
5.2	Semi-Flocking Approach	94
5.2.1	Semi-Flocking Algorithm	96
5.3	Experiments and Discussion	100
5.3.1	Evaluation Parameters	100
5.3.2	Experimental Setup	101
5.3.2.1	Random Moving Targets	101
5.3.2.2	Walking Pedestrian Dataset	101
5.3.3	Simulation Results and Analysis	102
5.3.3.1	Random Moving Targets	102
5.3.3.2	Walking Pedestrian Dataset	108
5.4	Summary	110
6	Constrained Clustering for Flocking-based Tracking in Maneuverable Target Environments	111
6.1	Introduction	112
6.2	Tracking Maneuverable Targets by Flocking-based Algorithms	113
6.2.1	Constraints Definition	114
6.2.2	Constrained Clustering Approach	114
6.2.3	Constrained K-means Clustering	116
6.3	Experiments and Discussion	119
6.3.1	Evaluation Parameters	119
6.3.2	Experimental Setup	120

6.3.3	Simulation Results and Analysis	123
6.4	Summary	126
7	Reliable Collaborative Multi-target Tracking Using Semi-Flocking Sensor Networks	128
7.1	Introduction	129
7.2	Distributed Tracking of Targets in Sensor Networks Under Noisy Environment	131
7.3	Coupled Semi-Flocking and KCF Algorithm	132
7.4	Experiments and Discussion	136
7.4.1	Experimental Setup	136
7.4.2	Evaluation Parameters	138
7.4.3	Simulation Results and Analysis	139
7.5	Summary	145
8	Multi-Target Engagement in complex Mobile Surveillance Sensor Networks	146
8.1	Introduction	147
8.2	Coupled Allocation and Control Algorithm	149
8.3	Pseudo Code for Solving Complex Target Engagement Problem	154
8.4	Experiments and Discussion	158
8.4.1	Experimental Setup	158
8.4.2	Evaluation Parameters	159
8.4.3	Simulation Results and Analysis	160
8.5	Summary	164

9 Conclusion and Future Work	166
9.1 Conclusion	166
9.2 Future Work	170
References	172

List of Tables

3.1	Mapping target-to-sensor allocation problem to DCOP	50
4.1	Mapping between each part of the sensor network to a DCOP	69
4.2	Experimental setup-1	76
4.3	Experimental setup-2	76
5.1	Results of walking pedestrians dataset	109

List of Figures

2.1	Target-to-sensor allocation problem [1]	10
2.2	Communication and data management issues of mobile wireless sensor networks [2]	11
2.3	Target engagement as a twofold problem	12
2.4	Problem solving strategies for target-to-sensor allocation	17
2.5	Simple directional sensor networks [3]	22
2.6	Constraint graph and its cost function [4]	25
2.7	Flocking for n=100 agent [5]	39
2.8	Flocking algorithm drawbacks in surveillance system	39
2.9	Anti-Flocking algorithm in surveillance application	41
3.1	Sensor network of a surveillance application	44
3.2	Target engagement problem	45
3.3	Allocation and control problems	45
3.4	Solution steps for solving target engagement problem	51
3.5	The Waterloo Region International Airport	56
3.6	Waterloo Region Airport simulation environment in Java	57
4.1	Hierarchical target-to-sensor allocation model for large and complex sensor networks	65

4.2	A typical sensor network with 12 sensors and 4 targets	67
4.3	Non-binary variable DCOP modelling of the sensor network of Figure 4.2	67
4.4	Taxonomy of DCSP Algorithms [6]	74
4.5	Hierarchical target allocation model for sensor networks in a surveillance system (Waterloo regional airport)	75
4.6	Average of the number of messages required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 0.2)	79
4.7	Average of the number of messages required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 1)	79
4.8	Average of the number of messages required by ADOPT and DBA to solve target-to-sensor allocation problem in sensor network (Dynamicity level = 2)	80
4.9	Average of the number of messages required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 4)	80
4.10	Average of the number of messages required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 6)	81
4.11	Average of the number of NCCCs required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 0.2)	81
4.12	Average of the number of NCCCs required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 1)	82
4.13	Average of the number of NCCCs required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 2)	82

4.14	Average of the number of NCCCs required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 4)	83
4.15	Average of the number of NCCCs required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 6)	83
4.16	The sum of costs of found solutions in solving the target-to-sensor allocation problem in sensor network by ADOPT and DBA (Dynamicity level = 0.2)	85
4.17	The sum of costs of found solutions in solving the target-to-sensor allocation problem in sensor network by ADOPT and DBA (Dynamicity level = 1) .	85
4.18	The sum of costs of found solutions in solving the target-to-sensor allocation problem in sensor network by ADOPT and DBA (Dynamicity level = 2) .	86
4.19	The sum of costs of found solutions in solving the target-to-sensor allocation problem in sensor network by ADOPT and DBA (Dynamicity level = 4) .	86
4.20	The sum of costs of found solutions in solving the target-to-sensor allocation problem in sensor network by ADOPT and DBA (Dynamicity level = 6) .	87
4.21	The number of messages required to solve the target-to-sensor allocation problem in a static sensor network by ADOPT and DBA.	88
4.22	The number of NCCCs required to solve the target-to-sensor allocation problem in a static sensor network by ADOPT and DBA.	89
4.23	The cost of found solution in solving the target-to-sensor allocation problem in a static sensor network by ADOPT and DBA.	89
5.1	Semi-Flocking algorithm for sensor management in surveillance applications	95
5.2	Semi-Flocking navigational-control pseudo code for sensor i	97
5.3	Multi-target navigational-control in Semi-Flocking algorithm	98
5.4	Moving toward least visited area in navigational-control of Semi-Flocking algorithm	99

5.5	Sample frame from the walking pedestrian dataset	102
5.6	Snapshot of executing the Semi-Flocking algorithm by 150 sensors	103
5.7	Average percentage of Target Coverage (TC) in three Flocking-based algorithms for random moving targets	104
5.8	Average Target Detection Time (TDT) in three Flocking-based algorithms for random moving targets	106
5.9	Percentage of Non Detected Targets (PNDT) in three Flocking-based algorithms for random moving targets	106
5.10	Cumulative Area Coverage (AC) in time interval $[0, 360000)$ (ms) in three Flocking-based algorithms for random moving targets	107
5.11	Cumulative Area Coverage (AC) in time interval $[0, 360000)$ (ms) in three Flocking-based algorithms for walking pedestrian dataset	109
6.1	Mismatch of sensor speeds with maneuvering target speeds in the Semi-Flocking algorithm	113
6.2	Constrained clustering approach to match the speed of sensors with a maneuvering target	115
6.3	A maneuvering target trajectory with various levels of dynamicity ($\sigma_0 \in [0, 40)$ in four unit steps) in a time interval of $[0, 360000)$ milliseconds.	122
6.4	Average Target Coverage (TC) in Flocking and Semi-Flocking algorithms (with and without constrained clustering) for maneuvering targets.	123
6.5	Davies-Bouldin Index of constrained and simple K-means clustering in Semi-Flocking approach.	125
6.6	Number of iterations of K-means algorithm to converge when it is applied for constrained clustering of sensors in the Semi-Flocking approach.	126
7.1	Kalman-Consensus Filter [7]	133
7.2	Coupled Semi-Flocking and KCF	134

7.3	Combination of Semi-Flocking and KCF algorithms	135
7.4	Snapshot of 150 sensors executing the coupled Semi-Flocking and KCF algorithm	139
7.5	Average mean square error in tracking three targets in the estimation-control Semi-Flocking and KCF algorithm	140
7.6	Average information value of Flocking (location estimate) sensors in tracking three targets in the estimation-control Semi-Flocking and KCF algorithm	141
7.7	Average algebraic connectivity of three networks of sensors tracking three targets in the estimation-control Semi-Flocking and KCF algorithm	142
7.8	Comparison of MSE and algebraic connectivity for Target 1 within the time interval $[0, 200]$	144
7.9	Target trajectory and fused estimates of three networks of sensors tracking three targets in estimation-control Semi-Flocking and KCF algorithm	145
8.1	Target engagement problem in an airport surveillance application. Level 1) non-controllable sensors/crowded area, Level 2) controllable sensors/non-crowded area, Level 3) controllable sensors/crowded area	151
8.2	Target engagement problem as a twofold problem	152
8.3	Example of target engagement using Semi-Flocking and hierarchical DCOP modeling approaches, a) before applying control and allocation algorithms, b) after applying the Semi-Flocking algorithm, c) after applying the hierarchical DCOP modeling (assignment of 10 sensors to each target)	153
8.4	Pseudo code for solving complex target engagement problem	155
8.5	Snapshot of executing coupled Semi-Flocking and hierarchical DCOP modelling approach by 15 targets and 98 sensors	161
8.6	Number of violated constraints (conflicts) in the coupled Semi-Flocking and hierarchical DCOP modelling algorithm	162

8.7	Average of the utilization factor over all of the sensors in the coupled Semi-Flocking and hierarchical DCOP modelling algorithm	163
8.8	Figure 11. Variance of the utilization factor over all of the sensors in the coupled Semi-Flocking and hierarchical DCOP modelling algorithm	164
9.1	An Overview of the proposed target-engagement solution steps.	168

Nomenclature

ABT Asynchronous Backtracking

AC Cumulative Area Coverage

ACO Ant Colony Optimization

AOI Area of Interest

APO Asynchronous Partial Overlay

AWC Asynchronous Weak Commitment

CSP Constraint Satisfaction Problem

DBI Davies-Bouldin Index

DCOP Distributed Constraint Optimization Problem

DCSP Distributed Constraint Satisfaction Problem

DDM Distributed Dispatcher Manager

DFS Depth-First Search

DynDCOP Dynamic Distributed Constraint Optimization Problem

FOV Field of View

GA Genetic Algorithm

HDCOP Hierarchical Distributed Constraint Optimization Problem

ILP Integer Linear programming

KCF Kalman-Consensus Filter

LP Linear programming

MAS Multi-Agent Systems

MEM Multiple Elastic neural network Modules

MSE Mean Square Error

NCCCs Number of Non-concurrent Constraint Checks

NITC Number of Iterations to Convergence

NP Non deterministic Polynomial

PDA Probabilistic Data Association

PNDT Percentage of Non-detected Targets

PSO Particle Swarm Optimization

SA Simulated Annealing

SHM Structural Health Monitoring

SOM Self Organizing Map

STAV Sensor-Target as Variable

TAV Target as Variable

TC Target Coverage

TDT Target Detection Time

UAV Unmanned Air Vehicles

VOI Volume of Interest

WSN Wireless Sensor Network

Chapter 1

Introduction

1.1 Motivation

The importance of sensor networks is highlighted by their wide range of applications in various domains, including but not limited to military applications [8], infrastructure security [9], environment and habitat monitoring [10], industrial sensing [11], traffic control, and surveillance applications [12]. A sensor network is a collection of sensors, where each sensor is of small size and embedded with data processing and communication capabilities, albeit with limited communication and computation abilities. The main purpose of a sensor network is to sense, collect, and process the information in its area collaboratively [13]. A sensor network may combine stationary and mobile sensors in order to deal with the dynamics in its environment. Today, millions of static and mobile sensors are installed throughout the world. Sensor networks can replace humans in monitoring insecure, large or inaccessible areas applications, thereby making human life easier. Recent years have witnessed exponential growth in this field, and this trend is expected to intensify as sensor networks advance in capabilities.

Sensor networks have demonstrated noticeable success in mobile surveillance applications [14, 15, 16, 17], showing advanced capabilities to self-organize, and to cooperate and coordinate their activities to collect information about targets and events in a given volume of interest (VOI). The information collected by the sensors is often fused to obtain

a complete picture of the environment and assess situations of interest. Although sensor networks have been successfully applied in many surveillance systems, several challenges still confront this category of research, especially in large-scale networks.

Large-scale sensor networks require intelligent management schemes to control their large numbers of sensor nodes and process the data that these sensors collect. Finding a suitable target engagement that optimizes the use of available resources is an important area of research in this field. The inability to provide a suitable engagement not only leads to serious collisions between sensors but also leads to missing some targets and eventually to missing portions of the information in the VOI. The target engagement refers to the timely optimal assignment of a set of sensors to monitor a set of targets, applying suitable allocation, control and coordination mechanisms. This problem is a general and twofold problem concerned with two main real-time sub-problems: target-to-sensor allocation and sensor control and coordination. The goals of these two sub-problems are related and interconnected.

There is a well-recognized need for a general and efficient approach to deal with the target-to-sensor allocation sub-problem; most current solutions work only in restricted situations and are not general enough to be adopted by diverse applications. Furthermore, existing approaches suffer from unrealistic simplifications of the original problem in terms of complexity and scale; typically, they consider only a small number of sensors and/or targets, and assume sensors to be static. Quite often, even the targets are assumed to be static. As a result, it is important to develop and implement effective strategies for optimal engagement schemes that ensure effective sensors-to-targets assignments and so guarantee continuous and timely information gathering in the VOI. Also, due to communication and energy restrictions, centralized management algorithms are not efficient and there is a need for distributed algorithms that restrict the communication between neighbours [18].

On the other hand, the ability to self-organize constitutes an indispensable attribute in surveillance applications where target mobility increases surveillance complexity. In this case, sensor mobility comes in handy to enable the network so as to set-up achieves dynamic area coverage and reliable target detection. An important challenge in self-organizing surveillance systems is the control and coordination of sensor mobility. This problem concerns the optimal movement of a set of mobile sensors to achieve maximum area and/or

target coverage [19], maximum radio coverage between the sensors [20, 21], or improved target coverage over maneuverable targets [22], etc. This research addresses the issue of sensor control and coordination for maximum area and target coverage.

1.2 Objectives

This research aspires to propose an efficient, general, and scalable multi-target engagement strategy for complex and large-scale sensor networks. This engagement framework is sought as the main objective of this research, which will address the following issues:

- **Simultaneous reliable dynamic area and target coverage:** Providing simultaneous coverage over all the targets in the VOI and also acceptable coverage over the Area of Interest (AOI) to detect new targets are two inherently conflicting objectives that are the main requirements in many dynamic surveillance applications.
- **Target-to-sensor allocation:** This selection process, which involves optimal assignment of targets to a set of sensors, is an NP-hard problem. Thus, for sensor networks to succeed in such applications, an efficient approach that can tackle this NP-hard problem in real-time is desperately needed.
- **Sensor mobility control and coordination:** Optimal movement of mobile sensors in such a way as to achieve a certain objective is one of the essential requirements of any sensor-management framework that has mobile and controllable sensors. An appropriate mobility control mechanism will greatly enhance the quality of the final solution.
- **Tracking both manoeuvring and non-manoevring targets:** The need to track manoeuvring targets in addition to non-manoevring ones has been recognized in the majority of the real-world tracking scenarios. Human motion tracking or military/civilian surveillance is an example of these applications in which the targets often move with high manoeuvring ability, and require more-advanced tracking approaches.

- **Reliable Target Tracking:** An important concern in tracking multiple targets in sensor networks is the ability of sensors to track targets in noisy measurement conditions, which is the case in many real-world scenarios. The success of a network to track targets in such environments hinges significantly on the sensors' ability to reach a consensus value on their measurement of targets' status. This value must have the minimum error possible.

1.3 Contributions

This work proposes an intelligent and distributed target engagement mechanism for multi-target and large-scale sensor networks in a surveillance context. The proposed approach controls and coordinates sensors to ensure simultaneous dynamic area and reliable target coverage. It also optimizes allocation between sensors and targets such that the maximum number of targets is covered using available sensors. The main contributions of this work are as follows:

- **Formulation of the target engagement problem:** This work presents a formal formulation of the target engagement problem as an optimization problem subject to some constraints. The distributed nature of the problem and also the mobility of both sensors and targets are considered in this formulation.
- **Hierarchical DCOP modelling:** This work proposes a technique to solve the target-to-sensor allocation problem by modelling the problem as a hierarchical Distributed Constraint Optimization Problem (DCOP). DCOPs tend to be computationally expensive and often intractable, particularly in large problem spaces such as Wireless Sensor Networks (WSNs) . To address this challenge, I propose changing the sensor-to-target allocation to a hierarchical set of smaller DCOPs with a shared system of constraints, avoiding significant computational and communication costs.
- **Non-binary variable DCOP modelling:** In contrast to other DCOP modelling methods, this research presents a non-binary variable model for reducing the number

of variables and the number of intra-agent constraints, and consequently reducing the communication cost.

- **Semi-Flocking algorithm:** This work presents Semi-Flocking, a biologically inspired algorithm that benefits from key characteristics of reported Flocking-based algorithms. The Semi-Flocking algorithm approaches the control and coordination problem in sensor networks by assigning a small flock of sensors to each target, while at the same time leaving some sensors free to explore the environment. This approach allows the algorithm to strike a balance between reliable area coverage and target coverage. This balance is facilitated via flock-sensor coordination.
- **Constrained clustering for tracking manoeuvrable targets:** This research presents a constrained clustering approach to be combined with Flocking-based algorithms to provide better coverage over manoeuvrable targets. To perform the constrained clustering, a novel extension of K-means algorithm is presented and applied to cluster the sensors. This extension clusters the sensors based on certain background knowledge. Then the information about the clusters is used to improve coverage over manoeuvrable targets.
- **KCF augmented Semi-Flocking algorithm:** Addressing the problem of robust multiple target tracking using a sensor network requires a coupled distributed estimation and motion-control approach. This work proposes a framework wherein the Semi-Flocking algorithm is employed for the purpose of multi-target motion control, and a Kalman-Consensus Filter (KCF) for the purpose of motion estimation. In the proposed coupled approach, each small group of Flocking sensors (semi-flock) applies a separate KCF algorithm to estimate the position of its target. Doing so allows sensors to collectively provide reliable target engagement and comprehensive area coverage.
- **Coupled allocation-control algorithm:** To ensure timely tracking of mobile targets, the surveillance sensor network needs to maintain continuous engagement with all targets. Thus, the network must be able to perform the following real-time tasks: 1) target-to-sensor allocation; 2) sensor mobility control and coordination. This work

presents a combination of the Semi-Flocking algorithm, as a multi-target motion control and coordination approach, and a hierarchical DCOP modelling algorithm, as an allocation approach, to tackle target-engagement problems in mobile multi-target multi-sensor surveillance systems.

This thesis maintains that by the use of a continuous intelligent target engagement mechanisms over a distributed large-scale sensor network, it is possible to achieve simultaneous dynamic area coverage and reliable target coverage, in addition to optimizing the allocation of the sensors to targets under mission-critical constraints.

1.4 Organization

This thesis is organized as follows: Chapter 2 provides a comprehensive background of the target engagement problem in large-scale sensor networks from the perspective of a surveillance system. Chapter 3 presents a formal formulation for the target engagement problem and lays the foundation for the solution strategy. A hierarchical dynamic distributed constraint optimization approach is devised in Chapter 4 to solve the target-to-sensor allocation problem. Chapter 5 presents the Semi-Flocking algorithm for motion control of mobile sensors in large-scale surveillance systems, describing its design details and evaluating its performance. In Chapter 6, a constrained clustering approach is combined with the Flocking and Semi-Flocking algorithms to make them more powerful in tracking manoeuvrable targets. Chapter 7 introduces a KCF-augmented Semi-Flocking algorithm for tracking multiple targets under a wide range of target dynamics in a noisy sensor network. A combination of the Hierarchical non-binary DCOP modelling approach and Semi-Flocking algorithm is presented in Chapter 8 to ensure a complete solution for the target engagement problem. Finally, the conclusion and future work is presented are Chapter 9.

Chapter 2

Background and Literature Review

This chapter provides an introduction to large-scale sensor networks and discusses the need and challenges of sensor management in such systems. It also reviews background research on sensor management focusing on the target engagement problem and summaries previous research work conducted to address this problem. A brief introduction of large-scale sensor networks and their applications is presented in Section 2.1. In Section 2.2, sensor management, its properties and challenges are discussed. Section 2.3 will introduce target engagement and discuss the main challenges in this problem. As target engagement is a twofold problem, this section will introduce briefly target-to-sensor allocation and mobility control and coordination as two key issues that need to be addressed in order to develop effective and efficient target engagement performance. Section 2.4 discusses techniques reported in the sensor management literature to address the target engagement problem focusing on those solutions that apply the DCOP/DCSP modelling tool for target-to-sensor allocation and Flocking-based approaches for mobility control and coordination. Finally, Section 2.5 concludes the chapter.

2.1 Introduction

Sensor Networks are widely considered as one of the most significant technologies in the current century that has the potential to change our way of living [23]. Recent years have

witnessed exponential growth in this field, from both academia and industry perspective and this trend is expected to intensify as sensor networks continue to advance in capabilities. A large-scale sensor network typically consists of a large number of sensors where each sensor is of small size and embedded with data processing and communication capabilities, albeit with limited communication and computation abilities. The main purpose of a sensor network is to sense, collect, and process the information in its area collaboratively [13]. A sensor network may combine stationary and mobile sensors in order to deal with the dynamics in its environment.

The sensors in a sensor network communicate and cooperate with one another to accomplish a common task. This common objective can be environment monitoring, military surveillance, habitat tracking, security applications or industrial process controls just to name a few. The main specific characteristic of sensor networks is that, while each sensor has low computation and communication abilities, the aggregate power of the entire network is sufficient for the required mission [23].

Large-scale sensor networks are faced with various challenges. Energy consumption, data compression, self-organizing, routing, quality of service, security and energy harvesting are just a few examples of sensor networks challenges where many researchers have focused their investigations for the past twenty years [24]. One key challenge in large-scale sensor networks is the efficient use of the network's resources to collect information about objects in a given volume of interest. To address this challenge sensor networks employ resource management schemes that aim for maximum performance with minimum cost. This very conflicting multi-criteria nature of resource management in sensor networks makes the problem difficult to tackle, particularly in mission-critical applications. Multi-sensor Multi-target tracking in surveillance applications is an example where the success of the network to track targets in a given volume of interest, efficiently and effectively, hinges significantly on the network's ability to allocate the right set of sensors to the right set of targets so as to achieve optimal performance. This task can be even more complicated if the surveillance application is such that the sensors and targets are expected to be mobile. Small mobile sensors that move in space over time can carry information between isolated parts, and can also disperse the energy consumption of sensors to deliver data [2, 25].

2.2 Sensor Management

Sensor management is an important and challenging topic in sensor networks. It can be described formally as “a system or process that seeks to manage or coordinate the usage of a suite of sensors or measurement devices in a dynamic, uncertain environment, to improve the performance of data fusion and ultimately that of perception” [26]. This topic comprises a set of challenging interwoven issues. Resource allocation, coordination, and scheduling are three key issues in the development of any sensor management system:

- **Resource allocation** is a process concerned with the assignment of network resources to the set of elements occurring in the sensor network environment. Since sensors are considered the most pivotal ingredients of the network, sensor allocation tends to be of utmost importance for effective and efficient sensing performance. The process is often formulated as a combinatorial optimization problem. The formulation stipulates an objective function(s), a set of network sensors, a set of tasks (targets); the goal is to find an assignment of sensors to tasks/targets that maximizes (minimizes) the objective function subject to a set of constraints. This problem in a sensor network context is concerned with the allocation of a number of sensors to a given number of activities or objects in order to achieve the sensing task most effectively and efficiently [27, 28]. Considering the network sensors as restricted resources, and the targets as objects, the target-to-sensor allocation problem can be considered a problem of resource allocation in sensor networks.

Another allocation problem in sensor management is target-to-sensor allocation, which aims to find the best assignment of available sensors to targets subject to a set of constraints. These constraints tend to be complex to set and hard to satisfy due to the intricate relationship between the sensors and that between them and their working environment. Target-to-sensor allocation includes stationary and mobile sensors, each controlled as an independent agent. I also includes targets moving through the sensors sensing range. All sensors must act as a team to cooperatively track the maximum number of targets. Tracking a target accurately requires k sensors [1].

Figure 2.1 shows an example of this problem [1]. The task of tracking Target T_1

requires three overlapping sensors: sensors A_1 , A_2 , A_3 and A_4 . To track the target effectively these three sensors should coordinate their tracking effort. Each sensor can detect the distance and speed of the target being tracked. The problem becomes more complicated in multi-target environments in which one sensor may have to select one of various targets in its field of view. For example, in Figure 2.1, Sensor A_4 must decide between Target $Target_1$ or Target $Target_2$. In addition, the dynamicity of the environment makes the problem even more difficult and quite interesting. What would have been an optimal allocation may be infeasible to satisfy as targets move around.

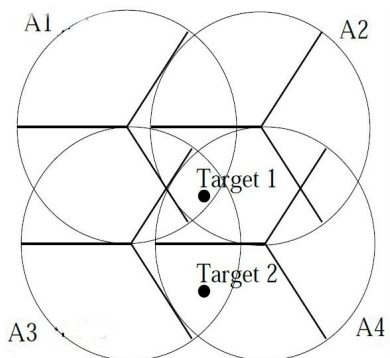


Figure 2.1: Target-to-sensor allocation problem [1]

- **Coordination** is concerned with the management of the sensors to ensure proper execution of their sensing task, avoiding conflicts and leveraging existing synergy between them. Conflict between sensors is resolved based on a priority scheme on the set of tasks being executed or targets being pursued [29]. Information exchange between the network sensors enables them to maximize synergy in executing their sensing tasks.
- **Scheduling** is an important process for the network to achieve timely and proper completion of its sensing tasks. Furthermore, it is quite an essential part of sensor coordination. The process involves the allocation of time segments to specific tasks or activities. Each task or activity starts at a specific time, continues for a fixed time interval, and terminates at a specific time [29].

The authors in [2] present a survey on sensor management issues in mobile sensor networks, focusing on communication and data management. As illustrated in Figure 2.2, topology control, coverage, localization, and target tracking are discussed as the main issues in communication, and the data gathering and replication methods with respect to data management are analyzed.

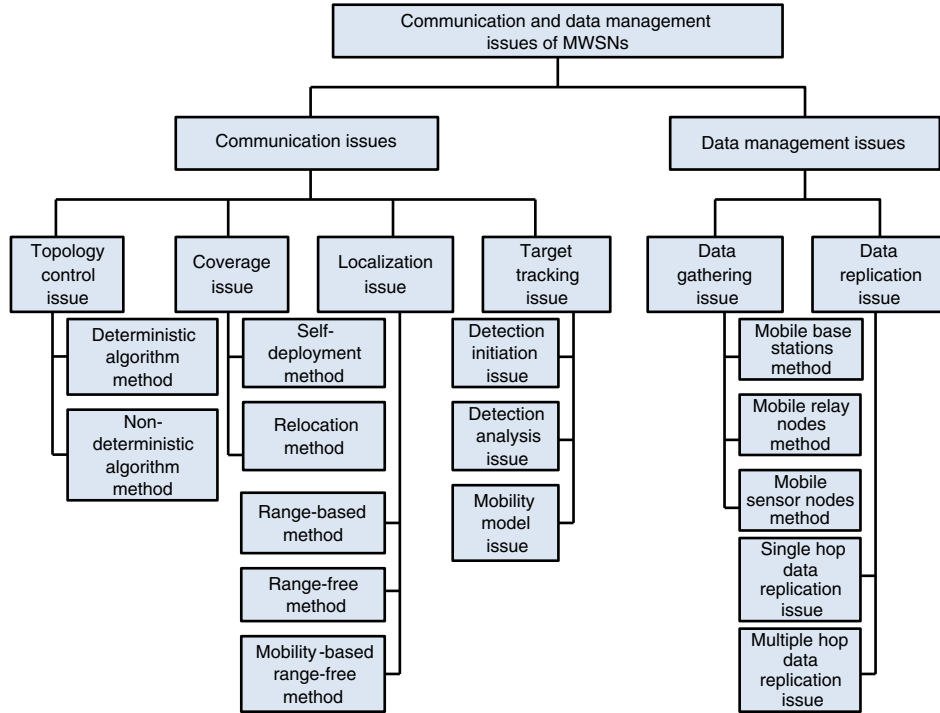


Figure 2.2: Communication and data management issues of mobile wireless sensor networks [2]

2.3 Target Engagement

Sensor-target engagement is one of the essential parts of any sensor management mechanism which aims to control the mobility of sensors to assign a proper subset of sensors to each target at each time slot while guaranteeing the tracking quality [30]. The engagement

problem is concerned with the detection, monitoring and tracking of a target. This process often implies the estimation and prediction of the target trajectory, so as to guide the sensor to maintain a continuous view of the target. Target engagement in a surveillance environment has to consider not only how each sensor goes from one point to another, but also which targets each sensor should track. This makes the engagement process to be essentially a twofold problem, *i.e.* target-to-sensor allocation, and sensor coordination. Figure 2.3 illustrates the Sensor-target engagement problem as a twofold problem.

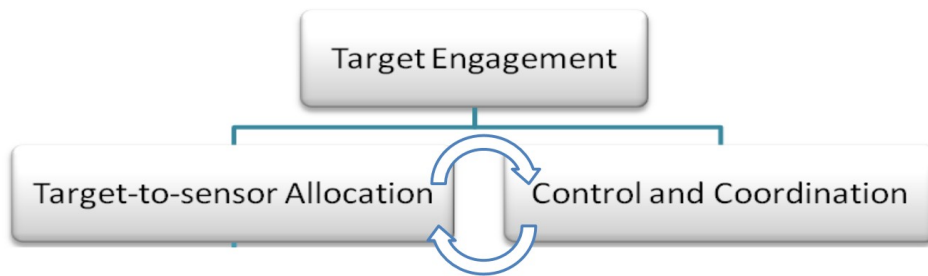


Figure 2.3: Target engagement as a twofold problem

Since the overall sensing performance, to a large extent, revolves around coordination, it is important to find an appropriate target-to-sensor allocation and motion control strategy that lead to optimal positioning of the sensors [31]. In applications, such as the surveillance application considered in this research, the number and density of targets varies in time. The target engagement problem in this case involves finding an appropriate motion control and target-to-sensor allocation strategy resulting in a sensor placement and sensor-to-target assignment that minimizes the number of constraints violation as a cost function. Sensor motion and target allocation, in this case, are not only a function of the relative positions of the sensors, but also the relative positions of the targets with respect to the locations of the sensors [31]. Each sensor should make an appropriate decision on its next location and on the target(s) it will monitor from that location.

2.3.1 Target-to-sensor Allocation

Target-to-sensor allocation deals with a selection process that involves assigning targets to a set of sensors. This selection process is a combinatorial optimization problem in which some sensors, targets and constraints are given and the goal of the system is to minimize constraint violation costs. This problem is an NP-hard one, requiring high computational and communication efforts, especially in large and dynamic sensor networks. Optimization constraints can be used to capture sensor restrictions in tracking a target, such as the inability to track more than one target at a time or a requirement for more than one sensor to track a target. In most of the applications, targets are mobile, and in this case the problem definition changes over time. In such dynamic conditions, even after sensors find a configuration that enables them to track all targets, this configuration may not work over time due to target mobility. Thus, at each moment, the goal is to find the best assignment of sensors to targets with minimum cost. This fact highlights the requirement for control strategies.

2.3.2 Mobility Control and Coordination

Controlled mobility enables a whole new set of capabilities in sensor networks and has attracted a lot of interest in the research community recently. Augmenting static sensor networks with mobile nodes addresses many design challenges of static sensor networks [32]. The authors in [33] categorise the sensor networks based on sensors' mobility level to three different categories: random, predictable and controlled. This research is specifically focused on controllable sensors. Exploiting intentional mobility in a sensor network, instead of considering it as a disturbance, is a fundamental concept that the research community is beginning to appreciate now [33, 32]. Sensor mobility control and coordination concerns the optimal movement of a set of mobile sensors to achieve maximum area and target coverage. The sensors also should coordinate to change their locations from their current position to next optimal positions, avoiding collision and conflicts. Sensor mobility allows better coverage in areas where events or targets appear more frequently. When a target or event appears, the sensors should move such that their positions will eventually approximate that target or event. In addition, such movements should be done by the minimum computation,

memory, and communication requirements [34]. Sensor networks benefit from controllable mobility of mobile sensors in various domains. Deployment, adaptive sampling, network repair, energy harvesting, localization, and event detection are just a few examples of the domains where mobile sensors become handy. This research focuses on the ability of self-organizing as a key benefit of mobile sensors.

2.3.3 Challenges of Target Engagement

Several challenges make the problem of target-sensor engagement more difficult but at the same time more interesting. The most important challenges are related to **scalability** and **dynamicity**. Under scalability, tackling allocation problems in a sensor network with a large number of sensors and targets requires large overheads of communication and computation. On the other hand, most of the sensors have limited computation and communication abilities. These challenges make problem solving complicated in large-scale sensor networks. Dynamicity is another important issue in this field and includes: mobile sensors/targets, new target(s) entering/leaving the VOI, the possibility of sensor failure, and the possibility of new sensors entering the VOI. Having mobile targets makes the problem solving more complicated as the solution for allocating a set of sensors to the set of targets may not work over time due to target mobility. Also, the possibility of additional targets entering VOI, or current ones leaving it, can change the problem. On the other hand, this problem includes both static and mobile sensors. Static sensors only have to select a target in their FOV for tracking, but the case is more complicated for mobile sensors as they have to make decisions about their physical position as well. This latter requirement involves motion control strategies to find optimal sensor movement, which eventually leads to optimal allocation between sensors and targets. In addition, some of the sensors may fail to track a target due to malfunctioning or the presence of obstacles between them and the target. All these dynamicity issues make target engagement a complicated and interesting part of sensor management.

2.4 State-of-the-Art in Target Engagement

The target engagement research field dates back to the early 1990s [35], when it was initially used for military applications. However, it has started to receive increased attention from the research community in the last decade in various new application domains. Over the last decade, a number of research projects have focused on target engagement, allocation or tracking. Many studies have also been published on sensor control and coordination. Nevertheless, a truly comprehensive critical study of target engagement is still absent.

In 1991 Carney [35] applied a method and apparatus for simultaneously engaging a multiplicity of selected targets with one or more missiles (sensors) at each target, to acquire, track, and guide each missile to its respective target. The proposed approach in that paper includes a target selection module, an acquisition and tracking module, a processor and a control module. These modules match the target-to-sensor allocation and control and coordination subsystems as defined in my research. In 2005, the work by Schumacher in [36] presented cooperative moving target engagement as an application example for the MultiUAV [37] cooperative control simulation in which unmanned air vehicles, UAVs, act as sensors, cooperate to track and attack moving ground targets. The described scenario requires a high level of cooperation between sensors and has the additional complexities of nonlinear target dynamics and sensor constraints. In this scenario, three UAVs are required for tracking each target. The authors illustrate that MultiUAV can be effective for scenarios involving severe timing constraints and extreme task coupling. In [38], Chen et al. present a recent study of target engagement. They first modelled the engagement problem and then applied particle swarm optimization based on genetic operators to solve the model.

A survey of the state-of-the-art target engagement approaches is provided in this work. This section studies various features of the target engagement strategies and categorizes the state-of-the-art target engagement approaches based on the classification of this problem that is presented in Figure 2.3. Based on this classification, target engagement approaches are categorized into two categories: target-to-sensor allocation approaches and mobility control and coordination approaches.

2.4.1 Target-to-sensor Allocation Approaches

The target-to-sensor allocation as a resource-allocation problem is an old and challenging problem that has attracted attention from the early 1970s [39, 40, 41, 42] to present [43, 44, 45]. In [42], the authors consider various methods for target tracking and allocation in wireless sensor networks. This book provides a multi-agent perspective of the sensor networks, and then presents solutions to address various resource allocation problems, including target allocation and tracking, using ideas taken from the field of multi-agent systems. Special limitations of sensor networks are considered in the proposed solutions. A sensor's restriction in tracking more than one target at the same time, needing at least k (depending on the type of sensors) sensors to track a target, restrictions on communications between sensors (agents), and the need for soft real-time performance are some examples of these limitations. In the various approaches proposed in this book, key commonalities are considered. For example it is assumed that there is no central decision maker because none of the sensors (agents) can have all distributed information due to communication costs, the size of information sets and other limitations. Therefore, all the approaches proposed in this book emphasize distributed problem solving. Furthermore, the approaches consider the requirement of a real time environment, such as dynamism and uncertainty.

In this study the author divides target-to-sensor problem-solving strategies into eight main subcategories: neural network-based, negotiation-based, classical and heuristic optimization, learning-based, probabilistic and organization-based approaches. Figure 2.6 illustrates these eight categories. In the rest of this section, each of these strategies is explored in the context of a related state-of-the-art approach.

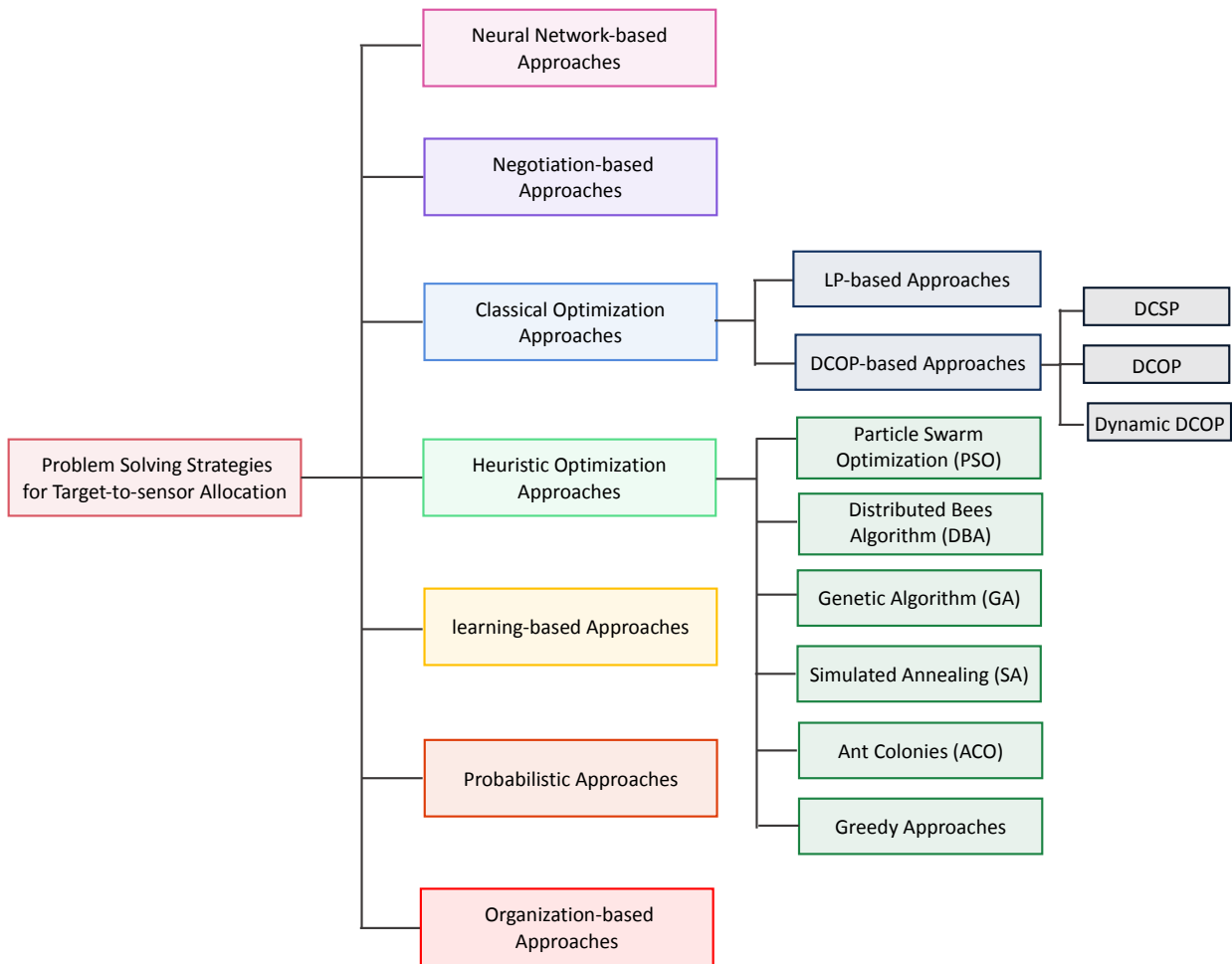


Figure 2.4: Problem solving strategies for target-to-sensor allocation

2.4.1.1 Neural Network-based Approaches

A neural network as defined by the inventor of one of the first neurocomputers, Hecht-Nielsen, is a "parallel, distributed information processing structure consisting of processing elements (which can possess a local memory and can carry out localized information processing operations) interconnected together with unidirectional signal channels called connections." [46]. The research community has focused significant attention on the use

of various types of neural networks in solving the allocation problem in sensor networks. A self-organizing map (SOM) is one of these networks that has shown successful results in the sensor-to-target allocation problem. SOM is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map [47, 48, 49]. Hatime *et al.* in [50] apply SOM to assign targets to sensors (robots) on the basis of competition. This approach uses the Euclidian distance to assign a target to the winning node. The authors compare SOM-based methods with two other approaches and show that SOM offers the lowest cost and running time with the worst workload distribution.

Liu *et al.* in [51, 52] present a new approach for multi-target tracking task allocation in wireless sensor networks, based on multiple elastic neural network modules (MEM) . MEM is a significant extension of SOM [53] and generalizes the self-organizing principles of the SOM to make the model amenable to a wide range of difficult optimization problems. The authors assume that three sensors are needed to form a dynamic coalition for one particular target tracking task. Meanwhile, because of the restrained ability of a sensor, one sensor should be allocated to join in only one dynamic coalition to track one target [51]. In the proposed approach, the disjointed fully connected subgraphs of neurons can be constructed by MEM to solve the sensor-to-target allocation problem. Then the problem changes into finding a dynamic coalition composed of three nodes in the surveillance area and getting the optimal result. With the attraction of the objective function in the optimal problem, the receptive field of the elastic neural network converges and locks each member of the coalition [51]. The authors compare the results of applying MEM with conventional methods and show that the proposed method significantly reduces the tracking system's energy consumption and improves its tracking accuracy.

The work presented in [53] describes two different neural network-based approaches for solving multi-target a multi-sensor passive tracking problem. This research specifically focuses on the allocation of three passive sensors to each target for the purpose of localization by applying a Hopfield neural network to preface the subsequent development of the MEMs. Finally, it describes some of the applications in which the proposed approach is helpful.

2.4.1.2 Negotiation-based Approaches

Negotiation refers to the communication process that facilitates coordination and cooperation among a group of agents [54]. Allocation problems in sensor networks is one of the domains where negotiation-based approaches have demonstrated successful results [55, 56, 57, 58, 59, 60]. Sujit *et al.* in [55] propose a negotiation scheme for task allocation in a military application where multiple Unmanned Air vehicles (UAVs) search for and attack targets in an unknown region. The proposed approach in this research is based on Rubinstein's model of strategic negotiation [61], in which agents make proposals that are either accepted or rejected by other agents. Whether an agent implements its proposal or not depends on what other agents do. The authors in [55] modify the Rubinstein's model to adopt it to allocation problems in sensor networks. In this approach, the UAVs locate a target and relay its coordinates to their closest neighbours. Negotiation among UAVs decides who should attack the target. The authors compare the proposed method with greedy strategy and a variation of the negotiation mechanism, and show that the proposed mechanism performs far better than greedy strategy.

Mataric *et al.* in [56] present a technique based on bidding for task allocation in multi robot systems in uncertain environments, focusing on commitment and coordination. Each robot has to decide whether to finish its current assignment or bid on other tasks, and whether to base its action on local versus global information.

Goldberg *et al.* in [57] present a market-based planning scheme for resource allocation based on a three-layer architecture: planning, executive and behavioural. At the planning level two main components (a trader and a scheduler) apply market-based techniques to allocate task and resources. The trader is responsible for auctioning and bidding, and the scheduler is responsible for determining task feasibility and cost, and interacting with the executive layer. At the executive level, the tasks are executed and synchronized. At the behavioural level, robots create feed-back loops for control and coordination.

Charles Ortiz *et al.* in [62] address the problem of group decision making in a real time distributed environment by considering the dynamicity aspects of the environment. Their main focus is on the decision problem of task assignment in situations for which the information relevant to the assignment problem is distributed over a group of agents

(sensors). They use a negotiation-based approach. Rather than restart a negotiation whenever a problem statement has changed, the authors propose an anytime algorithm named *Dynamic Mediation*, which can adopt a previously found solution to such changes. In the *Dynamic Mediation* approach, which is a partially center-based method, a leader called the mediator selects the working sensor nodes according to specific criteria. All the selected sensors send their negotiation information to the mediator, who then implements an iterative and interactive hill-climbing search in a subset of a solution space and sends the successive proposals to the group of sensors participating in the session. Next, the group members make their responses based on these proposals. The mediator can find a satisfactory outcome to the assignment problem based on those responses. This method suffers from all the problems common in centralized methods, including a single point of failure and computational overload of the central node. This method can only be used for small scale networks. Otherwise, the mediator will be overloaded. Moreover, when the mediator fails, the network fails as well.

2.4.1.3 Classical Optimization Approaches

Optimization problems involve finding the best solution among all feasible solutions. Mathematical optimization problems are able to model a wide range of engineering or scientific applications from modelling the evolution of species in biology to the creation of communication networks [63]. The target-to-sensor allocation problem is one of these applications that can be effectively translated into several optimization frameworks. Linear programming (LP) and Distributed Constraint Optimization Problem (DCOP) are two main classical optimization problems that are discussed in this section and have attracted lots of attention in the research community for modelling target-to-sensor allocation problems [18, 64, 65, 66, 50, 41, 67, 68, 3, 69, 42, 62, 70, 71]. This is due to the natural formulation of the target-to-sensor allocation problem as an optimization one with numerous attributes and constraints.

2.4.1.3.1 LP-based Approaches Linear programming (LP) is a method of finding the best outcome in a mathematical model whose constraints are represented by linear

relationships. LP has been widely used as a tool in decision making [72, 73, 74] and has a wide range application in the real world. This problem consists of a linear objective function, a set of linear constraints, and a finite set of variables. Integer LP (ILP) is a specific type of LP in which some or all of the variables are restricted to integers. Hatime *et al.* in [50] presents a translation of the target-to-sensor allocation problem as an ILP in which the decisions are limited to 1 (assigned to a target) or 0 (not assigned) and the objective is to find a minimum cost assignment of n robots (sensors) to m targets:

$$\text{minimize } \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}. \quad i = 1, \dots, n, \quad j = 1, \dots, m, \quad \text{subject to} \quad (2.1)$$

$$\sum_{j=1}^m x_{ij} \leq b_i \quad b_i = \begin{cases} 1 & \text{if } n \geq m \\ \text{ceil}(m \div n) & \text{otherwise} \end{cases} \quad (2.2)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad (2.3)$$

$$x_{ij} = \begin{cases} 1 & \text{if robot (sensor) } i \text{ is assigned to target } j \\ 0 & \text{otherwise} \end{cases} \quad (2.4)$$

where c_{ij} is the cost of robot i to service target j and b_i is the maximum number of targets that robot i can serve. After modelling the problem as an optimization one, the authors [50] apply a branch-and-bound base solution to solve the modelled problem.

In 1977 [41], Nash presented an optimal method for allocation of several prioritized targets to limited sensor resources in a surveillance application. In this research work he translates the target-to-sensor assignment problem into the framework of classical linear programming (LP) problem and employs mathematical programming and Kalman Filtering to generate preferred sensor-to-target assignments in a generic surveillance context.

In 2005 [67], Cardei *et al.* proposed a method to extend network life time by modelling the problem as the maximum set cover problem, and then applied LP and a greedy approach to solve the modelled problem. In this paper LP is applied as a solution instead of a modelling approach. In the same year, Liu *et al.* in [68] presented a scheduling

method based on LP to schedule the assignment of sensors to targets with the objective of maximizing the life time of the surveillance system.

Later in 2007 [3] Cai *et al.* modelled the same problem focusing on directional sensors that have a limited angle of sensing range due to technical constraints. Figure 2.5 shows the target-to-sensor allocation problem using such directional sensors in which s_1, s_2, s_3 are three sensors that can switch to three directions and the stars a_1, a_2, a_3 are three targets. Parts (a) and (b) of this figure show the status of the target-to-sensor allocation problem with two different active directions of the sensors.

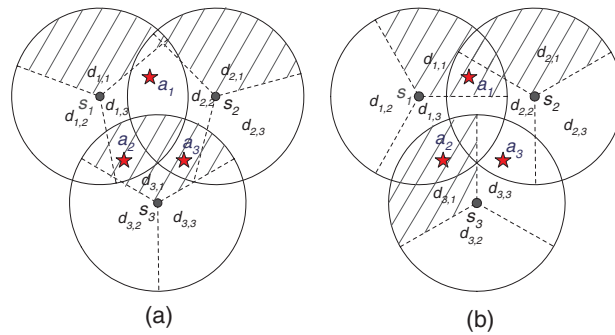


Figure 2.5: Simple directional sensor networks [3]

Recently in 2013 [69] Shen *et al.* considered the target-to-sensor allocation problem with equality or inequality constraints and applied linear programming for solving this problem in case the constraints are temporally inseparable.

2.4.1.3.2 DCOP-based Approaches One important approach for solving the target allocation problem in sensor networks is the Distributed Constraint Satisfaction/Optimization Problem (DCSP/DCOP) as a formulation environment. The target-to-sensor allocation is then tackled using DCSP algorithms. In this section, first the DCSP, DCOP and its dynamic version are introduced then related work on using this modelling technique to address the target-to-sensor allocation problem are presented.

2.4.1.3.2.1 Distributed Constraint Satisfaction and Optimization For many years, finding an optimal assignment of values to a set of variables spread over a number of agents that have interdependencies, has been an important issue in distributed problem solving. DCOPs provide a good formalism for representing problems involving different agents, each responsible for assigning a value from its finite domain to its variable. These agents coordinate their values to find value assignments that minimize constraint costs. Two agents are constrained if they share a constraint. Each constraint cost is a function of the values of the involved variables. The cost of a solution is the sum of all the constraint costs, depending on the values assigned to them in the solution. The goal of solving a DCOP (which is an NP-hard problem [4]) is to find an assignment of values to all the variables that minimize the solution cost. In this problem, each agent knows only about the constraints in which its variables are involved. Many real-world problems in Artificial Intelligence (AI) and Multi-Agent Systems (MAS) can be modeled as DCOPs. A number of resource allocation problems are examples of this real-world problems: resource allocation in sensor networks [42, 75], urban traffic signal control [76, 77], the disaster rescue problem [78] and many timetabling problems such as train timetabling [79], and university timetabling [80]. The common goal in solving all of these problems is to find suitable values to assign to distributed variables.

2.4.1.3.2.2 DCSP Definition: DCSP refers to the distributed form of the constraint satisfaction problem (CSP) . It involves multiple autonomous agents, each agent holding one or more variables. DCSP was first discussed by Sycaro *et al.* and Yokoo *et al.* [81, 82]. CSP as the basis of DCSP is formally defined as follows [81, 82]:

- A set of n variables: $V = \{X_1, X_2, \dots, X_n\}$
- A set of finite, discrete domains for each variable: $D = \{D_1, D_2, \dots, D_n\}$
- A set of constraints: $R = \{R_1, R_2, \dots, R_m\}$ where each $R_i = (d_{i1}, d_{i2}, \dots, d_{ij})$ is a predicate defined on the Cartesian product $D_{i1} \times D_{i2} \dots \times D_{ij}$. If the value assignment of these variables satisfies this constraint, the predicate returns true; otherwise it returns false.

The final goal of CSP is to find an assignment $A = \{d_1, d_2, \dots, d_n \mid d_i \in D_i\}$ that satisfies all the constraints in R . In the distributed form of this problem, one or more variables along with their constraints are assigned to each agent in the distributed environment. Each agent attempts to reach this goal not only by satisfying its local constraints but also by communicating with other agents to solve external conflicts. Clearly, the agent goals are interconnected. Agents are expected to have strong communication with one another because their goals are interrelated. For example, in order to solve its sub-problem, each agent may create new conflicts for other agents by changing its or other agents' values. In this report it is assumed that agents can communicate with one another by exchanging messages and that the receiver agent receives messages exactly in the order they were sent, of course, after a finite delay. It is further assumed that only one variable is under the control of one agent. Thus, the name of the agent can be the same as that of the variable it holds and manages. Each agent has complete information about the constraints on its variable.

2.4.1.3.2.3 DCOP Definition: DCOP is a generalized form and optimization version of *Distributed Constraint Satisfaction Problem (DCSP)*, in which the constraints are weighted [83, 84]. In contrast to DCSP, in which each constraint has two states: satisfied or unsatisfied, in DCOP each constraint has a cost that depends on the values selected for the variables involved, and the goal is to minimize the sum of the constraint costs. Therefore an objective function is defined for each DCOP, and the goal for the agents is to find an assignment of values to variables such that the objective function is minimized. The objective function is defined as the summation over a set of cost functions [4]:

$$F(A) = \sum_{x_i, x_j \in V} f_{ij}(d_i, d_j) \quad \text{where } x_i \leftarrow d_i, \quad x_j \leftarrow d_j \text{ in } A. \quad (2.5)$$

In which $f_{ij}(d_i, d_j)$ is the cost function for a pair of variables x_i, x_j and is defined as $f_{ij}(d_i, d_j) : D_i \times D_j \rightarrow N$. A in equation (1.1) represents an assignment of values to the variables. It is common to represent a DCOP as a constraint graph. In this weighted graph the vertices are the variables and the edges are the constraints. Figure 2.6 [4] shows an example of a constraint graph and its cost function. In this example, DCOP has four variables $\{X_1, X_2, X_3, X_4\}$. The domain of each variable is $\{0, 1, 2\}$. Considering f as the

cost function, each assignment provides a cost and the goal is to minimize the cost of the solution found.

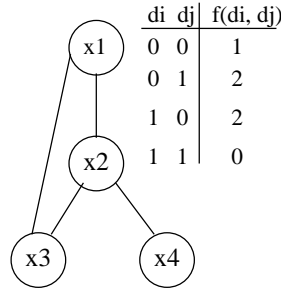


Figure 2.6: Constraint graph and its cost function [4]

2.4.1.3.2.4 Dynamic DCOP: In most real-world problems it is often the case that after a DCOP has been solved, small changes occur that make the solution of the original problem invalid. This fact introduces the Dynamic Distributed Constraint Optimization Problem (DynDCOP). DynDCOP is a generalization of DCOP that allows the variables and constraints of the problem to be added or removed. These changes in the problem statement are the results of external environmental condition changes. Considering the dynamicity is important when a real-world problem is modeled as a DCOP because it is an inseparable part of most real-world problems. In a dynamic problem, a DCSP/DCOP solution that works at one time may become outdated when the problem changes. DynDCOP, as the dynamic version of DCOP, can be formulated as $\langle P_{initial}, \Delta \rangle$, in which $P_{initial}$ is the initial state of DCOP and $\Delta : P_t \rightarrow P_{t+1}$ is a function that maps a DCOP in time t to a DCOP in time $t + 1$ by creating and expunging variables or constraints or changing the domains [85]. The dynamicity level of the problem can be evaluated by counting the number of changes between two states of the problem. For example, the number of created or expunged constraints or variables.

2.4.1.3.2.5 DCSP/DCOP for Target-to-sensor Allocation The research community has focused significant attention on the use of DCOP/DCSP in solving the target-to-sensor allocation problem. This is due to the several advantages of DCOP modelling

in the field of target allocation. The first and most important one is that it provides a good abstract formalization of the problem. In particular it provides a formalism that allows a general mapping of different types of target-to-sensor allocation problems into a well-known problem solving paradigm [27]. In addition this modelling has the advantage of using various existing DCSP algorithms to solve the modeled problem, which eventually leads to a solution for the main target-to-sensor allocation problem in sensor network domains. Because of the advantages of using this modelling approach to solve target-to-sensor allocation problems, various aspects of DCSP have been explored in the context of target-to-sensor allocation especially in last two decades [86, 87, 66, 18, 64, 65]. In general, the proposed methods in this field can be divided into two main categories. The first category includes research provided to evaluate a DCSP algorithm in a real-world domain such as sensor networks. The research focusing on this category mainly use simple target allocation scenarios in the sensor network and then try to evaluate the DCSP algorithm on the modelled problem by using a much-simplified model of the sensor networks. These approaches do not consider most aspects of a real sensor network. The second category includes methods that are presenting new modelling approaches for modelling target-to-sensor allocation problems. These approaches aim to present modelling approaches that better match with the requirements of a practical sensor network. Both categories use the same modelling technique (DCSP/DCOP modelling) to solve the same problem, but they differ in their purpose. As a result they focus on different aspects of this modelling. The Next paragraphs present the research already done in each of these two categories.

2.4.1.3.2.6 Sensor Network a Test-bed for DCSP/DCOP Algorithms Research presented in [86], [88], [89] and [64] can be categorized in the first category. Cesar Fernandez *et al.* [86, 88] introduce SensorDCSP as a real-world domain for evaluating some of the DCSP algorithms in this new area. SensorDCSP, which is the main contribution of this work, involves a sensor network in which static sensors track some mobile targets. The authors show that there is a satisfactory transition phase in this NP-hard problem, which makes it an interesting benchmark for testing DCSP algorithms on an easy-hard-easy problem. In addition the framework provides a truly distributed environment for this evaluation, which was lacking in this domain. After that, they evaluate two important

DCSP algorithms, asynchronous backtracking (ABT) [82] and asynchronous weak commitment search (AWC) [90] on the new proposed domain. They find that AWC works better in satisfiable instances but ABT has a better performance on unsatisfiable ones; and this is due to the dynamic-variable-priority-changing method used in AWC. To make the scenario more realistic, they add some communication delays to the SensorDCSP caused by different traffic conditions and then study the reaction of the algorithms on them. Roger Mailler *et al.* [89] present a new DCSP algorithm called Asynchronous Partial Overlay (APO) and used target-to-sensor allocation problems in sensor networks as a benchmark to evaluate their new proposed algorithm. In APO, some agents, called mediators, first centralize some parts of the problem, then use central search methods to solve the centralized sub-problems. Finally, they inform the other agents about the solution of the sub-problems. The sub-problems have some overlaps, and the agents increase the size of their sub-problems along critical paths within the DCSP. To test APO in a sensor network domain, the authors placed some sensors and some targets in an environment randomly. In this domain, each sensor can see only targets that are placed at a *dist* from that. To map the sensor-to-target allocation problem to a DCSP, they mapped the *targets* to the *variables* and the *set of sensors* that can see each target (variable) to the *domain* of that variable. Constraints are also defined using a “not intersects” relationship. This relationship is true if the sensors assigned to two targets do not have any elements in common with each other. Finally, this study shows that APO outperforms previous DCSP algorithms in this domain.

Reference [91] proposes a new algorithm called SDS for solving DCSP. They test the proposed approach on the sensor network domain. SDS uses piggybacking to decrease its communicational overhead, a fact that makes it suitable for use in sensor network domains. In addition, the proposed approach is reliable against message losses but cannot guarantee to find a solution for the problem if there is one.

Macarthur in [64] casts the dynamic task allocation problem in sensor networks to the DCOP. Then in this paper, a number of decentralized DCOP algorithms including DPOP, ADOPT, and the GDL family of algorithms are discussed for solving the modelled problem and their shortcomings are highlighted. Finally, the max-sum algorithm is reported and discussed as the best DCOP algorithm for performing multi-agent coordination in a

dynamic task allocation scenario.

All the research done in the first category suffer from a main drawback: the mapping of task allocation of a large-scale sensor network to a distributed constraint satisfaction problem is over simplified. Furthermore, none of the proper distributed constraint satisfaction algorithms identified work well, in complex task allocation issues in large-scale sensor networks. As explained earlier, the aim of research in this category is to test DCSP algorithms on a new domain not to find an appropriate solution for target-to-sensor allocation problems in sensor networks.

2.4.1.3.2.7 Target-to-sensor Allocation to DCSP/DCOP Modelling Approaches

The group of research that has been done in this field focus more on solving the allocating problem in sensor networks than on evaluating DCSP algorithms. Research done by Bejar *et al.* in [87], by Matsui *et al.* in [92], by Ota *et al.* in [93, 94], by Farinelli *et al.* in [65] and by Zivan *et al.* in [66] can be categorized in this group. These researchers are trying to provide more efficient techniques for modelling the target-to-sensor allocating problem of sensor networks as a version of DCSP. To the best of my knowledge, the first of these works can be found in [87]. They introduce two DCSP modelling techniques to model the tracking of mobile nodes in wireless sensor networks with static sensors. In the first modelling approach, called “sensor centered DCSP”, agents are associated with the sensors, while in the second one, called “target centered DCSP”, they are associated with the mobile nodes. They show that these two modelling approaches are duals of each other, and then demonstrate how these two modelling approaches differ in their inter-agent and intra-agent communication costs. Finally, the computational costs of these two approaches are compared. The authors use binary variables in both proposed modelling approaches; a variable is equal to one if a sensor is assigned to a mobile node and otherwise zero. This modelling approach and its limitation will be discussed in more detail in Chapter 4. This paper uses a very simplified and small version of the sensor network domain for this modelling, and the main focus of the paper is to present a basic modelling approach. The “mobile node centered” approach is later used by other researchers to model the problem. Although most of the work done in this field uses this modelling technique, the proposed approach cannot work efficiently in large-scale sensor networks. Matsui *et al.* in [92, 94]

and Ota *et al.* in [93] show two DCOP-based formalization for target allocation. They also present a two-layer DCOP formalization using an agency model to restrict the computational cost of the problem solving. Roman Bejar *et al.* [87] and Toshihiro Matsui [94] report common target-to-sensor modelling techniques. There are mainly two modelling techniques:

1. Sensor-based modelling
2. Target-based modelling

In the first case, the sensors are modeled as the variables of the DCOP. Then each variable's domain and constraints are defined based on this mapping; while in the second case, each target is modeled as a variable in DCOP. In the Sensor-based modelling approach, each sensor agent has a binary variable corresponding to each target in its range. A sensor agent assigns a value 1 to a variable if it is going to track this target. Since each sensor can track at most one target in its range, at most one of its variables can be set to one. In addition, k communicating sensors should track each target to ensure complete covering. These two criteria represent the DCOP constraints in this model. The target-based modelling approach is quite similar to the sensor-based modelling approach with the difference that the binary variables are mapped to targets and each element in the variable domain is corresponding to one sensor that can track the target [87].

Reference [94] emphasizes on presenting a constraint optimization approach instead of a constraint satisfaction one for resource allocation problems in sensor networks. The authors develop a distributed cooperative observation system based on an agency model and applied the model into the sensor network. They first compare the agency model with a DCOP based modelling approach and then present a cooperative formalization to integrate DCOP approach into the agency one to create a combined model. In this paper a more realistic view of the sensor network is presented, which includes autonomous camera sensor nodes, that use pan-tilt-zoom controlled cameras, computers and local area networks, and some randomly moving targets. The authors in [94] use two approaches for DCOP modelling: one called STAV (Sensor-Target as Variable) that maps each (sensor, target) to one variable and one called TAV (Target As Variable) that maps each (target)

to one variable. Then, for each case, appropriate constraints are defined. Although the authors tried for more realistic assumptions, as in a sensor network, targets are out of the control of the system, therefore neither approach works because they assign either targets or (sensor, target)s to variables, and in DCOP, the algorithm can control only the variables. In this approach, the sensors are also considered to be static. This paper suffers from poor experimental results in which none of the DCOP solvers are applied to solve the problem; instead, a simple hill-climbing approach is used.

Farinelli *et al.* in [65] present a novel agent-based representation of the allocation problem in sensor networks based on a factor graph, and then use state-of-the-art DCOP heuristics to generate sub-optimal solutions. This approach attempts to minimize sensors' energy consumption by controlling their sense/sleep schedule while maximizing target detection probability. Finally, they have tested the proposed DCOP-based model in a wide-area surveillance problem.

Recently Zivan *et al.* in [66] present a new method of DCOP modelling for target tracking in dynamic environments such as surveillance applications in which the physical position of the sensors are mapped to the variables, the nearby locations are mapped to the domain values for each variable and the targets are mapped to constraints. Consequently, variable domains and constraints change as the sensors move through the environment.

2.4.1.4 Heuristic Optimization Approaches

Heuristic approaches are designed to solve problems more quickly when classic optimization approaches are too slow, or to find an approximate solution when classic approaches fail to find an exact one. Particle Swarm Optimization (PSO) , Genetic Algorithm (GA) , Simulated Annealing (SA) , Ant Colonies (ACO) and greedy approaches are just a few examples of heuristic optimization methods that have been applied on the sensor allocation problem [95, 96, 97, 98, 99, 100, 101, 102].

Jevti *et al.* in [103, 96] optimize and apply Distributed Bees Algorithm to distributed target allocation in a swarm of robots. They have applied genetic algorithm to control the parameters of the Distributed Bees Algorithm to reach the cost efficient allocation between sensors and targets. The decision making in this approach is completely distributed

and each sensor independently decides on tracking a target based on targets' quantities and distance. Mei *et al.* in [104] apply PSO, based on nearest-neighbour algorithm, for task allocation in sensor networks to reduce energy consumption between nodes. In this approach targets are allocated to sensors initially based on nearest-neighbour algorithm which is a greedy approach, then the fitness function is compared through the change of task allocation matrix to achieve lower cost allocations.

Wang *et al.* in [98] present a hybrid algorithm to optimize the energy consumption in sensor networks, in which particle swarm optimization and simulated annealing are combined to find the an allocation solution in a distributed manner. Tseng *et al.* in [100] propose the solution to select three nearest neighbours to each target to form a group, given that at least three sensors are required to find the location of and track a target. In this greedy approach they do not consider very important factors such as the random distribution of sensors, energy consumption for communications and localization accuracy. Also many papers, such as the research presented in [105, 106], consider only a single target scenario. They do not mention multi-target tracking task allocation and the conflicts that targets may have in assignment to limited available sensors.

2.4.1.5 Learning-based Approaches

Learning-based approaches refers to a group of algorithms which can learn from data. These algorithms create a model based on inputs and apply that to make predictions and decisions [107]. A number of researchers in the field of resource allocation in sensor networks focus on various types of learning algorithms [108, 109, 110].

Campbell *et al.* in [108] propose two learning-based approaches: greedy and best fit. Through a series of experiments, they illustrate that learning is an important factor in task allocation. Kreucher *et al.* in [109] focus on detection and tracking smart targets in surveillance applications applying the reinforcement learning approach. In this context a smart target is a target that can react in a manner that makes future surveillance more difficult. As detecting and tracking smart targets in surveillance application is very challenging computationally, they propose a two stage approach in which the targets are first detected and then a tracking algorithm is applied to track already detected targets. Main-

land *et al.* in [110] present a resource allocation approach called Self-Organizing Resource Allocation in which sensors sell their readings or data aggregates as goods in response to prices that are established by the programmer. Sensors try to maximize their benefit in this virtual market, subject to energy budget constraints. Sensors apply reinforcement learning to adapt their operation over time based on received feedback from the payments.

2.4.1.6 Probabilistic Approaches

Probabilistic approaches are the approaches in which the result or the way the result is obtained depend on a degree of randomness or chance. The work presented in [111, 112, 113, 114, 115] are some of the examples of the research that are focusing on probabilistic methods for resource allocation in sensor networks.

In [111] the authors present a probabilistic approach for optimal sensor allocation in a structural health monitoring (SHM) system. They apply an approach based on weights of a neural network trained from simulations using a priori knowledge to determine the optimal number and locations of sensors for SHM. Bar-Shalom in [113] present an overview of the application of probabilistic data association (PDA) approaches in target tracking scenarios. Target motion analysis as one of the applications of PDAs is discussed in this paper, focusing on manoeuvrable targets. The use of the PDA technique for tracking highly manoeuvring targets combined with radar resource management is described in this paper.

2.4.1.7 Organization-based Approaches

Team-based organizations filter decision making down to all levels of management, while traditionally structured organizations rely on top management to make decisions. Various organization types have been proposed in the field of sensor networks: hierarchical, holonic, peer-to-peer and geographical are just a few examples of these organizations.

Osher Yadar *et al.* [70] are the first to propose a hierarchical structure in large-scale wireless sensor networks. They propose a system called Distributed Dispatcher Manager (DDM) for resource management in very large-scale domains. In this method sensors are modeled as cooperative mobile teams and targets are assumed to be distributed over the

space. Each agent has only its neighbours' information. DDM, which organizes teams hierarchically, employs an event-driven method such that when events occur they are reported to the top of the hierarchy in a bottom-up manner. Reactions to the reported events are processed in a top-down manner. In this paper the authors try to answer two important questions: 1. How can sensors expand their local information to better assess the problem? 2. How can partial solutions found by sensors be combined to make a global solution for the problem? The main drawback of this structure is that it cannot rapidly respond to events. Furthermore, this structure is so complex that it is hard for the sensor nodes to form a hierarchy autonomously.

Bryan Horling *et al.* [71] introduce organizational design for task allocation in large-scale wireless sensor networks. They first try to use various multi-agent organizations and show how they can address the challenges in large-scale sensor networks. For each organization, first a high-level architecture is presented, then the effects each architecture has on the problem due to its specific characteristics are discussed. Afterwards the authors show that dividing the problem into sub-problems leads to better locality and more efficient communications.

In the proposed partitioning design, the environment is first partitioned by the agents into sectors according to the location of the sensor nodes. The purpose of this division is to limit the interaction needed between sensors. Much of the communication is limited to within the sectors. The selected partitioning technique increases the scalability of the proposed architecture. Each sector has an organizational manager called a sector manager, which acts as a hub in the hierarchical structure. Each sensor interacts mostly with its sector manager, but it can also have limited interaction with other sectors' sensors or managers. One of the most important factors in this partitioning method is the size of the partitions. The authors demonstrate how an improper division can increase load on the partitions or the problem as a whole. This method reduces the energy used by limiting the communication range. However, when a target moves from one sector to another, target-to-sensor allocation becomes difficult.

2.4.2 Motion Control and Coordination Approaches

Control and coordination of sensor mobility is an important challenge in self-organizing surveillance systems that has attracted a great amount of effort within the research community [116, 117, 118, 119, 120, 121, 122]. Various research studies follow different objectives for this problem. Maximizing area and/or target coverage [19], maximizing radio coverage between the sensors [20, 21] or improving target coverage over manoeuvrable targets [22] are some of the examples of these objectives. This research focuses on maximizing simultaneous area and target coverage.

Cortes *et al.* [120] present a control and coordination algorithm for a group of mobile sensors. The deployment of sensors in this work is region based and the quality of sensing is simplified as a smooth function of distance. Therefore the coverage performance is described by a smooth utility function. This function is then applied to propose a distributed, gradient descent motion control algorithm for a team of sensors to achieve optimal coverage of the field. Shu and Liang [121] propose a fuzzy optimization algorithm to find an optimal deployment of a set of sensors, thus maximizing the field coverage, given a certain number of sensors. The motion decision of a sensor in this approach is mainly based on the distance between each sensor and its neighbours.

A body of research on sensor motion control and coordination focuses on resource allocation and management of unmanned air vehicles (UAVs) as an example of mobile sensors. For example, Bertuccelli *et al.* [117] and Bellingham *et al.* [123] study target allocation and engagement problems in the presence of uncertainties. In [118] a stochastic game-based approach is applied for unmanned ground and aerial vehicle motion control in a pursuit-evasion problem.

Parker and Emmons [124] present a motion control strategy for monitoring multiple targets based on the ALLIANCE architecture. They developed a distributed control strategy in which the sensors attempt to maximize the collective time during which each target is monitored by at least one sensor. They also show that this problem is an NP-hard one. Jung and Sukhatme [125] suggest a two-level region based approach for multi-target tracking using both static and mobile sensors. In the first level, a controller distributes sensors across regions based on density estimates. Then a target-following controller tries

to maximize the number of covered targets in each region. In this approach, the motion control is designed based on the distance between the sensors and the center of gravity of the targets.

Tang and Ozguner [31] address the problem of motion planning in a surveillance system with multiple targets and multiple sensors in which the number of sensors is less than the number of targets. They formulate the motion planning problem as an optimization problem in which the objective is to minimize the average time duration between two consecutive observations of each target. They apply this objective function for developing a cooperative online motion planning approach. A common point in many papers on motion planning and sensor deployment problems in sensor networks is the problem formulation method. These researchers usually first formulate the problem as an optimization problem. Therefore, the method that is applied to solve the optimization problem in each case is confined by the characteristics specific to the objective of that optimization problem. In some cases, the objective function is a smooth one, so a linear programming approach [123] or a gradient descent one [120, 116, 121, 126] can be useful. If the objective function can be divided into some sub-objectives, then a decentralized control solution strategy can be applied [127, 120, 116, 121]. In a general view, it seems that the goal of target engagement is almost no-smooth, multi-objective and NP-hard. Therefore, heuristic and near to optimal approaches [128, 125, 122] better match with the requirements of the problem [31].

2.4.2.1 Flocking-based Approached

The Flocking algorithm is one of the approaches recently reported in the literature that addresses the issue of sensor control and coordination in sensor networks. This algorithm has attracted significant interest in recent years in the field of mobility control [129, 5, 130, 131, 132, 133, 134, 135, 136, 137]. Flocking is a biologically inspired behaviour that embodies a form of cooperative behaviour of a large number of autonomous interacting agents to achieve a coordinated group behaviour. Group movements of birds, fishes, insects and bacteria are examples of the flocking behaviour in nature. To conceive flocking behaviour, each agent follows a set of flocking rules and maintains some sort of communication with its neighbouring agents. Self-organization and local communication

requirements of the flocking process provide an inspiring behaviour in the management of sensors in mobile sensor networks.

Flocking-based algorithms have several advantages that make them suitable for use in sensor management. First, they are completely distributed algorithms; therefore, they are highly compatible with the distributed nature of sensor management in sensor networks. Second, in Flocking-based algorithms, each particle needs to communicate only with its neighbours; thus, using Flocking-based algorithms for sensor management requires only local communication between sensors. Third, because in Flocking-based algorithms, particles apply simple Flocking rules, using this type of algorithm for sensor management has low computation overhead for the sensors. In addition, Flocking-based algorithms are inspired from nature, and have been shown to behave well in self-organized networks. Furthermore, they are highly flexible and scalable, and thus, they are suitable for large sensor networks. The advantages of Flocking-based algorithms and their high compatibility with mobile sensor networks motivate us to focus on them as a useful approach to tackle the sensor management problem in a self-organizing surveillance system. Flocking and Anti-Flocking algorithms are two Flocking-based algorithms that can be used for mobility control of sensors in surveillance systems and applications.

2.4.2.1.1 Flocking Algorithm Flocking is a process that embodies a form of collective behaviour of a distributed group of autonomous agents. This behaviour is accomplished via simple local interactions, a common group objective, and without any global information. The Flocking algorithm is a biologically inspired process that mimics the collective behaviour of birds. This algorithm is based on three main rules [129]:

- Flock centring: stay close to nearby flock-mates;
- Collision avoidance: avoid collision with nearby flock-mates;
- Velocity matching: match velocity with nearby flock-mates

Olfati-Saber [5] propose a theoretical framework with a solid mathematical background for the design and analysis of a distributed Flocking algorithm and prove that this frame-

work does not suffer from fragmentation and collapse. He subsequently applies this algorithm to control the movement of sensors [130, 133, 132]. This algorithm consists of n sensors (called α – agent) that moves in a m – dimensional space with dynamics:

$$\begin{cases} \dot{q}_i = p_i \\ \dot{p}_i = u_i \end{cases} \quad (2.6)$$

where $q_i, p_i, u_i \in \mathbb{R}_m$ denote the position, velocity, and control of agent i , respectively. In this algorithm each sensor applies a control input that consists of three components:

$$u_i = f_i^g + f_i^d + f_i^\gamma \quad (2.7)$$

where f_i^g is a gradient-based component, f_i^d is a velocity consensus component and f_i^γ is navigational feedback due to a group objective. The control function for each sensor i (u_i) composed of two elements $u_i^\alpha + u_i^\gamma$ ($u_i = u_i^\alpha + u_i^\gamma$) in which:

$$u_i^\alpha = \underbrace{\sum_{j \in N_i} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij}}_{\text{Gradient-based term}} + \underbrace{\sum_{j \in N_i} a_{ij}(q) (p_j - p_i)}_{\text{Consensus term}} \quad (2.8)$$

where, in Equation 2.8, N_i represents the set of neighbours of sensor i , $\phi_\alpha(z)$ is an action function that is defined in [5] as follows:

$$\phi_\alpha(z) = \rho_h(z/r_\alpha)\phi(z - d_\alpha) \quad (2.9)$$

$$\phi(z) = 1/2[(a + b)\sigma_1(z + c) + (a - b)] \quad (2.10)$$

where r_α and d_α are constant parameters of α -lattice. $\sigma_1(z) = z/\sqrt{(1 + z^2)}$; $\phi(z)$ is an uneven *sigmoidal function* with parameters a, b, c such that

$$0 < a \leq b, c = |a - b|/\sqrt{4ab}. \quad (2.11)$$

$\rho_h(z)$ is a bump function that smoothly varies between 0 and 1 and is defined in Equation 2.12 [5]:

$$\rho_h(z) = \begin{cases} 1 & z \in [0, h) \\ 1/2[1 + \cos(\pi((z - h)/((1 - h))))] & z \in [h, 1] \\ 0 & otherwise \end{cases} \quad (2.12)$$

The other parameters of Equation 2.8 are defined as follows: $\|q_j - q_i\|_\alpha$ represents the σ - norm of a vector connecting q_i to q_j defined as [5]. This σ - norm is defined as:

$$\|z\|_\sigma = 1/\varepsilon[\text{sqr}(1 + \varepsilon \|z\|^2) - 1] \quad (2.13)$$

$$n_{ij} = \nabla \|q_j - q_i\|_\sigma = (q_j - q_i)/\sqrt{(1 + \varepsilon \|q_j - q_i\|^2)} \quad (2.14)$$

where n_{ij} is a vector along the line connecting q_i to q_j , and $(0, 1)$ is a fixed parameter of the σ - norm.

Finally, $a_{ij}(q)$, in the consensus term of Equation 2.8, is an element of the spatial adjacency matrix and is defined as follows [5]:

$$a_{ij}(q) = \rho_h(\|q_j - q_i\|_\sigma / r_\alpha) \in [0, 1] \quad (2.15)$$

The second part of $u_i = u_i^\alpha + u_i^\gamma$, i.e., u_i^γ , is the navigational feedback and is defined based on the sensors' group objective. If the group objective is to track a target at position q_t moving with velocity p_t , u_i^γ is as defined in Equation 2.16. In this equation c_1 and c_2 are positive constant values.

$$u_i^\gamma = f_i^\gamma(q_i, p_i, q_t, p_t) = \underbrace{-c_1(q_i - q_t) - c_2(p_i - p_t)}_{\text{Navigational feedback}} \quad (2.16)$$

To use the Flocking algorithm for target tracking, each sensor i needs to apply u_i as an input control. The result of this application is a mass of sensors around the target. Fig. 2.7 shows the final position of sensors applying this algorithm [5]. It is shown that creating this mass improves the performance of target tracking [131].

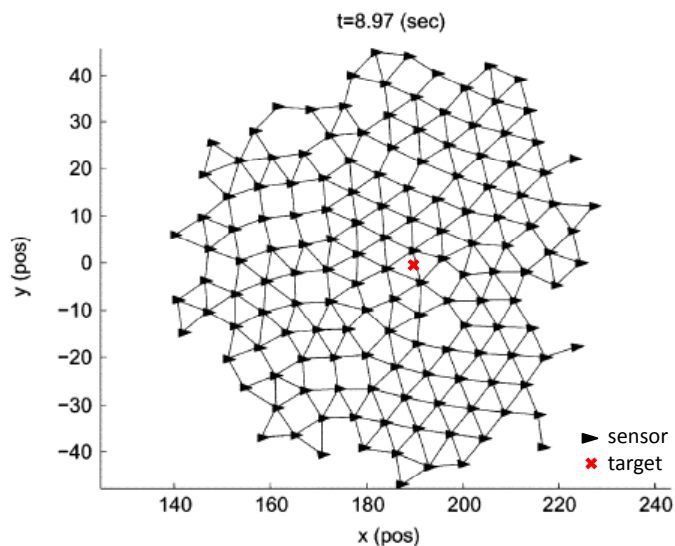


Figure 2.7: Flocking for $n=100$ agent [5]

Applying the Flocking algorithm in a surveillance application results in the creation of a flock of sensors around the first entered target. This algorithm provides reliable target coverage for the first target, but as all the sensors are engaged in tracking the first target, none remains to track other targets (this problem is depicted in Fig. 2.8).

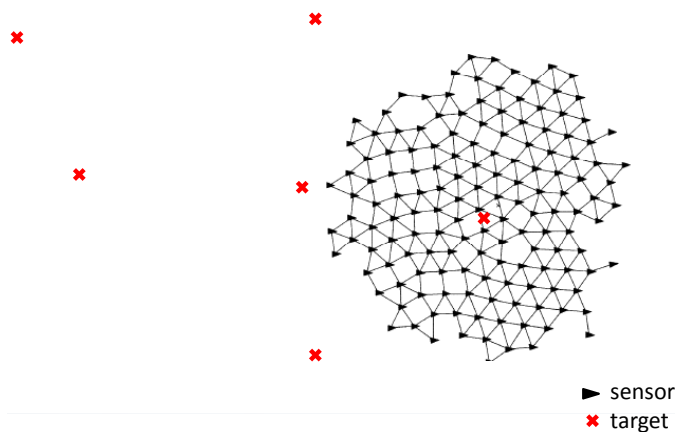


Figure 2.8: Flocking algorithm drawbacks in surveillance system

In addition, none of the sensors remains to search the surveillance area to detect new targets or events. This drawback decreases the area coverage and, as a result, increases the detection time for new targets. Therefore, the Flocking algorithm is not only unable to provide acceptable dynamic area coverage, but also cannot obtain reliable target coverage in a multi-target system.

2.4.2.1.2 Anti-Flocking Algorithm The Flocking algorithm is a well-known algorithm with sound mathematical foundation, and noticeable promising results when applied in target tracking. However, this algorithm has limitations with respect to a number of sensor management objectives, especially when applied in surveillance systems. This includes the objective that the sensors of the surveillance system must simultaneously achieve acceptable dynamic area coverage and reliable target coverage. This limitation is one of the main motivations for the Anti-Flocking algorithm [131].

The Anti-Flocking algorithm is based on three main rules [131]:

- De-centring: attempt to move apart from neighbours;
- Collision avoidance: stay away from the nearest obstacle that is within a safe distance;
- Selfishness: if neither of the above two situations happens, move on a direction that maximizes one's own gains.

Since the objective of the Anti-Flocking algorithm is to maintain high area coverage, the gain of a mobile sensor's optimal moving direction is set to be the area that has the longest time being unvisited [131]. The result of applying the Anti-Flocking algorithm in a surveillance application is illustrated in Figure 2.9. As depicted in this figure, and also by experimental results reported in [131], the Anti-Flocking approach achieves acceptable dynamic area coverage.

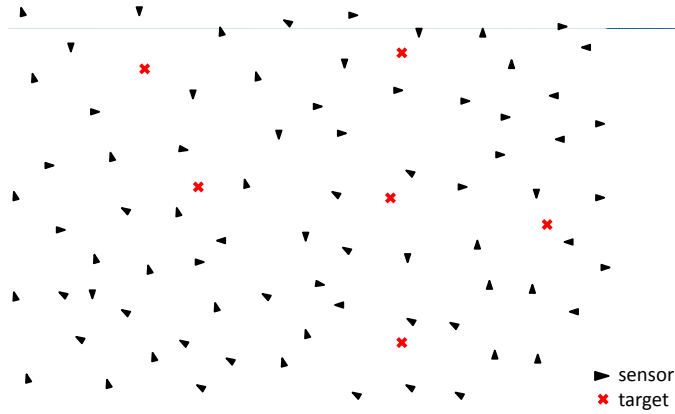


Figure 2.9: Anti-Flocking algorithm in surveillance application

Although the Anti-Flocking algorithm achieves acceptable dynamic area coverage, which is essential for surveillance applications, it does not assemble a mass of mobile sensors around each target. This behaviour is mostly due to the de-centring rule that scatters sensors over the whole volume of the monitoring area. The lack of an adequate number of sensors around a target in the Anti-Flocking algorithm greatly increases the network's chance of missing already detected targets. Furthermore, even if the sensor network is to employ a prediction algorithm (e.g., Kalman Filter [132]) to predict future positions of targets, the lack of adequate sensors flocking around the target could result in a poor prediction.

2.5 Summary

This chapter introduced sensor management, its challenges and opportunities. Then it focused on target engagement as one of the main parts of sensor management which involves the target-to-sensor allocation problem as well as coordination and control of sensors. In the second part various solution approaches that are proposed in literature are discussed including those for motion control of mobile sensors and those for target allocation. As DCOP is a powerful tool to deal with the target-to-sensor allocation problem and is selected in this research as a useful tool to tackle the target-to-sensor allocation problem, this

chapter reviewed the strategies using DCOP modelling in the target-to-sensor allocation problem in more detail.

Although DCSP/DCOP provides a formalized and general approach for the allocation problem in sensor networks, to date, all DCSP/DCOP modelling techniques presented to solve this problem have several limitations. Almost all of them consider a simplified allocation problem in sensor networks and fail to consider important characteristics of a real sensor network. These characteristics include:

- Dynamic sensors that can move freely through the environment in addition to static ones
- The possibility of a sensor failure
- Sensor networks with a large number of sensors

On the other hand, for the issue of sensor control and coordination in sensor networks, this research focused on biologically inspired approaches as they have gained significant attention in recent years. These approaches are exemplified by the two well-known algorithms, namely, the Flocking algorithm and the Anti-Flocking algorithm. Generally speaking, although these two biologically inspired algorithms have demonstrated promising performance, they expose deficiencies when it comes to their ability to maintain simultaneous reliable dynamic area coverage and target coverage. These two coverage performance objectives are inherently conflicting.

The next chapter formalizes the target engagement problem discussed in this chapter and then goes through the solution strategy of this research.

Chapter 3

Problem Formulation and Solution Strategy

Chapter 2 introduced the sensor-target engagement problem, discussed key issues that need further research and highlighted some important challenges these issues present. The chapter also presented some important methods that can be considered in solving this problem. Recognizing the promise of the well established DCSP/DCOP formulation to set the foundation of sensor-target engagement based solutions for solving the sensor-to-target problem was provided. This chapter presents a formal formulation of the target engagement problem and lays the foundation for the solutions strategy. The chapter is organized as follows: Section 3.1 formulates the target engagement problem, then Section 3.2 describes the solution strategy. Section 3.2.3 describes the use case adopted in this thesis and Section 3.3 summarizes and concludes the chapter.

3.1 Target Engagement

This research work is concerned with a sensor network deployed to perform surveillance operations. It is assumed that the network consists of multiple stationary and mobile sensors, each controlled as an independent agent. Initially, all sensors are positioned at

locations deemed to be appropriate for the surveillance task. Without loss of generality, these locations in the context of the proposed research can be considered arbitrary. Figure 3.1 depicts an example of the sensor network of a surveillance system that is used for monitoring a small size airport (e.g., Waterloo Regional Airport).

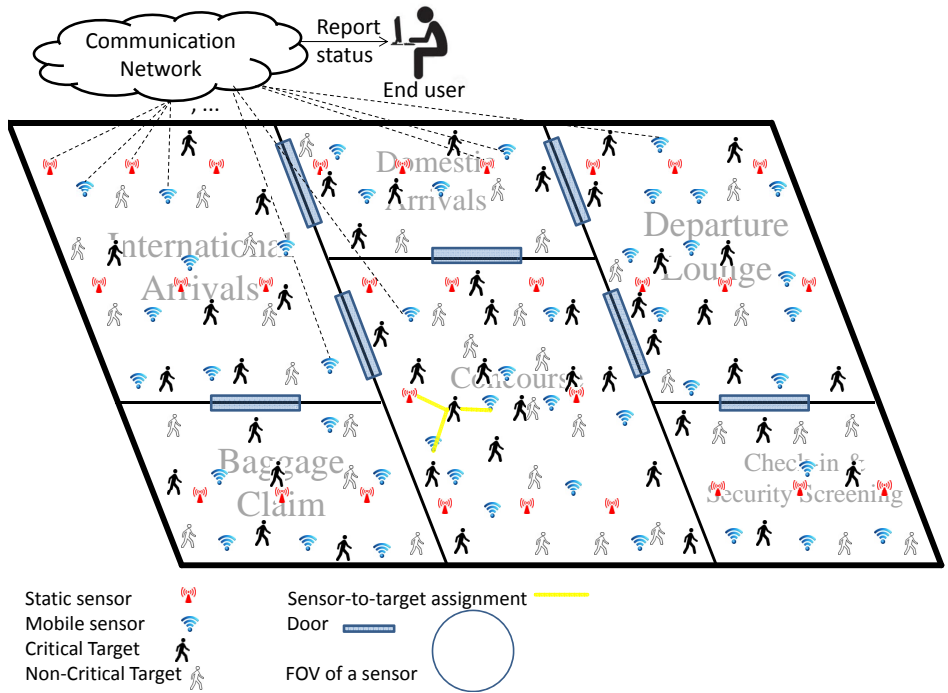


Figure 3.1: Sensor network of a surveillance application

The target engagement problem concerns managing a set of sensors to monitor a set of targets. The network plans the movements of its sensors to ensure that these targets are monitored and tracked continuously. Sensor movement planning aims for optimal performance with respect to target acquisition, coverage, real-time response, and power consumption, subject to a set of constraints. Two optimality formulations will be considered. In one formulation, optimality will be defined as maximizing (or minimizing) an objective function. In another formulation, optimality will be defined as minimizing constraint violation, for example, a maximum number of targets are covered at minimum constraints violation. Figure 3.2 depicts a taxonomy of the target engagement problem as a

twofold problem. As illustrated in this figure, the target engagement problem is an important and general problem which is concerned with two main sub-problems: target-to-sensor allocation and sensor control and coordination.

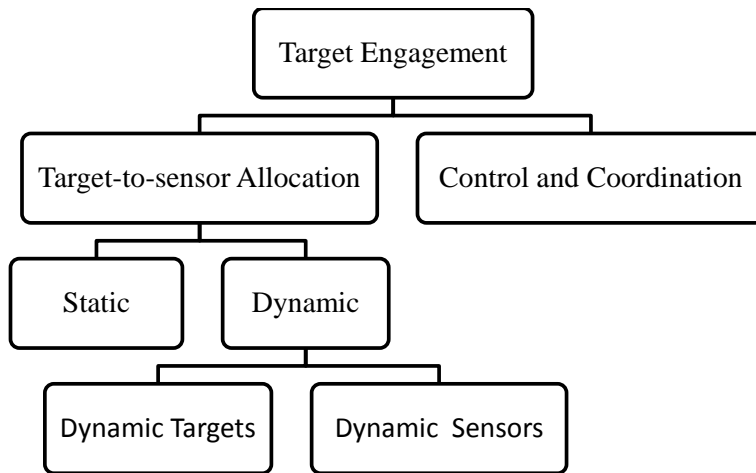


Figure 3.2: Target engagement problem

The goals of these two sub-problems are related and interconnected. The allocation problem is concerned with finding the best assignment of sensors to targets based on the current position of sensors. However, as the environment is considered to be dynamic (moving sensors and targets), a solution does not remain optimal. Therefore, the control and coordination problem addresses the continuous problems of positioning and control of the sensors, such that the positions and control parameters of the sensors, at any point in time, yield successful engagement.

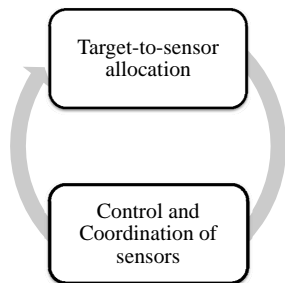


Figure 3.3: Allocation and control problems

Coordination is considered an embedded task within control. As illustrated in 3.3 the system cycles between two modes of operation: assignment and control.

As shown in Figure 3.2, the target-to-sensor allocation problem includes dynamic and static scenarios. In the static scenario, both the sensors and the targets are assumed to be static. The goal in this case is to find optimal assignment of targets to sensors. As the problem is static, this solution remains optimal as long as the sensors and targets remain static. The dynamic scenario includes mobile targets and mobile sensors. Assuming the network of sensors is static while the targets are mobile, the static sensors should find an optimal assignment subject to the constraints imposed by the positions of the sensors. Even if a rearrangement of the sensors would provide enhanced performance (for example, better coverage), this rearrangement cannot be contemplated due to sensor immobility. In the next case both the sensors and the targets are mobile but the movements of the sensors are ad-hoc and lack any objective motion control strategy. Although the mobile sensors, in this case, may end up in an assignment that can outperform the static sensors case, the performance of the network may be suboptimal in comparison to the case in which the sensors use an objective control strategy to find optimal tracks/positions.

In this problem it is assumed that each sensor has a field of view (FOV) of radius R and can sense any target within that field of view. The sensor should select one of the targets in its FOV for tracking (assuming a sensor can track one target only). The network sensors act as a team and coordinate among each other to find optimal movement plans and a sensor-to-target assignment. Each sensor can detect the distance to, and speed of the target, and can communicate with other sensors that are within distance x meters from it. The sensors can communicate information about their positions and the target(s) that are allocated to them. To summarize, the sensor network to be studied in this research has the following characteristics:

- The surveillance system consists of n mobile sensors deployed in a two-dimensional geographical region with width w and length l .
- Each sensor at each time can track only one target.
- At least k sensors are required to track a target completely.

- Each sensor can communicate with all its neighbouring sensors by exchanging messages through a communication network.
- The sensors that are within distance \mathbb{R} can communicate with one another.
- Each sensor can sense all the targets that are within distance R from its position, and thus the sensing range of each sensor is a circle with radius R centered at the sensor position. Targets that come within this range are always detected.
- The problem is dynamic and both sensors and targets can move through the environment.
- The motion of each sensor is controlled independently but in coordination with the motion of other sensors. Let $q_i, p_i \in \mathbb{R}^2$ denote the position and velocity of sensor i respectively. The motion of sensor i is governed by the following equation:

$$\begin{cases} \dot{q}_i = p_i \\ \dot{p}_i = u_i \end{cases} \quad \text{where } q_i, p_i, u_i \in \mathbb{R}^2 \quad (3.1)$$

- The surveillance system consists of m manoeuvring or non-manoevring mobile targets ($n > m$) randomly entering and leaving the area of interest (AOI). Let $q_{tj}, p_{tj} \in \mathbb{R}^2$ denote the position and velocity of target j respectively. All targets follow the following equation of motion:

$$\begin{cases} \dot{q}_{tj} = p_{tj} \\ \dot{p}_{tj} = u_{tj} \end{cases} \quad \text{where } q_{tj}, p_{tj}, u_{tj} \in \mathbb{R}^2 \quad (3.2)$$

If target j is a non-manoevring target then $u_{tj} = 0$

- Each sensor observes the state of the targets in its environment using a measurement system:

$$z_i = H_i x + v_i \quad i = 1, 2, \dots, n; \quad z_i \in \mathbb{R}^2 \quad (3.3)$$

where, z_i is the measurement of sensor i about the state of the target, H_i is a time-varying matrix. This matrix is the observation model which maps the true state space into the observed space. v_i is the observation noise which is assumed to be zero mean Gaussian noise.

- Sensors' knowledge of targets is limited to targets positions and velocities.
- Sensors have low communication and computation abilities.
- Targets can join or leave the problem.
- Sensors may fail or malfunction.

The dynamicity of the problem makes it hard but interesting as the optimality of an assignment of sensors to targets may become invalid and as such the network needs to maintain closed-loop engagement to ensure reliable optimality performance. The problem can be formulated as follows:

- A network of sensors: $S = \{s_1, \dots, s_n\}$, including both static and mobile sensors
- A set of targets: $T = \{T_1, \dots, T_m\}$, including both static and mobile targets
- Position of S at time t : $P_S(t) = \{P_{s_1}(t), \dots, P_{s_n}(t)\}$ where $P_{s_i}(t) \in \mathbb{R}^2$
- Position of T at time t : $P_T(t) = \{P_{T_1}(t), \dots, P_{T_m}(t)\}$ where $P_{T_i}(t) \in \mathbb{R}^2$
- $FOV_i(t)$ is a circle with radius R around $s_i \in S$ at time t
- $H_i(t) = \{T_x, \dots, T_y\}, x, y \in [1, m]$ is an assignment of targets to sensor s_i in time t
- $G_i(t) = \{s_x, \dots, s_y\}, x, y \in [1, n]$ is an assignment of sensors to target T_i in time t
- $(H(t) = \bigcup_{i=1}^n H_i(t)) \equiv (G(t) = \bigcup_{i=1}^m G_i(t))$ is an assignment of targets to sensors
- $U(t) = \{U_{s_1}(t), \dots, U_{s_n}(t)\}$ is a set of motion controls for sensors in S in which each $U_{s_i}(t) : P_{s_i}(t) \rightarrow P_{s_i}(t+1)$ is a function that changes the position of s_i in time t

- $d(a, b)$: the Euclidean distance between two points in \mathbb{R}^2

The goal is to find a set of suitable motion controls $U = \{U(0), \dots, U(t_f)\}$ and a set of minimum cost assignment of sensors to targets $G_{opt} = \{G_{opt}(0), \dots, G_{opt}(t_f)\}$ such that for each time $0 \leq t \leq t_f$

$$G_{opt}(t) = \text{Minimize} \left(\sum_{i=1}^n \text{SCost}(H_i(t)) + \sum_{i=1}^m \text{TCost}(G_i(t)) \right) \quad (3.4)$$

Subject to:

$$\forall_{i:1,\dots,m} \quad \forall s_j \in G_i(t) \quad d(P_{T_i}(t), P_{s_i}(t)) \leq R \quad (3.5)$$

$$\forall_{i:1,\dots,n} \quad \forall T_j \in H_i(t) \quad d(P_{T_j}(t), P_{s_i}(t)) \leq R \quad (3.6)$$

$$\forall_{i:1,\dots,m} \quad |G_i(t)| \geq k \quad (3.7)$$

$$\forall_{i:1,\dots,n} \quad |H_i(t)| \leq 1 \quad (3.8)$$

where:

$$\text{SCost}(H_i(t)) = \begin{cases} 1 & |H_i(t)| > 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{TCost}(G_i(t)) = \begin{cases} 1 & |G_i(t)| < k \\ 0 & \text{otherwise} \end{cases}$$

Where, SCost is the cost of the first constraint and TCost is the cost of the second constraint. As H_i and G_i are equivalent at time t , two constraint 3.5 and 3.6 are equivalent as well, and show that only targets that are in the FOV of each sensor can be assigned to that sensor. The next two constraints (3.7 and 3.8) show the requirement of participation of at least k sensor to track a target and the ability of each sensor to track just one target at each moment of time respectively.

3.2 Solution Strategy

The goal of this research is to develop an efficient, general and scalable approach for the target engagement problem in complex and large-scale sensor networks. To address this problem, as illustrated in Figure 3.2, the main problem is divided into sub-problems: Target-to-sensor allocation and sensor motion control and coordination. This research will address these problems in a modular manner. Figure 3.4 summarizes the solution strategy as a multi-step process. Solution strategies selected for each step are explained in more detail in the following sections.

3.2.1 DCOP Modelling for Target-to-sensor Allocation

To provide a general solution for the target-to-sensor allocation problem, the DCOP modeling approach is selected. This modelling approach provides a formal and general view of the problem. The generality of this method makes it suitable to tackle a wide range of sensor networks problems. Table 3.1 shows a mapping between the target-to-sensor allocation problem and DCOP.

Table 3.1: Mapping target-to-sensor allocation problem to DCOP

target-to-sensor allocation problem	DCOP
Sensors/ targets	Variables
Targets in FOV of sensor/ Sensors in FOV of target	Domain of Variable
$\forall i:1,\dots,m \quad \forall s_j \in G_i \rightarrow t_i \text{ is within } FOV(s_j)$ $\forall i:1,\dots,n \quad \forall t_j \in H_i \rightarrow t_j \text{ is within } FOV(s_i)$ $\forall i:1,\dots,m \quad G_i \geq k$ $\forall i:1,\dots,n \quad H_i \leq 1$	Constraints
Goal: $Minimize (\sum_{i=1}^n SCost(H_i) + \sum_{i=1}^m TCost(G_i))$	Goal: Minimize the cost of violated constraints

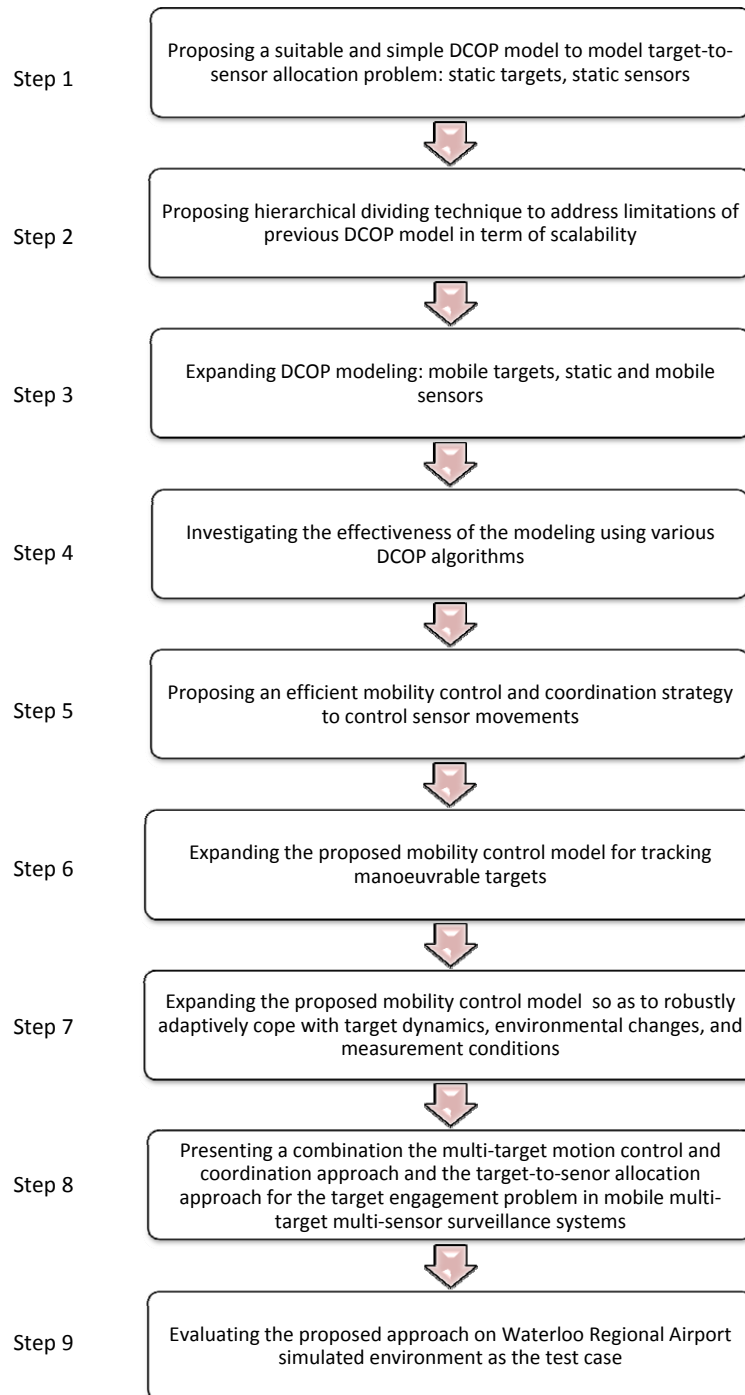


Figure 3.4: Solution steps for solving target engagement problem

In the first scenario both the targets and the sensors are considered to be static. Various dynamicity aspects are then introduced to the initial DCOP to investigate network performance in response to target and sensor mobility considerations. Although DCOP modelling provides a general and formal view of the problem, their scalability and dynamicity issues should be carefully investigated.

3.2.1.1 Scalability of the DCOP Model

To deal with the scalability issue in the target-to-sensor allocation problem, I divide the coverage area of the surveillance problem into smaller interconnected partitions. This partitioning process is based on the geographical locations of the sensors and the environment they monitor. Sensors that are geographically far apart may have only few conflicts with one another. Although the geographical regions of the problem are not entirely separable, only a few constraints exist between them. These facts point to the potential merit that partitioning can lead to significant enhancement to the solution process. Such partitioning yields a set of separate DCOPs. This partitioning process is recursively applied to the sub-partitions, yielding a hierarchy of DCOPs of manageable sizes. As the main problem is NP-hard, increasing the number of variables increases the complexity of the problem exponentially. Therefore, using current methods in large-scale sensor networks that consists of a large number of sensors, introduces high computational and communicational costs. Partitioning can have a great impact on reducing the communication and computation overheads, especially in large-scale sensor networks.

3.2.1.2 Dynamicity of the DCOP Model

As illustrated in Figure 3.2, dealing with the dynamicity issue has several aspects, including: mobile targets, mobile sensor and controlling the movement of sensors. In addition, the possibility of a target entering the FOV at any time or the possibility of a target leaving the FOV, are some realistic scenarios that can be considered as an aspect of dynamicity due to target mobility. On the other hand, the possibility of sensor failure and the possibility of new sensors joining the sensor network can also be considered as an aspect of dynamicity due to sensor mobility.

This research plans to tackle the dynamicity issue of the problem in phases. In the first phase both the targets and the sensors are considered static. In this case the stationary problem is modelled as a DCOP, and is solved using an appropriate DCOP algorithm. In the second scenario the targets can move around arbitrarily. In this case the problem is divided into predefined time intervals. In each time interval, the problem is considered a DCOP instance, and is solved using a suitable DCOP algorithm. In the third scenario, both the sensor and the targets move around, but the movement of the sensors in this case is assumed to be random, and is not controlled. In this scenario, the impact of the presence of mobile sensors is compared with the static sensors scenario. The last and the most important scenario of dynamicity is the case where a control strategy is put into play to manage the movement of the sensors so as to achieve optimum performance.

After providing a general and scalable modelling approach for the target-to-sensor allocation problem, a suitable DCOP solver should be selected to solve the modelled problem. There are several DCOP algorithms with various characteristics. The DCOP solver selection should be based on the requirements of the original target-to-sensor allocation problem. For example, the selected algorithm should be able to provide quick responses even if they are not the best solutions. As targets move around the optimal solution may not work after a while. This shows the requirement of providing fast and acceptable (even not optimal but near to optimal) solutions. On the other hand the algorithm should not impose high computation and communication overhead on the sensor nodes because as described before, sensor nodes have low computation and communication abilities. In addition, the selected algorithm should not require high memory on sensor nodes because it cannot be provided by most of the sensor networks. Considering the requirements of the problem, it seems that incomplete DCOP algorithms are better choices. The next chapter discusses this point in more detail.

The strategy used to tackle the target-to-sensor allocation problem is able to find the optimal or near optimal solution for each problem instance. However, this solution is optimal given the current position of the sensors and considering their restriction in moving around. In other words, the solution may not be optimal if one assumes the sensors can move around to find better positions to improve engagement performance. This point stresses the importance of using motion control strategies to control the position and

motion of sensors so as to optimize network performance.

3.2.2 Flocking-based Algorithm for Control and Coordination

This work aims to use a biologically inspired approach to find better solutions. One of the most interesting aspects of dynamicity in sensor networks is the ability of sensors to move around. This ability is exploited to seek optimal assignment and positioning schemes to optimize engagement performance. Biologically inspired approaches have recently gained significant attention as a tool to address the issue of sensor control and coordination in sensor networks. These approaches are exemplified by the two well-known algorithms, namely, the Flocking algorithm and the Anti-Flocking algorithm. Generally speaking, although these two biologically inspired algorithms have demonstrated promising performance, they expose deficiencies when it comes to their ability to maintain simultaneous reliable dynamic area coverage and target coverage. These two coverage performance objectives are inherently conflicting.

This research proposes a biologically inspired algorithm that benefits from key characteristics of both the Flocking and Anti-Flocking algorithms. This algorithm strikes a balance between reliable area coverage and target coverage. This balance is facilitated via flock-sensor coordination.

The issue of control and coordination is even more challenging when sensors are dealing with manoeuvrable targets that change their speed and direction frequently or suddenly. Most of the surveillance applications require accurate tracking of manoeuvrable targets such as pedestrians, animals, vehicles, aircraft, *etc.* Many of the algorithms that show promising results in tracking non-manoeuving targets, fall into trouble when it comes to their ability to track manoeuvring targets. This work expands the primary proposed mobility control model for tracking manoeuvrable targets.

In the next step, this research expands the mobility control algorithm to reliably adaptively cope with target dynamics, environmental changes, and measurement conditions. In most of the surveillance applications the sensory information is inherently noisy. Therefore, it is imperative that this noise is reduced. To address the problem of reliable multiple target

tracking using a sensor network, this research proposes a coupled distributed estimation and motion control approach.

To ensure the timely tracking of mobile targets, the surveillance sensor network needs to keep continuous engagement with all targets in its working area. Thus, the network must be able to perform the following tasks in real-time: 1) target-to-sensor allocation; 2) sensor mobility control and coordination. As the last step, this research aims to present a combination of the target-to-sensor allocation mechanism proposed in Step 3 and the multi-target motion control and coordination approach presented in Step 5 to tackle the target engagement problem in mobile multi-target multi-sensor surveillance systems.

3.2.3 Airport Security as a Test Case

Border control deals with the challenge of controlling the movement of people, animals and goods into, as well as, out of a country. Effective border control has become an increasing challenge over the past few decades. Faced with enormous political pressure to stop not only illegal immigration and to prevent the entry of potential terrorists, but also more serious issues such as incoming weapons of mass destruction, governments have devoted even more resources to enforcing border policies. Most countries have thousands of kilometres of blue (sea side) or green (land side) borders between each other. For the purposes of border control, in addition to the border crossing points and seaports, airports are also classed as borders.

Airports, as locations with facilities for commercial aviation flights, have a significant contribution in international travels. They consist of buildings and airfields that serve to house planes and runways for airplane take-off and landing. A large number of people pass through airports every day. Due to the large number of people located in a particular location, airports present a great potential for terrorism and other forms of crime. Furthermore, the high concentration of people on large airliners especially international airports, the ability to use a hijacked air plane as a lethal weapon, and the potential high death rate with attacks on aircraft, can provide an attractive target for terrorism.

Airport security aims to prevent any threat or harm to aircrafts, passengers, staff,

and airport infrastructure. Airport security also supports national security and counter-terrorism policy. Airports are extremely busy public places and, unfortunately, so they are prime targets for terrorism. Therefore, airport defence really does present a labyrinth of security challenges. Moreover, the need of real-time situational awareness, data fusion from different types of sensors and statistics collection in case of an event, impose further challenges on the problem.

This research presents airport security as a case study for the target engagement problem. The scenario adopted in this work is the surveillance of the Waterloo International Airport. Figure 3.5 illustrates the layout of the airport.

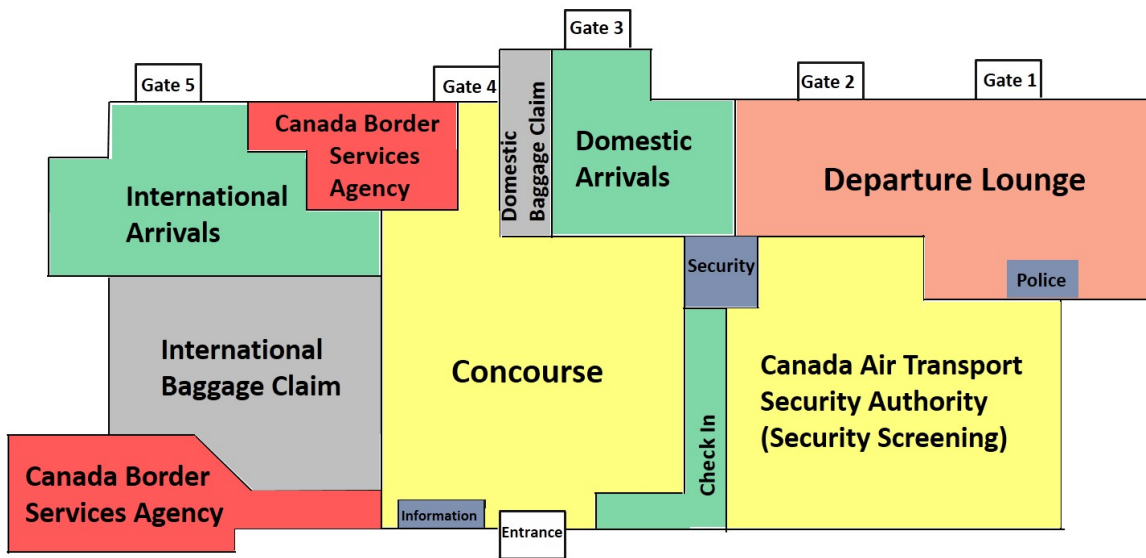


Figure 3.5: The Waterloo Region International Airport

The airport halls are virtually divided into a mesh of grid cells. This environment is simulated using a Java development as represented in Figure 3.6. Two types of sensors: static and mobile, are represented by red and blue icons respectively and two types of targets: critical and non-critical, by black and white pedestrian icons as shown in Figure 3.6. The passengers enter, leave and move around in the airport. The goal is finding the best assignment of mobile or static sensors to targets so that a minimum cost of constraints are violated. The mobile sensors can move around in the environment and can go from

one section to another.

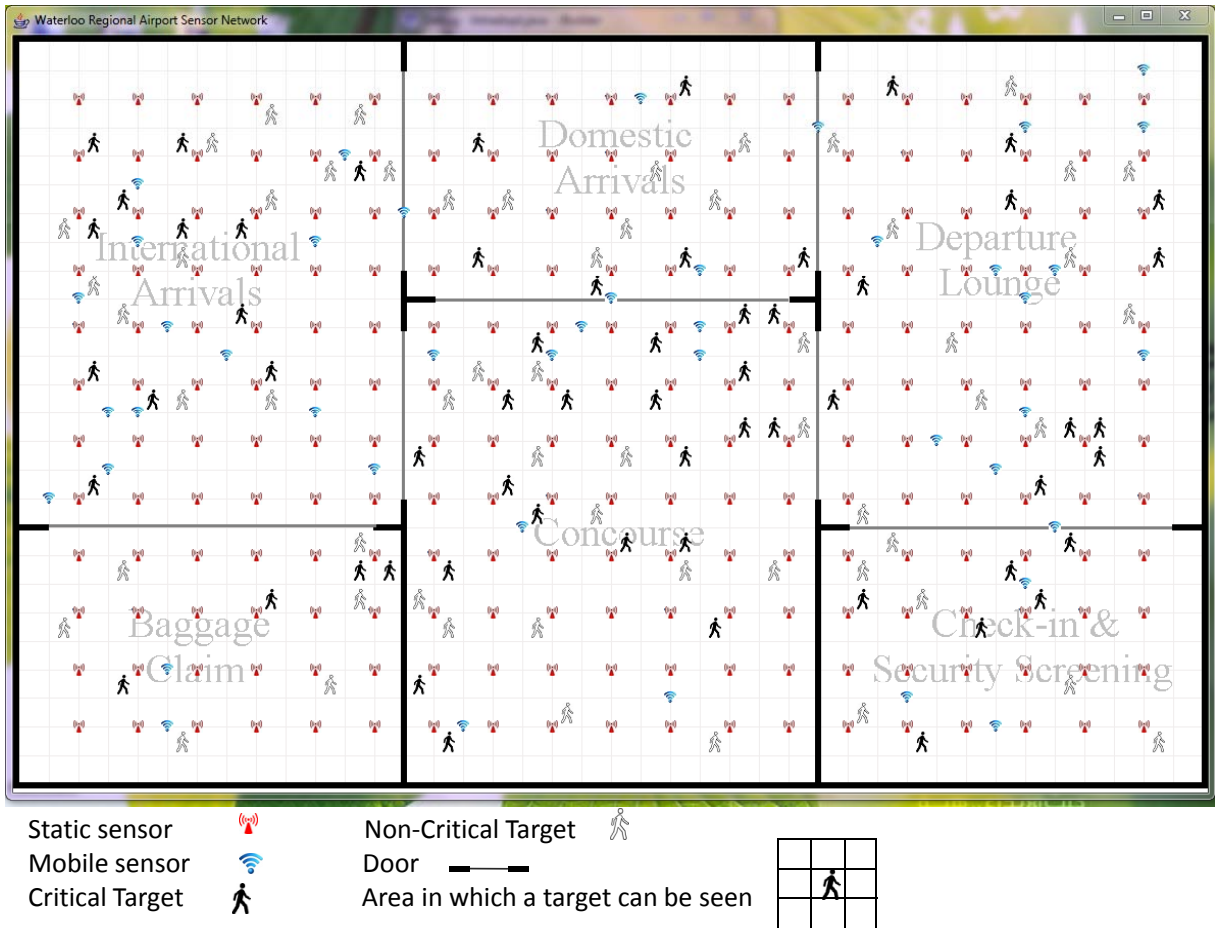


Figure 3.6: Waterloo Region Airport simulation environment in Java

3.3 Summary

The target engagement problem is defined and formulated in this chapter. The solution strategy of this research presented as a multi-phase. The DCOP modelling and flocking-based approaches, the solution strategy, and the method for evaluating the proposed solution strategy are presented. Various aspects of the solution process, such as generality,

scalability and dynamicity, are discussed. The next chapter presents the proposed solution of the target engagement problem focusing on the target-to-sensor allocation. It will also discuss considerations for the sensor control and coordination problem.

Chapter 4

Target to Sensor Allocation: A Hierarchical Dynamic Distributed Constraint Optimization Approach

This chapter introduces a novel approach to solve the target-to-sensor allocation problem by modelling the problem as a Hierarchical Distributed Constraint Optimization Problem (HDCOP) . Distributed Constraint Optimization Problems (DCOP's) tend to be computationally expensive and often intractable particularly in large problem spaces such as Wireless Sensor Networks (WSNs). To address this challenge I propose changing the sensor to target allocation to a hierarchical set of smaller DCOPs with a shared system of constraints. Thus, we avoid significant computational and communication costs. Furthermore, in contrast to other DCOP modelling methods, a non-binary variable modelling is employed to reduce the number of intra-agent constraints. The chapter is organized as follows: Section 4.1 presents a brief introduction and discusses the motivation for the work presented in this chapter. Section 4.2, formulates the target to sensor allocation problem. Section 4.3 describes the proposed hierarchical modelling technique. The hierarchical architecture, the non-binary variable modelling, the combination of the hierarchical architecture and the non-binary variable modelling, their formal representation and a pseudo code for the proposed approach, are described from subsections 4.3.1 to 4.3.4. Section 4.4 shows a

taxonomy of popular DCOP algorithms and, finally, Section 4.5 reports work conducted to test the proposed hierarchical approach in the context of two important DCOP algorithms, DBA and ADOPT. ADOPT and DBA are also used to solve the same problem in its simple non-hierarchical form previously introduced by [75]. Algorithm performance is compared with respect to three parameters: “constraint violation cost”, “number of messages” and “number of constraint checks”. Conclusions and future work are provided in Section 4.6.

4.1 Introduction

Distributed Constraint Optimization Problems (DCOPs) provide a good formalism for representing problems where each agent is responsible for assigning a value from its finite domain to its variable. These agents coordinate their choice of values so as to find value assignments that minimize the constraint cost. Two agents are constrained if they share a constraint. Each constraint cost is a function of the values of the involved variables. The cost of a solution is the sum of all the constraint costs, depending on the values assigned to the variables in the solution. The goal of solving a DCOP, which is an NP-hard problem [4], is to find an assignment of values to all variables that minimizes the constraints’ cost. DCOP is a generalized form of the Distributed Constraint Satisfaction Problem (DCSP) [83, 84]. Many real-world problems can be modelled as DCOPs. Resource allocation in sensor networks [42, 75], event scheduling [124, 138], and synchronization of traffic lights [139] are examples of these real world problems.

Sensor to target allocation is an important problem in many sensing applications, including surveillance, traffic management, and habitat monitoring, to name a few. In this problem there are a set of moving targets and a set of sensors, each sensor having limited sensing, computational, and communication capabilities. The objective is to find a suitable assignment of sensors to targets such that the number of constraint violations is minimized. Constrains can be used to represent restrictions on sensors, for example, to set the maximum number of sensors to track a target, and/or the maximum number of targets to be tracked by one sensor. The problem setup changes over time due to sensor and/or target mobility. In order to relax the computational complexity of the problem, it is divided into

instances, and the solution of each is assigned to a solver. In this chapter, the target to sensor allocation problem in a complex sensor network is modelled as a hierarchical and dynamic DCOP. Most of the sensor networks, such as those employed in surveillance systems or in military systems, tend to deal with large and complex sensor networks that are composed of many sensors and numerous dynamic targets. This complexity should be carefully considered in any modelling technique contemplated for addressing the problem. This chapter, unlike previous DCSP/DCOP target allocation problems [124, 86, 89, 92, 93, 94], first divides the main problem into several DCOP sub-problems, which themselves are divided further into smaller DCOP sub-problems. At each level, each DCOP employs non-binary variables to construct a model of the problem. Constraints that are shared by different sub-problems define another DCOP at the next level of the proposed hierarchical structure.

4.2 Target-to-Sensor Allocation Problem: the Modelling

The target-to-sensor allocation problem in a sensor network is concerned with finding a suitable assignment of a network of sensors to a set of targets such that, a certain objective function is optimized subject to some constraints. Another formulation of the problem aims at achieving the objective function with minimum constraint violations. In this research a network of sensors that are subject to the following constraints is considered:

- Each sensor at each time can track only one target,
- At least k sensors are required to track a target completely,
- Each sensor can observe only targets placed within its field of view (FOV).

This problem is formalized as follows:

- A network of sensors: $S = \{s_1, \dots, s_n\}$

- A set of targets: $T = \{t_1, \dots, t_m\}$
- $FOV_{i,t}$ is the field of view area for sensor s_i in S , at time t .
- $H_{i,t} = \{t_x, \dots, t_y\}, x, y \in [1, m], H_{i,t} \subseteq T$ is an assignment of targets to sensor s_i .
- $G_{i,t} = \{s_x, \dots, s_y\}, x, y \in [1, n], G_{i,t} \subseteq S$ is an assignment of sensors to target t_i .
- Domain $(H_{i,t}) =$ all the subsets of $\{T_1, \dots, T_m\}, i = 1, \dots, n$.
- Domain $(G_{i,t}) =$ all the subsets of $\{S_1, \dots, S_n\}, i = 1, \dots, m$.
- $(H = \bigcup_{i=1}^n H_i)$ is the assignment of all targets to all sensors from the point of view of sensors.
- $(G = \bigcup_{i=1}^m G_i)$ is the assignment of all targets to all sensors from the point of view of targets.
- H and G refer to the same assignment A , but from different points of views, where A is a set of $\langle target\ T, sensor\ S \rangle$ and a pair $\langle target\ T, sensor\ S \rangle$ implies that target T is assigned to sensor S .

The goal is to minimize the assignment cost of sensors to targets:

$$G_{opt,t} = Minimize \left(\sum_{i=1}^n SCost(H_{i,t}) + \sum_{i=1}^m TCost(G_{i,t}) \right) \quad (4.1)$$

Subject to :

$$\forall t_i \wedge \forall s_j \in G_{i,t} \wedge t_i \text{ is within } FOV_{j,t}$$

$$\forall s_i \wedge \forall t_j \in H_{i,t} \wedge t_j \text{ is within } FOV_{i,t}$$

$$t_j \in H_{i,t} \Leftrightarrow s_i \in G_{j,t}$$

where:

$$SCost(H_{i,t}) = \begin{cases} 1 & |H_{i,t}| > 1 \\ 0 & otherwise \end{cases}$$

$$TCost(G_{i,t}) = \begin{cases} 1 & |G_{i,t}| < k \\ 0 & otherwise \end{cases}$$

SCost is the cost of a constraint violation due to assigning one sensor to more than one target. TCost is the cost of a constraint violation due to assigning less than k sensors to each target.

As the problem is dynamic and sensors and targets are moving around, the assignment parameters change over time. Of course, this problem can be framed as a constraint satisfaction problem and the objective is to find a solution in which no constraints are violated, *i.e.*, zero constraints cost.

4.3 Hierarchical target-to-Sensor: Non-Binary Variable DCOP Representation

The DCSP/DCOP modelling approach, unlike other methods, provides a clear and general mathematical view of the problem. As such, solution techniques that are based on DCSP/DCOP are applicable in various sensor network application domains.

Although DCOP provides a general modelling approach, current DCSP/DCOP modelling techniques in sensor network applications suffer from critical and impractical simplifications. These limitations include small network assumptions (e.g, a small number of sensors), static sensor assumptions whereby sensors cannot pursue mobile targets, impractical constraints on the number of targets in the FOV of each sensor and, in some cases, restrictions on target mobility (*i.e.*, a static targets assumption). Furthermore, scalability as an important solution aspect has not been considered. As DCSP/DCOP problems are NP-hard [4], raising the number of variables increases the complexity of the problem exponentially. Therefore, using current methods in large-scale sensor networks that consist of a

large number of sensors introduces high computational and communicational costs to the DCSP/DCOP algorithm. Moreover, such methods do not take advantage of the fact that sensors that are geographically far apart may have only a few conflicts with one another. Although the geographical regions of the problem are not entirely separable, only a few constraints exist between them. These facts point to the potential merit that partitioning can lead to significant enhancement of the solution process.

In this research, I present a technique to overcome the limitations of current DCSP/DCOP modelling methods and to eliminate impractical simplifications in such methods. I address the problem in two steps. First, I partition the complex target allocation problem into hierarchically smaller problems. This partitioning is based on the geographical locations of the sensors and the environment they monitor. Such partitioning results in a set of distinct DCOPs. This partitioning strategy is recursively applied to the sub-partitions, in order to have a hierarchy of DCOPs of manageable sizes.

Second, existing DCOP techniques [87, 93, 94], usually represent sensors and/or targets as binary variables. Representation of sensors and/or targets as binary variables increases the number of variables and constraints in the DCSP formulation, leading to a larger and more complex DCSP/DCOP. This could be prohibitive, particularly, in large sensor/target problems. I propose to represent sensors/targets as non-binary variables yielding smaller and more tractable DCOPs.

4.3.1 Hierarchical Architecture

In the proposed method, I first partition the problem into distinct geographical regions. Defining each region is important and has considerable impact on the performance of the solution; the regions are constructed such that they have the maximum intra-region constraints and minimum inter-region constraints. Figure 4.1 depicts this idea. In the lowest level of the proposed hierarchical architecture, the main problem is partitioned into various regions: Region 0, Region 1, *etc.* Each regions is tackled as a DCOP.

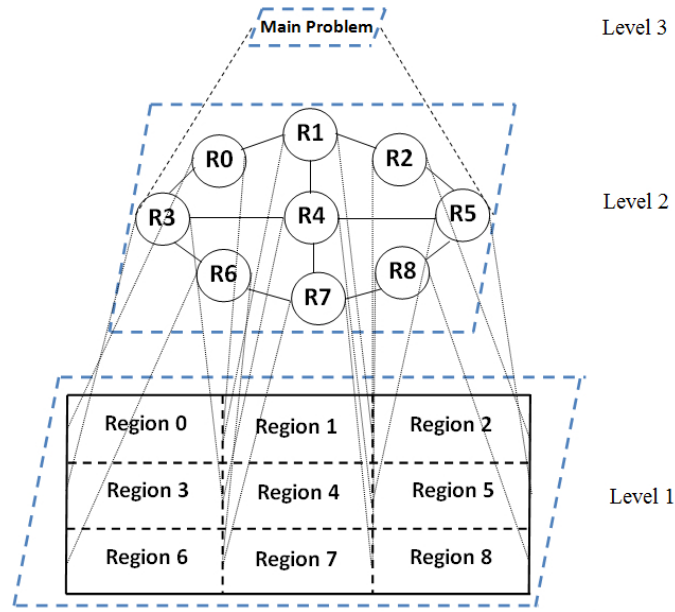


Figure 4.1: Hierarchical target-to-sensor allocation model for large and complex sensor networks

There are several DCOPs on the lowest level. The next layer of the hierarchy can be tackled as another DCOP that encompasses variables and constraints shared among the connected regions. DCOPs at this multi-region layer do not maintain a view of each region’s internal constraints. In fact, only the constraints that exist between different regions are maintained in this layer. Region specific constraints are managed at a lower layer. Problem solving using this architecture uses a top down approach. Therefore, each lower level’s solver starts after ending its upper level solver. More formally, considering the lowest level to be equal to 1, solving the problem (s) in level “ L ” is started exactly after solving the problem in level “ $L + 1$ ”. First, the problem of the upper layer is modelled as a DCOP, which is subsequently solved using a DCOP algorithm. After finding an optimal solution for this problem, the lower level’s DCOP solvers start solving each region’s problem separately. There are common variables between the upper level DCOP and lower level DCOPs. In solving the lower level problems, the algorithm is not concerned with the values maintained by the upper level. Section 5.3 discusses how DCOP modelling is realized in each layer for each sub-problem.

Using this method prevents the creation of large DCOPs. It is mathematically conceivable that partitioning a large exponential problem into several smaller exponential complexity problems can decrease the computational complexity greatly. Equation 4.2 shows this fact.

$$\sum_{i=1}^n a^{p_i} \ll a^{(p_1+p_2+\dots+p_n)} \quad \forall a > 1, p_i > 1 \quad (4.2)$$

In this Equation the notation a^{p_i} shows the exponential complexity of a problem with size p_i . Also, $(p_1 + p_2 + \dots + p_n)$ shows the size of a non-partitioned problem whose size is equal to the summation of the sizes of all p_i s ($1 \leq i \leq n$). It is mathematically provable that if a and p_i s are larger than 1, then the left-hand side of this equation showing the complexity of a partitioned problem is much smaller than the right hand side of the equation which shows the complexity of a non-partitioned problem. Of course, this fact is true for the problems with exponential complexity.

Obviously, the larger and more complicated problems need a higher number of layers and regions.

4.3.2 Non-Binary Variable Modelling

To model a problem as a DCOP, three main elements of the DCOP must be defined: the Variables, the Domain of each variable, and the Constraints. In Binary modelling that can be found in [87] and [94], for every sensor-target combination, a binary variable has been defined, so, the number of required variables and constraints to solve the problem tends to be large.

In this research, I propose a modelling technique in which each sensor is represented as a non-binary variable. All targets within the FOV of a sensor construct its domain. Table 4.3 shows an example of this modelling for a small sensor network with 12 sensors and 4 targets which is illustrated in Figure 4.2.

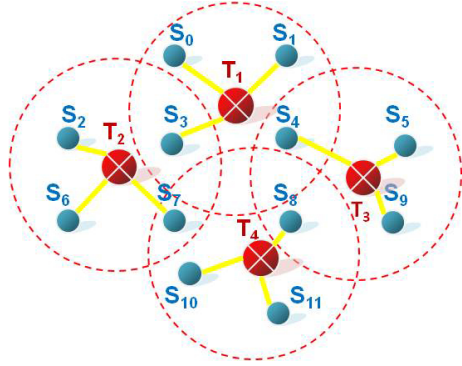


Figure 4.2: A typical sensor network with 12 sensors and 4 targets

Figure 4.3: Non-binary variable DCOP modelling of the sensor network of Figure 4.2

Variables	Domain	Constraints	
S_0	$\{T_1\}$	Constraint	Cost
S_1	$\{T_1\}$	Constraint $S_3 S_4$	1
S_2	$\{T_2\}$	Nogood $T_2 T_3$	
S_3	$\{T_1, T_2\}$		
S_4	$\{T_1, T_2\}$	Constraint	Cost
S_5	$\{T_3\}$	Constraint $S_3 S_7$	1
S_6	$\{T_2\}$	Nogood $T_1 T_4$	
S_7	$\{T_2, T_4\}$		
S_8	$\{T_4, T_3\}$	Constraint	Cost
S_9	$\{T_3\}$	Constraint $S_4 S_8$	1
S_{10}	$\{T_4\}$	Nogood $T_1 T_4$	
S_{11}	$\{T_4\}$		

In this example, each sensor is modelled as a non-binary variable in DCOP. Therefore, this specific problem has 12 variables because the problem has 12 sensors. All the targets within the FOV of each sensor represent its domain. The second column of Table 4.3 shows

these domain values for each sensor. The modelling technique uses non-binary variables; therefore, the domains are not restricted to zero/one values. Each target represented in the second column can be assigned to the corresponding sensor in the first column by the algorithm used to solve the problem. The third column shows the constraints that arise from the fact that every target must be tracked by at least three sensors, and, each sensor can select only one target from its domain. Each constraint contains one or more “*Nogood*” values and a “Cost”. The “Cost” illustrates the cost of the violation of a constraint. A constraint is violated if one of the “*Nogood*” pair values is selected for the pair of sensors that shares the constraint. For the presented example in Figure 4.2, S_3 and S_4 share a constraint (with the cost 1) in which (T_2, T_3) is a “*Nogood*” value (the first constraint in Table 4.3). Thus, the cost of assigning targets (T_2, T_3) to sensors (S_3, S_4) , respectively, is 1. This constraint arises from the fact that at least three sensors are required to track T_1 . Therefore, in addition to S_0 and S_1 , one more sensor is required. Assignment of S_3 to T_2 and simultaneously S_4 to T_3 limits the number of available sensors to track T_1 , which is defined as a “*Nogood*” for a constraint between S_3 and S_4 . The other constraints are defined in a similar way. Modelling the same problem using previous binary variable techniques [87] requires 16 variables. Each variable is a combination of a sensor and a target within its FOV. In this example the variables are: $S_{0,1}, S_{1,1}, S_{2,2}, S_{3,1}, S_{3,2}, S_{4,1}, S_{4,3}, S_{5,3}, S_{6,2}, S_{7,2}, S_{1,4}, S_{8,3}, S_{8,4}, S_{9,3}, S_{10,4}, S_{11,4}$. $S_{i,j}$ denotes the combination of sensor S_i and target T_j .

The difference between the number of variables in binary and non-binary modelling techniques increases in more complex problems in which more targets are common between the FOV of sensors.

The non-binary modelling technique reduces the number of variables and constraints. Constraint reduction is because of two facts. First, the variable definition inherently captures some of the constraints. The non-binary modelling technique applies an implicit constraint in assigning only one target to each sensor, because using this modelling, only one value (target) can be assigned to each variable (sensor) at each time. As a result, the number of required explicit constraints to model the problem reduces. Second, reducing the number of variables by itself leads to less number of constraints. Therefore the only remaining constraint is the number of communicating sensors (k) for full coverage in target tracking.

Table 4.1: Mapping between each part of the sensor network to a DCOP

part of the sensor network	DCOP
Sensors/Targets	Variables
Targets/Sensors that are placed in the FOV of a sensor/target	Domain of a Variable
$\forall_{i:1,\dots,m} \quad \forall s_j \in G_i \rightarrow t_i \text{ is within } FOV(s_j)$ $\forall_{i:1,\dots,n} \quad \forall t_j \in H_i \rightarrow t_j \text{ is within } FOV(s_i)$ $\forall_{i:1,\dots,m} \quad G_i \geq k$ $\forall_{i:1,\dots,n} \quad H_i \leq 1$	Constraints
<p>Goal:</p> $\text{Minimize } (\sum_{i=1}^n SCost(H_i) + \sum_{i=1}^m TCost(G_i))$ $SCost(H_i) = \begin{cases} 1 & H_i > 1 \\ 0 & \text{otherwise} \end{cases} \quad TCost(G_i) = \begin{cases} 1 & G_i < k \\ 0 & \text{otherwise} \end{cases}$	<p>Goal:</p> <p>Minimize the cost of violated constraints</p>

It is important to note that the proposed non-binary modelling approach can be extended to the target-centered case, in which targets are represented as non-binary variables, and all the sensors that have that specific target within their FOV construct that variable's domain. In this case, targets have constraints on the sensors that are common between their domains. Typically, systems possess some degree of control on the sensors but not on the targets. For example, in the selected test case, the airport will employ a set of sensors to monitor targets in selected areas. The targets in this case can be people. Therefore, if one is to adopt a target-centered engagement, controlling the value of a variable is the responsibility of one of the sensors that has that target in its FOV. Table 4.1 shows the mapping between each part of the hierarchical model and DCOP.

4.3.3 Combination of Hierarchical Architecture and Non-Binary DCOP Modelling

This section details how the proposed non-binary modelling technique can be applied to the various regions at the various levels of the hierarchical architecture depicted in Figure

4.1.

As mentioned in Section 4.3.1, the problem solving in this method is a top down approach. First, the problem in the top-most layer is modelled and solved; lower level layers employ the solution found at the upper layers. At the upper layer, only sensors whose fields of view include more than one lower-level region, participate in the solution. It is important to note that, at this layer, only constraints between sensors of different regions are considered; the internal constraints in each region are relaxed.

Modelling each region’s problem in the lower level is rather straightforward and is done by mapping the sensors to variables and mapping the targets in the field of view of each sensor to domain values of that sensor (c.f. 4.1). Then, the constraint that a minimum of k communicating sensors are required to track each target is imposed.

Some sensors may belong to both upper and lower level regions. Targets assigned to sensors at the upper layer should be kept unchanged when their regions are solved at the lower level. For example, in Figure 4.1 some of the variables of the DCOP of Level 2 may be common to the DCOP’s of Region 0, Region 1, . . . , or Region 9. Of course, these common variables share different constraints in different levels of the hierarchy. They are sharing inter region constraints in Level 2 and intra region constraints in Level 1 of Figure 4.1. Although hierarchical problem solving decreases computation and communication costs efficiently, as described in Eq. 4.2, addressing the problem hierarchically may lead to finding near to optimal solutions because some values are selected for the common variables in the upper level and these selections may impose extra constraints on these common variables in the lower level DCOPs. Lower level DCOP algorithms have to keep previously selected values unchanged. As the number of common variables in each region of Level 1 is much less than the number of non-common variables, this drawback of the proposed method does not change the optimality of the solution.

The experimental results in Section 4.5 confirm this fact. On the other hand, considering the communication and computation complexity reduction of the hierarchical method, in dynamic situations when DCOP algorithms do not have enough time to reach the optimal solution, the benefit of using a hierarchical approach strongly outweighs its costs drawback.

This hierarchical modelling can be simply applied to the target-centered approach as well. It is important to note that when using a hierarchical approach, this method does not risk a single point of failure because in all the levels of the hierarchy the problem solving is a completely distributed process. This approach only partitions the problem and tackles parts of the problem in various levels of the hierarchy. Thus, a part of the problem is tackled at the upper level (the inter region constraints) of the hierarchy and then the other parts of the problem are addressed in the lower level (intra region constraints). Obviously none of the levels of the hierarchy perform a centralized search.

Once the problem is modelled as a non-binary hierarchical DCOP, the performance of two solution methods, ADOPT and DBA, is investigated to determine which constitutes the best candidate for solving the proposed non-binary hierarchical DCOP.

4.3.4 Formal Representation of Hierarchical Non-Binary DCOP Modelling

The hierarchical non-binary DCOP modelling approach is described in Sections 4.3.1 to 4.3.3. To have a more formal view of the presented approach, this section first introduces some notations and then presents a pseudo code which shows how the original problem is transformed into the hierarchical problem. These notations are as follows:

- L : a variable that shows the level of the hierarchy.
- X : the size of the largest manageable problem.
- P_L : the problem in Level L .
- $P_{L,i}$: the sub-problem i in Level L .
- $Target_{L,i}$: targets in sub-problem i in Level L .
- $Sensor_{L,i}$: sensors in sub-problem i in Level L .
- constraint (P_L): total constraints in level L .

- inter-constraint $(P_{L,i}, P_{L,j})$: inter region constraints between sub-problems $P_{L,i}$ and $P_{L,j}$ in level L .
- intra-constraint $(P_{L,i})$: intra region constraints of $P_{L,i}$ in level L .

Algorithm 1 illustrates how a target-to-sensor allocation problem in sensor networks is modelled as a DCOP using the presented hierarchical DCOP modelling approach. This algorithm starts construction of the hierarchy from Level 1. It continues partitioning the main problem into smaller problems until it reaches a problem of manageable size at the topmost level of the hierarchy (Line 2 of Algorithm 1). The partitioning at each level should create manageable sub-problems, minimal inter region constraints and maximal intra region constraints. Also sub-problems should not have any subscription. They should cover all the sensors and targets that are participating at their level of the hierarchy. In addition, the union of inter and intra region constraints should create the total constraints of the problem (Line 3 to 10 of Algorithm 1).

After partitioning, each sub-problem is modelled as a separate DCOP as described in Section 4.3.2 (Line 12 of Algorithm 1). Then, to construct the next level of the hierarchy (Line 18 of Algorithm 1), the algorithm should define the targets, sensors and constraints participating in the next layer. The next layer contains all the sensors and targets that are placed near to the partitioning borders of the current layer. More accurately, the targets that can be sensed by the sensors of more than one region are included in the set of the next layer's targets (Line 21 to 24 of Algorithm 1). Also, all the sensors whose FOV covers more than one region are included in the set of the next layer of sensors (Line 26 to 30 of Algorithm 1). Finally, the inter region constraints of each layer construct all constraints of its next layer (Line 33 of Algorithm 1). In the topmost level of this hierarchy, there is just one manageable sized DCOP, which is modelled using the technique described in Section 5.2 (Line 35 to 38 of Algorithm 1).

Algorithm 1 : Pseudo code for Hierarchical non-binary DCOP modelling

```

1:  $L \leftarrow 1$ 
2: while  $|P_L| > X$  do
3: partition  $P_L$  into a set of sub-problems:  $\{P_{L,1}, P_{L,2}, \dots, P_{L,n}\}$  such that:
4:   a.  $\forall i \quad i = 1 \rightarrow n: \quad |P_{L,i}| \leq X$ 
5:   b.  $\forall i, j \quad i, j = 1 \rightarrow n: \quad \min |inter - constraint(P_{L,i}, P_{L,j})|$ 
6:   c.  $\forall i \quad i = 1 \rightarrow n: \quad \max |intra - constraint(P_{L,i})|$ 
7:   d.  $\forall i, j \quad i, j = 1 \rightarrow n: \quad P_{L,i} \cap P_{L,j} = \emptyset$ 
8:   e.  $Sensor_L = \bigcup_{\forall i} Sensor_{L,i}$ 
9:   f.  $Target_L = \bigcup_{\forall i} Target_{L,i}$ 
10:  g.  $constraint(P_L) = (\bigcup_{\forall i} intra - constraint P_{L,i}) \cup (\bigcup_{\forall i, j} inter - constraint(P_{L,i}, P_{L,j}))$ 
11:
12:  for all  $P_{L,i}, i = 1 \rightarrow n$  do
13:    model  $P_{L,i}$  as a DCOP as described in Section 4.3.2:
14:    a.  $Sensor_{L,i}(/targets_{L,i}) \rightarrow$  variables
15:    b.  $Target_{L,i}(/sensors_{L,i})$  in the FOV of each sensor(/target)  $\rightarrow$  Domain
16:    c.  $intra - constraint(P_{L,i})$  (Equation 4.1)  $\rightarrow$  constraints
17:  end for
18:  $L \leftarrow L + 1$ 
19:
20:    $\triangleright$  all targets that are within FOV of sensors of more than one region participate in the set of targets of next layer
21:  for all  $t_i \in Target_{L-1}$  do
22:    if  $\exists P_{L-1,i}, P_{L-1,j} \mid (FOV(t_i) \cap P_{L-1,i} \neq \emptyset) \wedge (FOV(t_i) \cap P_{L-1,j} \neq \emptyset)$  then add  $t_i$  to  $Target_L$ 
23:    end if
24:  end for
25:
26:    $\triangleright$  all the sensors that their FOV covers more than one region participate in the set of sensors of next layer
27:  for all  $s_i \in Sensor_{L-1}$  do
28:    if  $\exists P_{L-1,i}, P_{L-1,j} \mid (FOV(s_i) \cap P_{L-1,i} \neq \emptyset) \wedge (FOV(s_i) \cap P_{L-1,j} \neq \emptyset)$  then add  $s_i$  to  $Sensor_L$ 
29:    end if
30:  end for
31:    $\triangleright$  inter region constraint of previous level constructs constraints of this level
32:   $constraint(P_L) = inter - constraint(P_{L-1})$ 
33: end while
34:    $\triangleright$  model the last level of hierarchy
35: model  $P_L$  as a DCOP as described in Section 4.3.2:
36: a.  $Sensor_L(/targets_L) \rightarrow$  variables
37: b.  $Target_L(/sensors_L)$  in the FOV of each sensor(/target)  $\rightarrow$  Domain
38: c.  $constraint(P_L)$  (Equation 4.1)  $\rightarrow$  constraints

```

4.4 Complete versus incomplete DCOP algorithms

Figure 4.4 shows a taxonomy of popular DCOP algorithms [6]. This taxonomy describes the DCOP algorithms based on two important characteristics: completeness (guaranteed to find optimal solutions) and centralization (which has a great impact on the distribution of computational costs over a system’s distributed nodes).

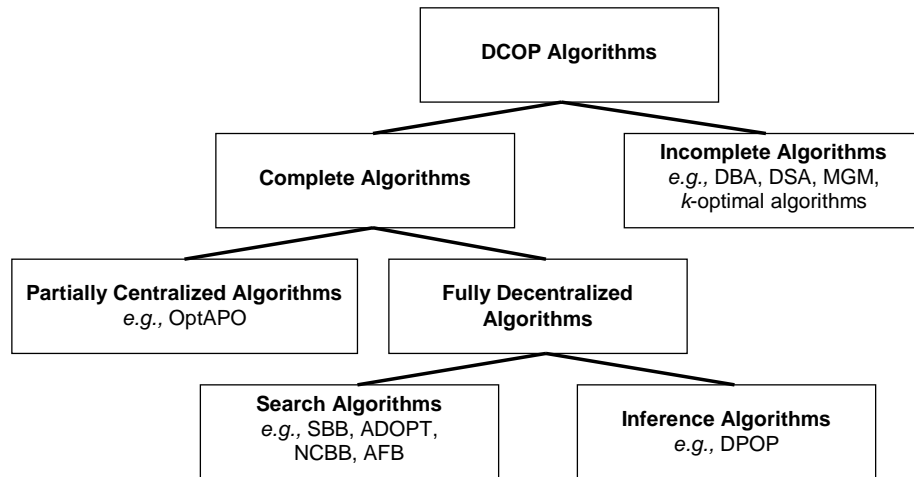


Figure 4.4: Taxonomy of DCSP Algorithms [6]

Incomplete algorithms usually use local search methods to find local optimal solutions, and as a result, can potentially get trapped in local minima. In contrast, complete algorithms typically do an exhaustive search over the problem space; therefore, they can guarantee finding the optimal solution. As DCOP is an NP-hard problem, incomplete algorithms are more practical since finding minimum cost solutions can be intractable [6].

Although completeness is a basic characteristic of many DCOP algorithms, its practical impact as a prerequisite has not been studied in the context of sensor networks. Thus, this research presents a comparative performance study of a complete DCOP algorithm (ADOPT) and a non-complete one (DBA) in the context of the target to sensor allocation in sensor networks.

4.5 Java Implementation and Results

4.5.1 Test Setup

To test the proposed hierarchical approach using ADOPT and DBA, I defined a grid-based sensor network with 224 sensors under various target considerations, ranging from 22 targets to 59 targets. Therefore, the sensor-to-target ratio varied from 10:1 to 3.8:1. This range creates a spectrum of easy to hard problems [140]. Sensors and targets were spread throughout a simplified version of the Waterloo Regional Airport, shown in Figure 4.5.

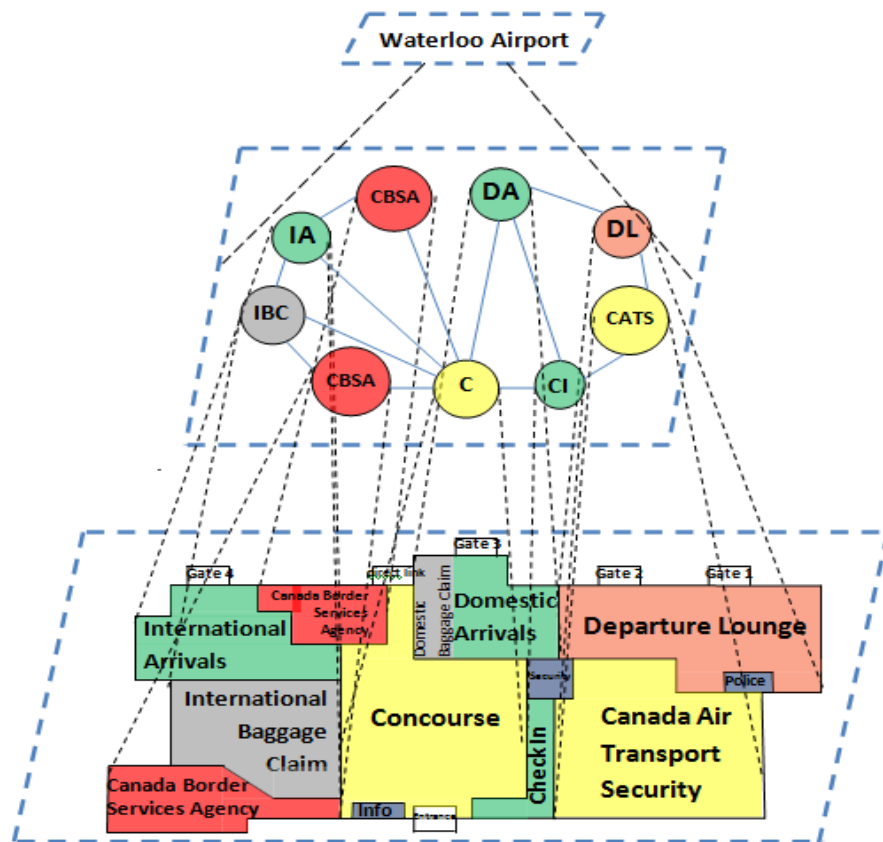


Figure 4.5: Hierarchical target allocation model for sensor networks in a surveillance system (Waterloo regional airport)

This area forms the test case in this research. The lowest level of this figure consist of different sections in the airport, such as, the International Arrivals, Departure Lounge and Baggage Claim section. These sections were separated from one another and hence are modelled as different DCOPs. They were selected because they provide mostly independent geographical regions. The sections or regions share a few constraints with their neighbouring sections and these constraints occur at the pathways between the sections.

As targets are considered mobile in this step of the research, the set of targets that are within the FOV of each sensor changes over time. Such dynamicity of the environment is manifested as changes in the set of constraints. However, I maintained the same number of constraints by ensuring that the number of new constraints is offset by the same number of expunged constraints. To study the system under various dynamicity conditions, the number of changes in each time interval is varied at a rate of 2, 4 and 6 changes. Furthermore, the system performance is studied under 2 changes every 4 seconds (1 rate of change) and 2 every 20 seconds (0.2 rate of change). Constraints were randomly introduced and expunged, and 540 random problems were generated at different sensor-to-target ratios (10:1 to 3.8:1). For each ratio, 20 different random sensor target setups are considered. Taking dynamicity into consideration, 40500 DCOP instances were generated. The experimental setup is summarized in Table 4.2 and Table 4.3.

Table 4.2: Experimental setup-1

number of sensors	number of targets	dynamicity levels	number of time intervals
224	22 to 59	0.2, 1, 2, 4, 6	15

Table 4.3: Experimental setup-2

time intervals	number of problems generated in each case	number of original problems	total number of problems
2 s	20	540	40500

4.5.1.1 ADOPT and DBA algorithms

Two algorithms are considered to solve the proposed hierarchical DCOP, namely, ADOPT [4] and DBA [141, 142]. ADOPT is a complete algorithm; DBA is faster than ADOPT but it is an incomplete one. This section provides a brief overview of these algorithms but readers are directed to [4] and [141] for a more comprehensive description.

ADOPT is a state-of-the-art complete algorithm for DCOP. It is an asynchronous algorithm that guarantees to return an optimal solution while agents execute concurrently. This algorithm lets distributed agents select their variable values in parallel. This algorithm performs a completely distributed search using the communication of costs to guide agents toward globally optimal values to be selected for their variables. Agents first are prioritized into a Depth-First Search (DFS) tree structure in which each agent has multiple children and a parent. Agents communicate their current variable value assignments (VALUE messages) down the DFS tree. Another type of message, a THRESHOLD message, is sent only from parent to child. This message contains a single number representing a backtrack threshold, initially zero. On receiving a message, the receiver calculates its new cost and, if it can, changes its variable value and backtrack threshold. Then it sends VALUE messages to its lower priority neighbours and THRESHOLD messages to its children and also sends cost feedback (COST messages) to its parents. A COST message is sent only from child to parent. COST messages contain lower bounds on objective function F computed based on the values currently selected for higher priority agents. Higher priority agents respond by exploring new values. When x_i receives a COST message, it adds $lb(d, x_i)$ to its local cost $\delta(d)$ to calculate a lower bound for value d , denoted $LB(d)$. More specifically LB is a lower bound for the sub-tree rooted at x_i , when x_i chooses d . Similarly, x_i adds $ub(d, x_i)$ to its local cost $\delta(d)$ to calculate an upper bound for value d , denoted $UB(d)$. As the algorithm continues the lower bound increase gradually, until ultimately, the lower bound of the optimal solution become equal to its upper bound. This condition in which the two bounds are equal to one another and, as a result, equal to the cost of the optimal solution indicates that the optimal solution is found and leads to termination of the algorithm.

DBA is the distributed version of the centralized breakout algorithm. In this algorithm, each agent communicates with its neighbours using two "ok?" and "improve" messages.

The search starts by each agent assigning an initial value to its variable. The algorithm first assigns a weight of one to all constraints. In each step of the algorithm, if none of the agents find a constraint violation, the algorithm terminates. Otherwise, each agent exchanges its current variable value with its neighbours (via an "ok?" message), computes the possible weight reduction if it changes its current value, sends a computed improvement value to its neighbours (via an "improve" message) and decides if it has the largest amount of improvement. If so, it changes the value and applies the improvement. This algorithm, to avoid local minima, increases the weight of constraint violations, which are assigned initially to 1. To avoid simultaneous variable changes among neighbouring agents, only the agent having the maximal weight reduction has the right to alter its current value. If ties occur, the agents break the ties based on their identifiers. In this research, I used the method presented in [85] and [143] to change the DBA to work in dynamic and optimization problems respectively.

It is important to mention that ADOPT is not an iterative improvement algorithm while DBA is listed in this category of DCOP algorithms [142]. Therefore, the any-time results of ADOPT (before ending) can be completely far from its final optimal result. It is while, in DBA, the solution improves iteration by iteration.

4.5.2 Test Results

For evaluation, three main parameters were measured while solving each problem: "the number of messages", "the number of non-concurrent constraint checks (NCCCs)" which shows the computational effort required to solve the problem, and "the cost of the solution".

Each problem is modelled as DCOP(s), once using the proposed non-binary hierarchical method and once using a non-hierarchical method. In the non-hierarchical method, instead of hierarchically portioning the problem into various sub-problems and modelling each problem as a separate DCOP, the whole problem is modelled as one DCOP [87]. The same dynamicity rate is applied in each problem. The results are evaluated with respect to the same parameters (number of messages, number of NCCCs, cost of the solution). Figure 4.6 through Figure 4.16 shows the result of evaluating the performance of ADOPT and DBA algorithms for both the hierarchical method and the non-hierarchical method.

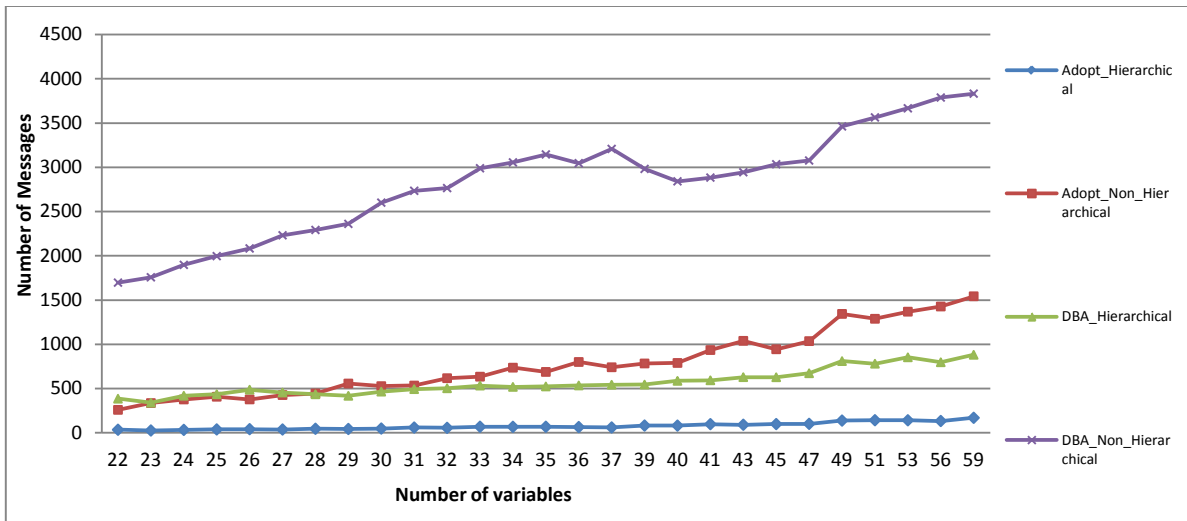


Figure 4.6: Average of the number of messages required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 0.2)

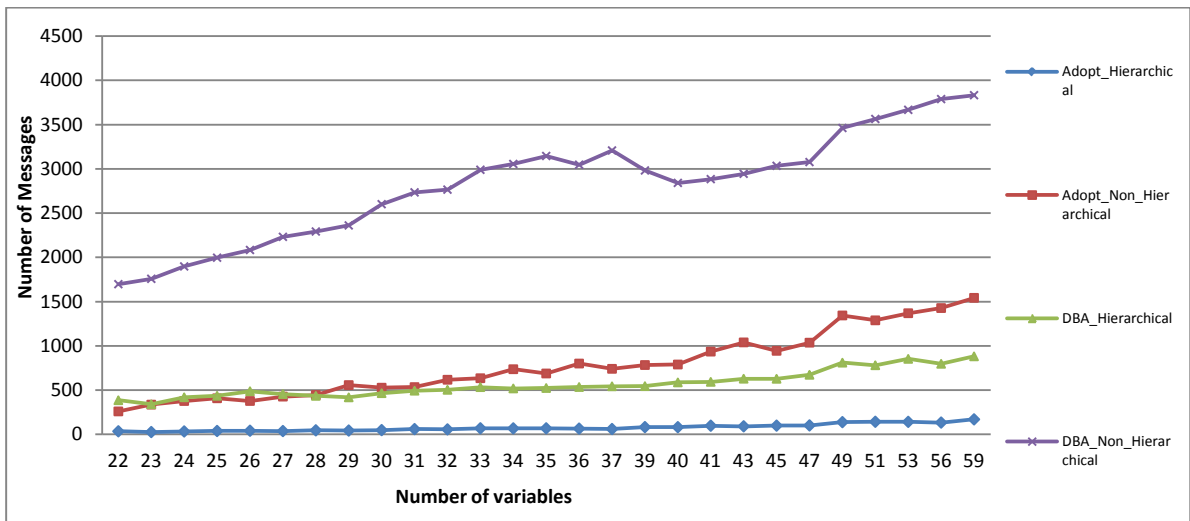


Figure 4.7: Average of the number of messages required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 1)

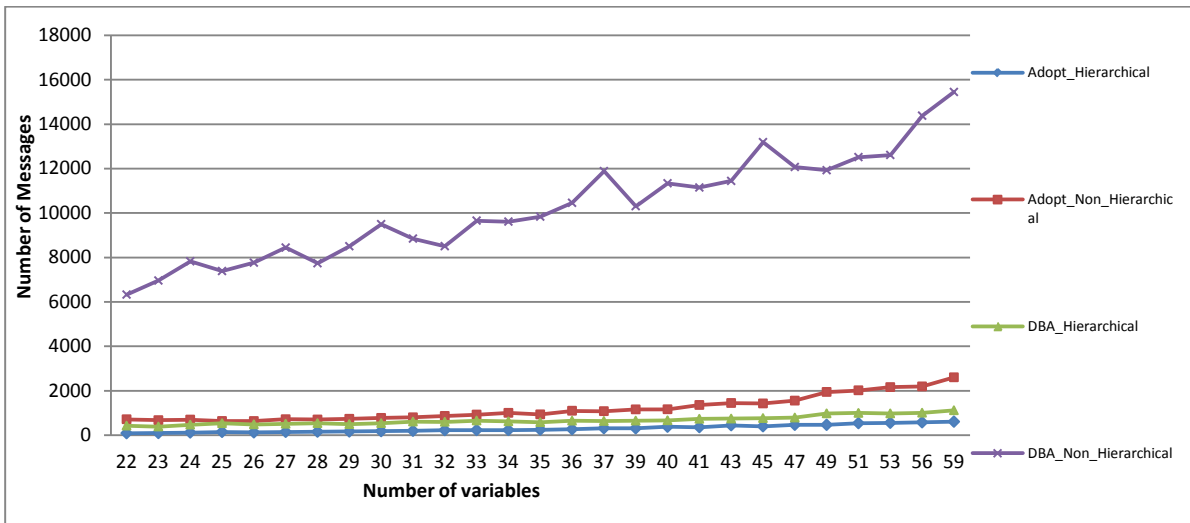


Figure 4.8: Average of the number of messages required by ADOPT and DBA to solve target-to-sensor allocation problem in sensor network (Dynamicity level = 2)

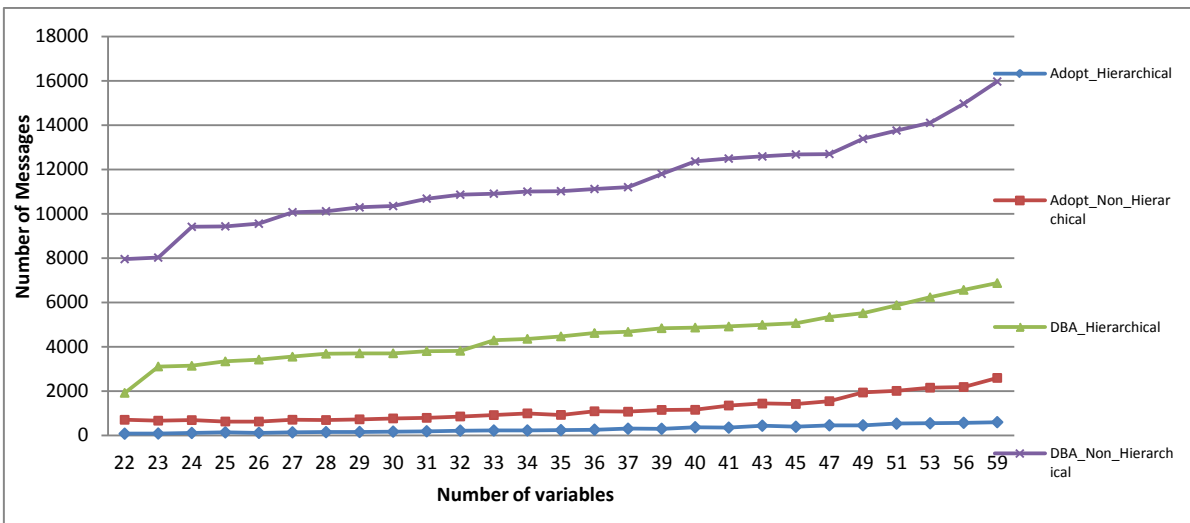


Figure 4.9: Average of the number of messages required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 4)

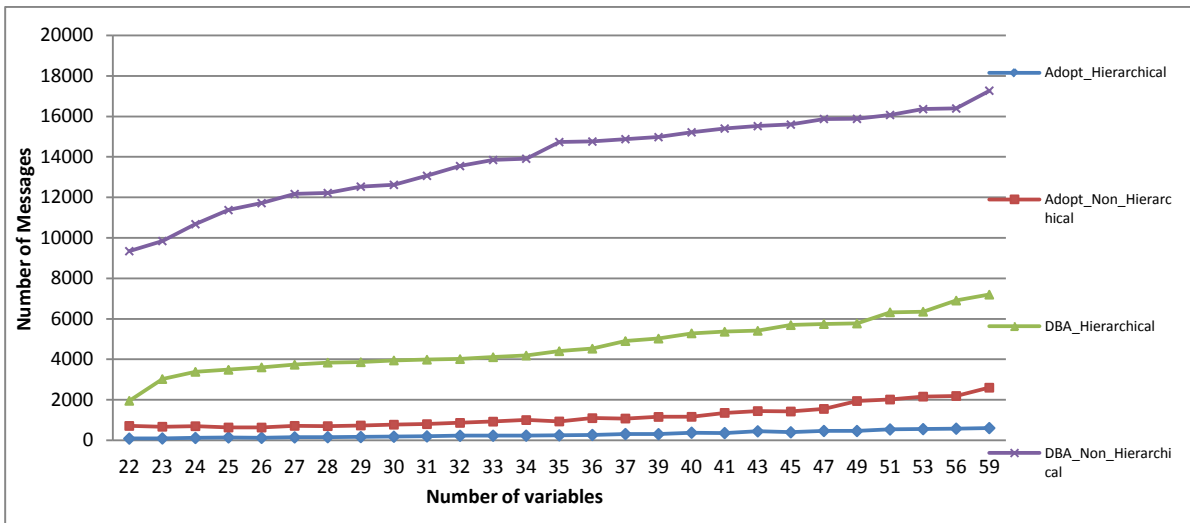


Figure 4.10: Average of the number of messages required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 6)

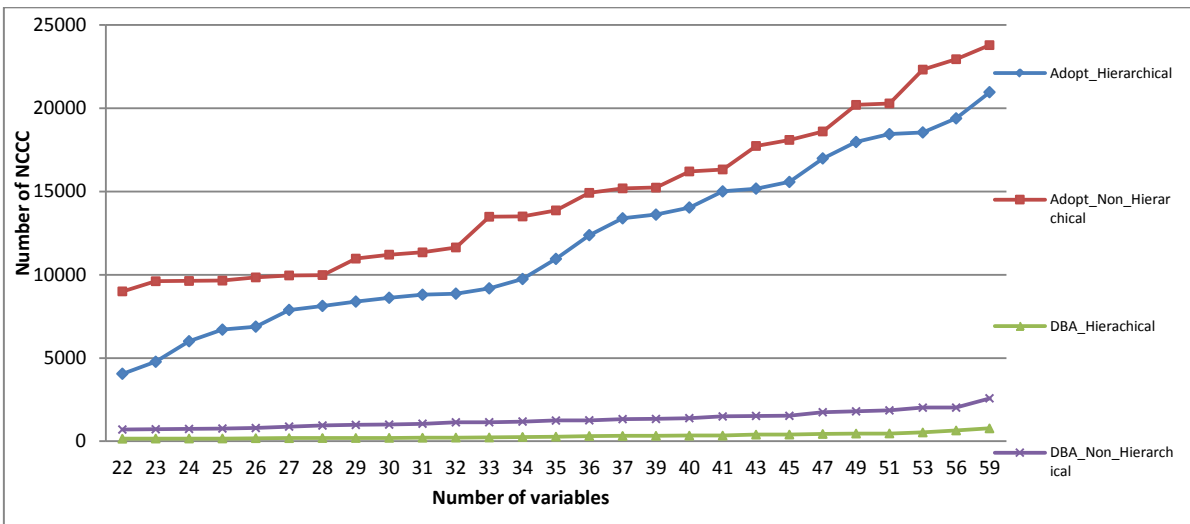


Figure 4.11: Average of the number of NCCCs required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 0.2)

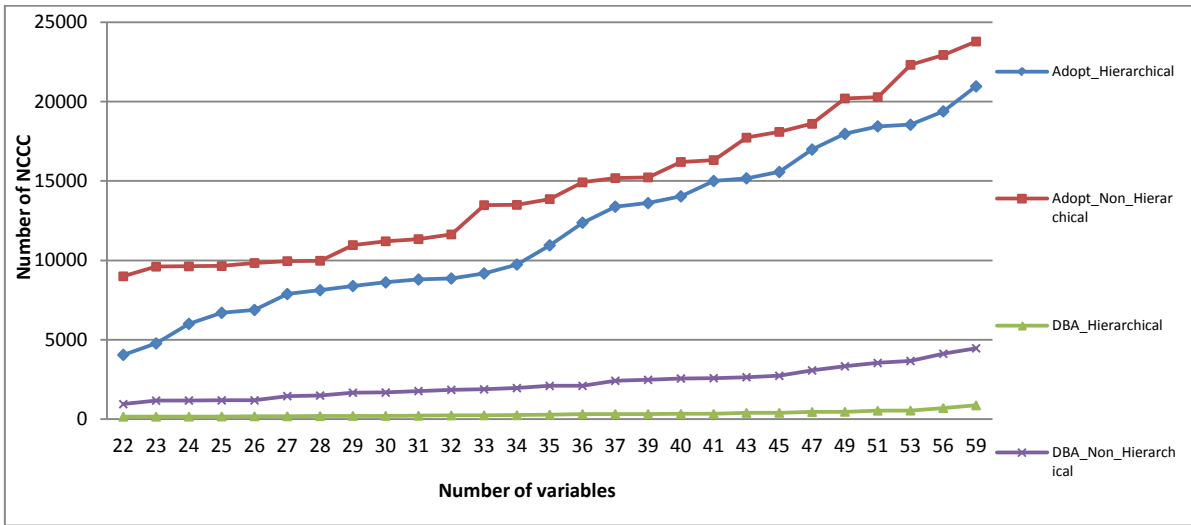


Figure 4.12: Average of the number of NCCCs required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 1)

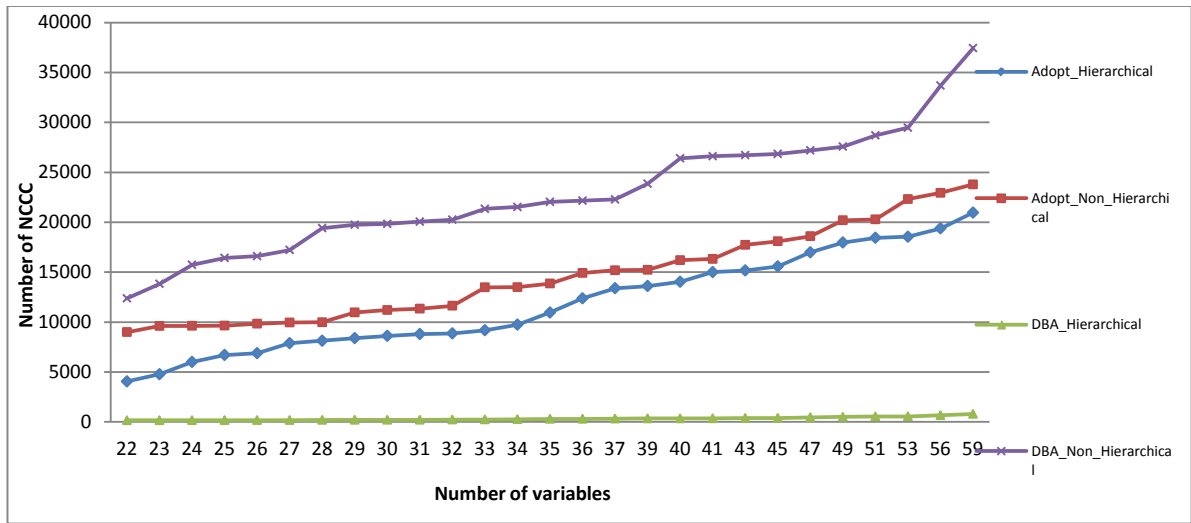


Figure 4.13: Average of the number of NCCCs required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 2)

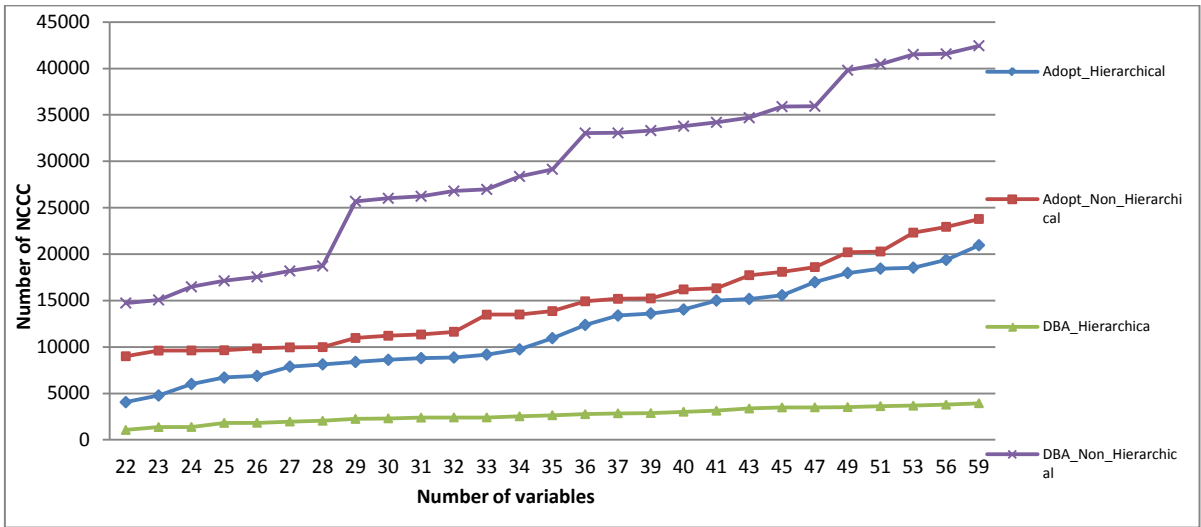


Figure 4.14: Average of the number of NCCCs required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 4)

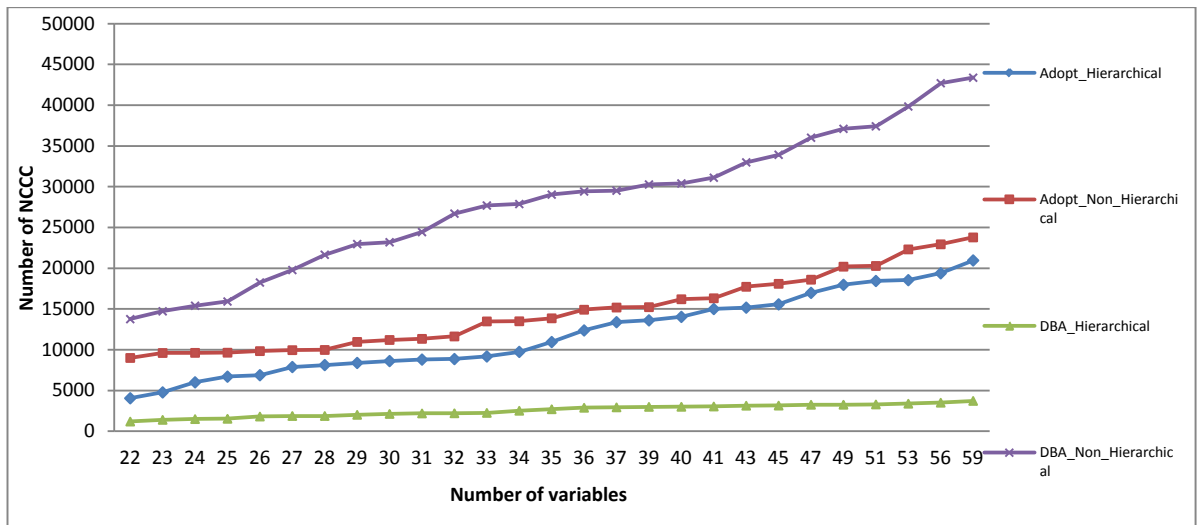


Figure 4.15: Average of the number of NCCCs required by ADOPT and DBA to solve the target-to-sensor allocation problem in sensor network (Dynamicity level = 6)

The first five figures show the evaluation with respect to "the number of messages"

considering various dynamicity levels. The second five figures show the evaluation with respect to "NCCCs", and finally, the third five figures show evaluation with respect to the "total cost of the solution".

These experiments highlight the communicational complexity, computational complexity and the solution quality of each algorithm in converging to an optimal result. It is important to note that "The number of cycles" has been traditionally used in the literature as a metric for evaluation of computational complexity. Nevertheless, and as discussed in various references [144, 145], such a metric is not suitable for evaluating DCSP/DCOP algorithms as it cannot provide insight into the computational efforts needed to solve the problem. NCCC is a good alternative to this metric for evaluating the computational costs of DCSP/DCOP algorithms. To use NCCC as a computational complexity metric, I adopted the method presented by Meisels *et al.* [144].

Figure 4.6 through Figure 4.10 show the number of messages used to solve the problems under various number of variables. The dynamicity level varies from 0.2 to 6 in these figures. The figures illustrate the accumulative number of messages required by the algorithm to solve various instances of the original problem. The average of the results for 20 problems is depicted in the figures. In each case, the results are computed based on 15 consecutive time intervals.

The results of applying ADOPT and DBA in the proposed hierarchical modelling technique as well as in the non-hierarchical method demonstrates that using the proposed hierarchical approach reduces the "number of messages" and the "number of NCCCs" significantly, especially in larger and complex problems. These results were conceivable based on the explanations presented in Section 4.3.1. As Equation 4.2 shows, the cost incurred in solving the problem via partitioning is much less than that of solving the un-partioned large problem. This finding validates the effectiveness of using the hierarchical modelling technique in solving large and complex sensor network problems.

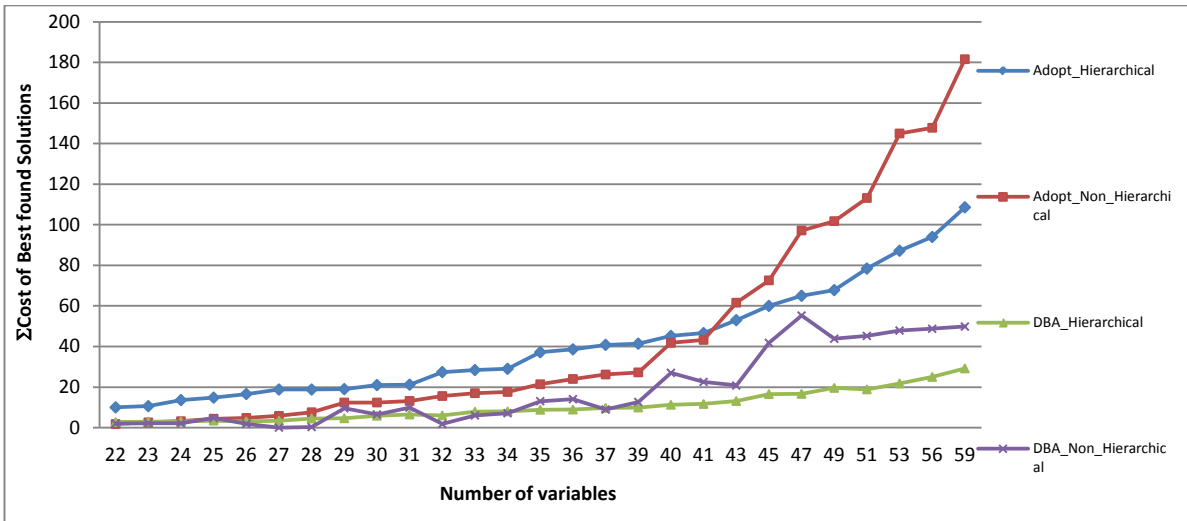


Figure 4.16: The sum of costs of found solutions in solving the target-to-sensor allocation problem in sensor network by ADOPT and DBA (Dynamicity level = 0.2)

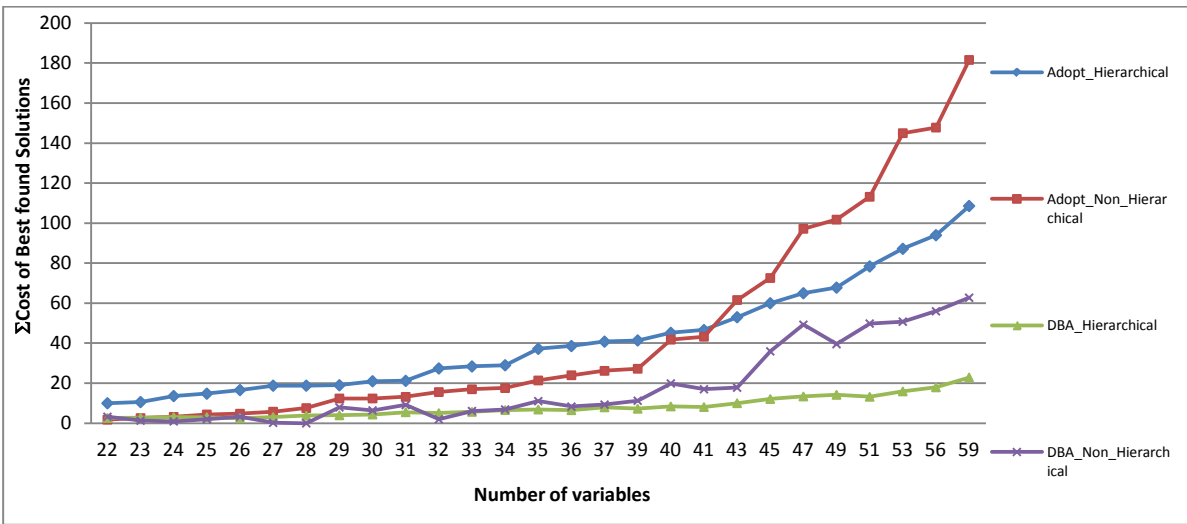


Figure 4.17: The sum of costs of found solutions in solving the target-to-sensor allocation problem in sensor network by ADOPT and DBA (Dynamicity level = 1)

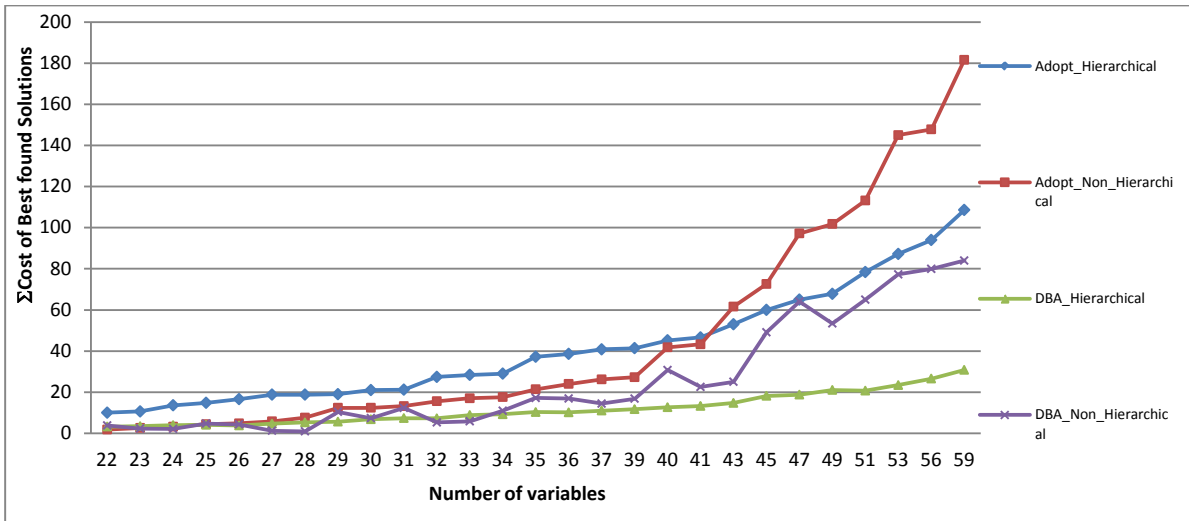


Figure 4.18: The sum of costs of found solutions in solving the target-to-sensor allocation problem in sensor network by ADOPT and DBA (Dynamicity level = 2)

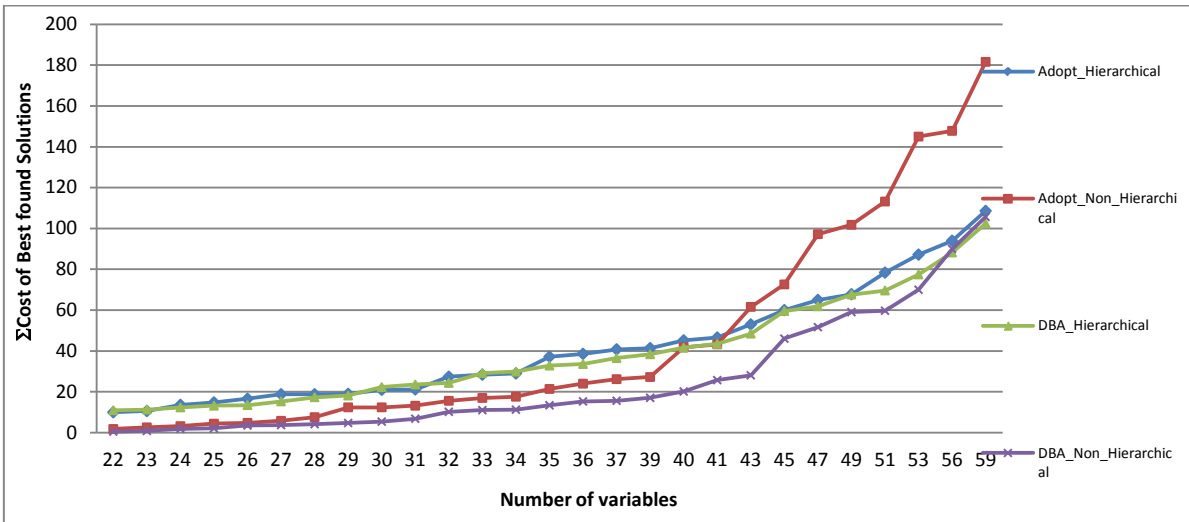


Figure 4.19: The sum of costs of found solutions in solving the target-to-sensor allocation problem in sensor network by ADOPT and DBA (Dynamicity level = 4)

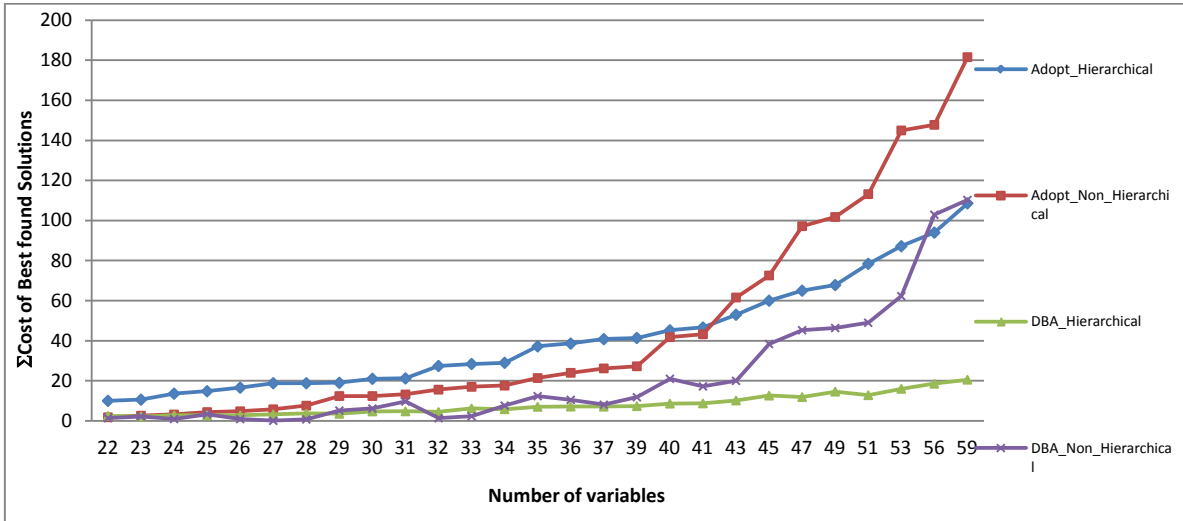


Figure 4.20: The sum of costs of found solutions in solving the target-to-sensor allocation problem in sensor network by ADOPT and DBA (Dynamicity level = 6)

The quality of the solutions, as can be seen from Figure 4.16 through Figure 4.20, is signified by the total cost of all solutions in all the intervals. Assigning a sensor to more than one target, or not ensuring that at least three sensors track a target, increases the cost of the solution. Actually the cost of a solution, in which all the sensors are assigned to one target and each target is tracked by at least three sensors, is zero; and any violation in these criteria increases the solution cost.

The results show that the cost of solutions in both the ADOPT and DBA algorithms, for the hierarchical modelling technique, in most of the cases, is less than that for the non-hierarchical technique, especially in larger problems. Of course, in some cases, especially in smaller problems, a non-hierarchical technique incurs lower costs because in the proposed hierarchical technique, there could be variables that are common across multiple layers and may increase the cost of the solution. In other words, partitioning the problem can introduce some extra costs to the technique.

The experiments highlight the fact that the DBA has better performance as an incomplete algorithm than ADOPT as a complete one, in terms of solution quality. As explained before, the DBA is an iterative improvement algorithm, while ADOPT is not. Also, it is

mentioned that the results reported for both algorithms are their final found results before stopping the algorithm. Therefore, if the algorithm stops due to changes in the problem before ending, then DBA will likely provide better results than ADOPT. In other words, before ending the problem solving, in contrast to DBA, ADOPT may provide a solution far from its final optimal results. The experimental results confirm this fact. The results show that while DBA requires more messages and in some cases more NCCCs to solve the problem, in almost all cases it can find more optimal solutions than ADOPT. DBA works better in dynamic environments, such as the sensor networks problem considered in this research. In such a problem, changes are introduced every 2s to induce the dynamicity typically seen in real sensor networks. Therefore, in most cases, ADOPT is unable to solve the problem in such a short period of time. In contrast, DBA is a faster algorithm and is able to find solutions before the problem changes; also it can provide better any-time results because it is an iterative improvement algorithm. However, the most important drawback of DBA is its high communication costs, which I propose to mitigate using the proposed hierarchical modelling method. Therefore, DBA, under hierarchical modelling, constitutes a better choice for tackling large and complex sensor network problems (Figures 4.18 and 4.19).

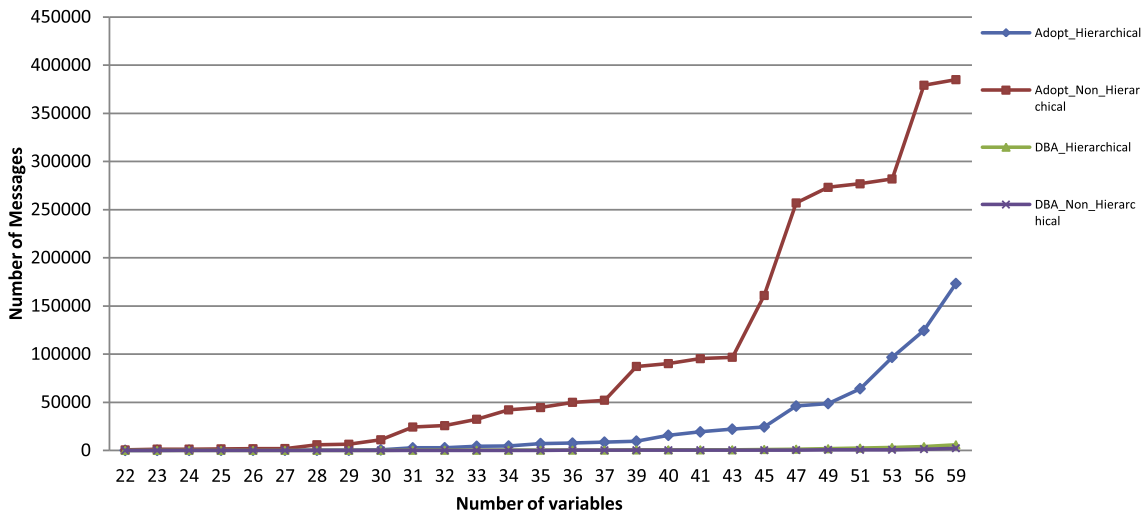


Figure 4.21: The number of messages required to solve the target-to-sensor allocation problem in a static sensor network by ADOPT and DBA.

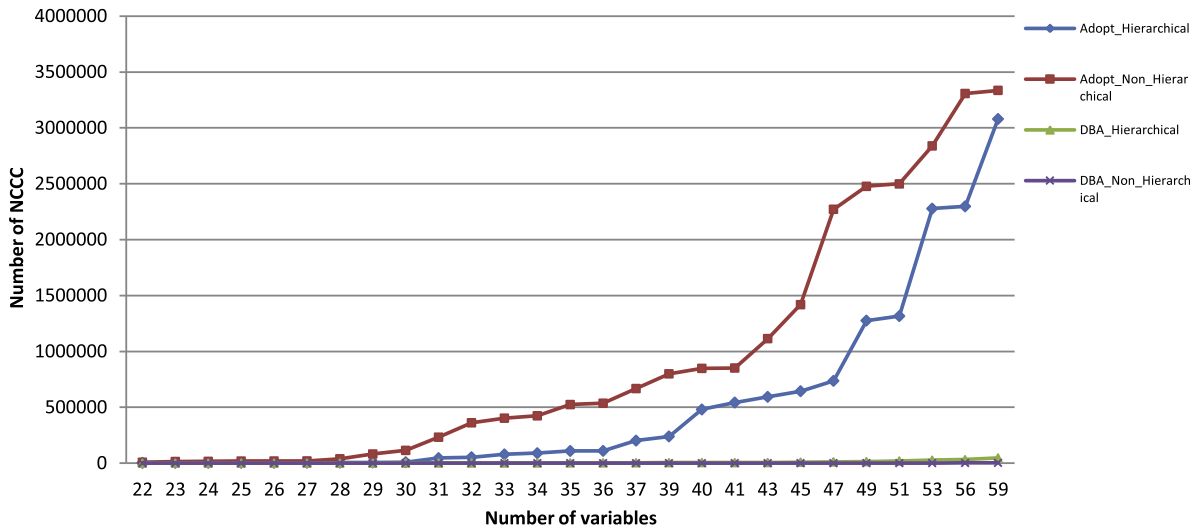


Figure 4.22: The number of NCCCs required to solve the target-to-sensor allocation problem in a static sensor network by ADOPT and DBA.

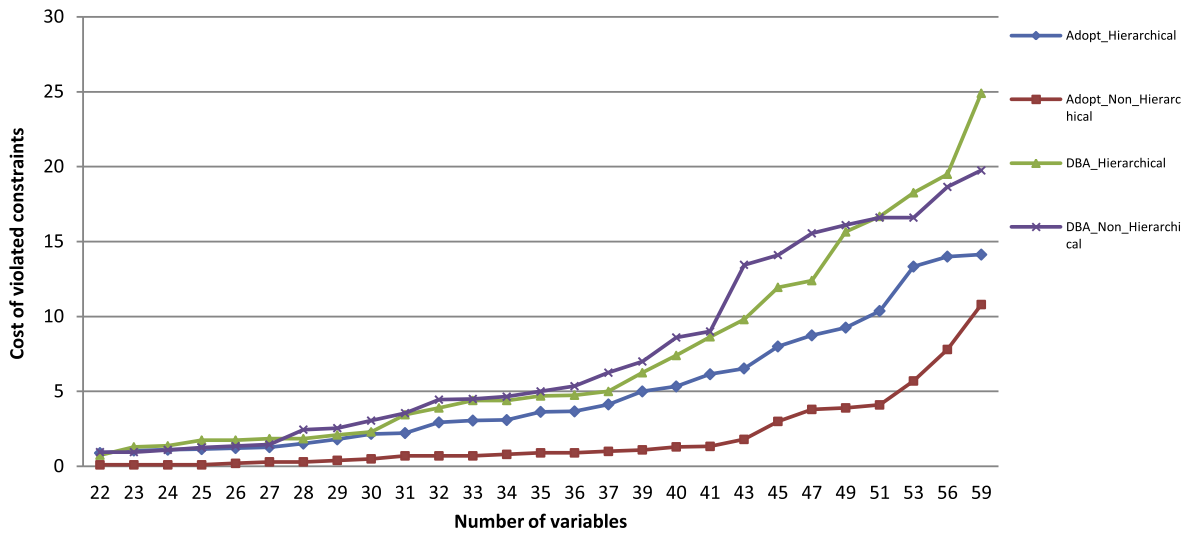


Figure 4.23: The cost of found solution in solving the target-to-sensor allocation problem in a static sensor network by ADOPT and DBA.

To evaluate the proposed method in a static situation, I conducted a set of experiments

in which the problem does not change and the algorithms have enough time to solve the problem. In this experiment, I used the same problems previously used in a dynamic case for starting the dynamic situation. The results are presented in Figures. 4.21 to 4.23. These results confirm the previously found results in the dynamic case. The results show that the communication and computation complexities of the hierarchical approach are much less than those of the non-hierarchical one. As was expected, partitioning reduces the problem solving costs especially in the ADOPT algorithm in which the algorithm's complexity is exponential. On the other hand, the results show that the cost of the best solution found by ADOPT hierarchical algorithm is larger than the best solution cost found in ADOPT's non-hierarchical approach. This is the result of the hierarchical method's inaccuracy, as explained in Section 4.3.3. Although the cost of a hierarchical approach in ADOPT is more than the non-hierarchical one, this cost is still less than the cost found by DBA (both hierarchical and non-hierarchical). Also the solution's cost with DBA non-hierarchical approach in some points is a little bit better than DBA hierarchical approach.

4.6 Summary

A new approach called "hierarchical non-binary variable DCOP modelling" is presented in this chapter to address the limitations of current DCOP/DCSP modelling approaches in solving the target-to-sensor allocation problem. The main characteristics of this method can be summarized as follows:

- Provides a general and formalized method for solving the target-to-sensor allocation problem
- Models the target-to-sensor allocation problem as DCOP instead of DCSP, which is more realistic
- Provides scalability (suitable for large-scale sensor networks)
- Decreases the complexity of modelling using non-binary variables
- Uses a hierarchical structure to address the problem

- Considers the dynamicity issue of the target-to-sensor allocation problem in terms of mobile targets

These characteristics make the proposed approach a suitable method for tackling the target-to-sensor allocation problem in large and complex sensor networks, although several other characteristics related to the dynamicity of sensor networks should be added to the proposed approach in the next chapters.

This chapter also shows how the proposed method can be combined with an appropriate DCOP algorithm and then used to solve a practical target-to-sensor allocation problem in an airport surveillance system as an example of a large-scale, complex and dynamic sensor network. Two DCOP algorithms from two different categories of this field's algorithms were selected for evaluation of the proposed method. They are compared over randomly generated problems which are modelled once using the described hierarchical non-binary scenario and once using the simple non-hierarchical approach.

The results show that, using both complete and incomplete algorithms, the hierarchical non-binary modelling technique provides better results than the simple non-hierarchical approach. This advantage is more pronounced when the size and complexity of the problem is increased, a fact that shows the advantage of using the proposed method in large, complex, and crowded sensor networks such as surveillance applications. Furthermore results show that the incomplete algorithm that uses hierarchical modelling technique is the best choice among tested methods for solving the considered problem.

Chapter 5

Semi-Flocking Algorithm for Motion Control of Mobile Sensors in Large-Scale Surveillance Systems

The ability of sensors to self-organize is an important asset in surveillance sensor networks. Self-organize implies self-control at the sensor level and coordination at the network level. Biologically inspired approaches have recently gained significant attention as a tool to address the issue of sensor control and coordination in sensor networks. These approaches are exemplified by the two well-known algorithms, namely, the Flocking algorithm and the Anti-Flocking algorithm. Generally speaking, although these two biologically inspired algorithms have demonstrated promising performances, they expose deficiencies when it comes to their ability to maintain simultaneous reliable dynamic area coverage and target coverage. These two coverage performance objectives are inherently conflicting.

This chapter presents Semi-Flocking, a biologically inspired algorithm that benefits from key characteristics of both the Flocking and Anti-Flocking algorithms. The Semi-Flocking algorithm approaches the problem by assigning a small flock of sensors to each target, while at the same time leaving some sensors free to explore the environment. This allows the algorithm to strike a balance between the reliable area coverage and target coverage. Such balance is facilitated via flock-sensor coordination. Section [6.1](#) provides an

introduction to the proposed work. Section 5.2 introduces the concept of Semi-Flocking motion modelling. The performance of the proposed Semi-Flocking algorithm is examined and compared with the other two Flocking-based algorithms, once using randomly moving targets and once using a standard walking pedestrian dataset in Section 5.3. The results of both experiments show that the Semi-Flocking algorithm outperforms both the Flocking algorithm and the Anti-Flocking algorithm with respect to the area of coverage and the target coverage objectives. Furthermore, the results show that the proposed algorithm demonstrates shorter target detection time and fewer undetected targets than the other two Flocking-based algorithms. Finally, concluding remarks are given in Section 5.4.

5.1 Introduction

Sensor Networks have demonstrated noticeable success in mobile surveillance applications [14, 15, 16, 17], showing advanced capabilities to self-organize, and to cooperate and coordinate their activities to collect information about targets and events in a given volume of interest (VOI). The information collected by the sensors is often fused to obtain a complete picture of the environment and assess situations of interest. Due to communication and energy restrictions, centralized data fusion algorithms are not efficient and there is a need for distributed algorithms that restrict the communication between neighbours [18]. The ability to self-organize constitutes an indispensable attribute in surveillance applications where target mobility increases surveillance complexity. In this case, sensor mobility comes in handy to enable the network to achieve dynamic area coverage and reliable target detection.

An important challenge in self-organizing surveillance systems is the control and coordination of sensor mobility. This problem concerns the optimal movement of a set of mobile sensors to achieve maximum area and/or target coverage [19], maximum radio coverage between the sensors [20, 21], or improved target coverage over maneuverable targets [22], etc. This research addresses the issue of sensor control and coordination for maximum area and target coverage.

The Flocking Algorithm is one of the approaches recently reported in the literature

that addresses the issue of sensor control and coordination in sensor networks. This algorithm has attracted significant interest in recent years in the field of mobility control [129, 5, 130, 131, 132, 133, 134, 135, 136, 137]. Flocking is a biologically inspired behaviour that embodies a form of cooperative behaviour of a large number of autonomous interacting agents to achieve a coordinated group behaviour. Group movements of birds, fishes, insects and bacteria are examples of the flocking behaviour in nature. To conceive flocking behaviour, each agent follows a set of flocking rules and maintains some sort of communication with its neighbouring agents. Self-organization and local communication requirements of the flocking process provide an inspiring behaviour in the management of sensors in mobile sensor networks.

This chapter introduces the Semi-Flocking algorithm, a modified version of the Flocking algorithm [132]. Two other Flocking-based algorithms are discussed and used as benchmarks to study the performance of the proposed Semi-Flocking algorithm. These two algorithms are Flocking [132] and Anti-Flocking [131]. They are described in detail in Sections 2.4.2.1.1 and 2.4.2.1.2. This chapter demonstrates how the proposed Semi-Flocking algorithm provides a better alternative to conceiving self-organizing capabilities in mobile sensor networks contemplating reliable surveillance performance.

5.2 Semi-Flocking Approach

Section 2.4.2.1 described the main Flocking-based algorithms (Flocking and Anti-Flocking algorithms) as they pertain to sensor management in sensor networks. Although each of these algorithms possess intriguing characteristics, each has indispensable drawbacks. The proposed Semi-Flocking algorithm attempts to address these drawbacks. This section introduces this algorithm and shows how self-control and coordination is facilitated so as to ensure a high level of area coverage and target coverage.

Analysis of the Flocking algorithm and the Anti-Flocking algorithm in the context of surveillance applications reveals that their drawbacks are expected due to their emphasis on one aspect of performance, and not the two. The Flocking algorithm focuses only on reliable target coverage, while the Anti-Flocking algorithm focuses only on dynamic

area coverage. As mentioned in Section 6.1, surveillance, by definition, is to achieve both dynamic area coverage and reliable target detection. Ignoring either one of these two objectives compromises the intended purpose of surveillance. The proposed Semi-Flocking algorithm presented in this chapter attempts to fill this gap. The main idea of the Semi-Flocking algorithm is to create small flocks of sensors around each target while still leaving some sensors free to search the surveillance environment for the detection of new targets. The Semi-Flocking concept is depicted in Figure 5.1.

As illustrated in Figure 5.1, while the Semi-Flocking approach obtains acceptable dynamic area coverage, it still creates small flocks of sensors around the position of each target. Although these flocks are smaller than those in the Flocking algorithm, they still can efficiently avoid missing targets. Furthermore, the Semi-Flocking algorithm is able to cover targets better than the Flocking algorithm (on average). Another interesting feature of the Semi-Flocking approach is its ability to allow sensors to switch between two modes, *i.e.*, the tracking mode and the searching mode. For example if a target leaves the area of interest (AOI), then the members of the flock around it either join the other flocks to increase the strength of target coverage, or join the searching sensors to increase the chance of detecting new targets.

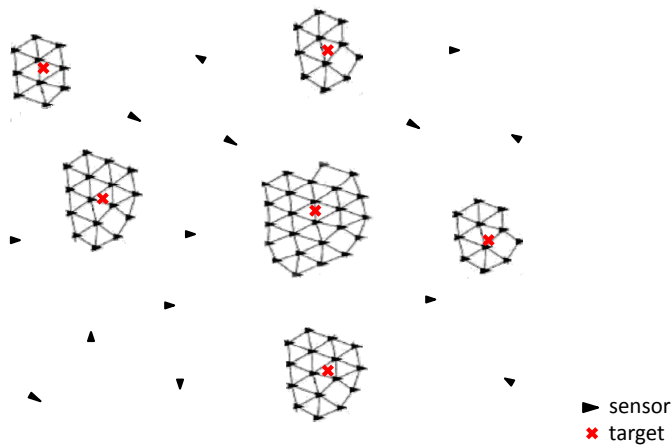


Figure 5.1: Semi-Flocking algorithm for sensor management in surveillance applications

There are many questions to be answered regarding the Semi-Flocking algorithm, in-

cluding: how does each sensor select the best target for tracking? (decide which flock to belong to); How do different flock members avoid collision?; How many sensors track the targets and how many of them search the AOI?; What happens when a new target comes into the AOI?; Which particles should create the flock for a new target?; What happens when a target leaves the AOI?; What is the job of free sensors? Section 5.2.1 introduces the sensor motion control model that captures these questions.

5.2.1 Semi-Flocking Algorithm

Semi-Flocking behaviour is a result of applying simple rules by each sensor. Although these rules are very simple and can be represented as an input vector for each sensor, the result is a complicated group behaviour which I call Semi-Flocking behaviour. This section describes how this input vector is calculated for Semi-Flocking algorithms and how it is different from the Flocking and Anti-Flocking rules. In the Flocking algorithm [5], as was stated in Section 2.4.2.1.1, sensors apply a control input vector *i.e.* $u_i = f_i^g + f_i^d + f_i^\gamma$, in which the first two terms are related to the three Reynold's rules: flock centering, collision avoidance and velocity matching. The third term (f_i^γ) is a navigational feedback that attracts all the sensors toward one target in the Flocking algorithm. In the Semi-Flocking algorithm, each sensor i applies a control input vector $u_i = f_i^g + f_i^d + f_i^\gamma$ similar to the control vector in the Flocking algorithm except that it makes an essential modification in the third term (f_i^γ) to modify the navigational method of the Flocking algorithm. This modification induces the behaviour where sensors either get attracted toward one of the surrounding targets, or alternatively emerge free to search the AOI to look for new targets. The sensors are selected to track a target based on two important factors: 1) distance between the sensor and the target, 2) the number of sensors already tracking the target. Each sensor applies Equation 5.1 to calculate its navigational part.

$$\begin{aligned}
u_i^\gamma &= f_i^\gamma(q_i, p_i, q_{t1}, \dots, q_{tm}) = \sum_{j=1}^m \varphi(q_{tj} - q_i) \frac{c_{1j}(q_{tj} - q_i)}{n_{tj}} + \sum_{j=1}^m \varphi(q_{tj} - q_i) \frac{c_{2j}(p_{tj} - p_i)}{n_{tj}} \\
&= \sum_{j=1}^m \varphi(q_{tj} - q_i) \frac{c_{1j}(q_{tj} - q_i) + c_{2j}(p_{tj} - p_i)}{n_{tj}} \quad (5.1)
\end{aligned}$$

Where m represents the number of targets, c_{1j}, c_{2j} are positive constant values, $q_{tj} - q_i$ is a vector along a line connecting sensor i to target tj ; n_{tj} represents the number of sensors currently tracking the target tj ; $p_{tj} - p_i$ represents the difference between the velocity of sensor i and target tj and $\varphi(q_{tj} - q_i)$ is a switching function taking 0 – 1 values defined by

$$\varphi(q_{tj} - q_i) = \begin{cases} 1 & q_{tj} - q_i < \theta_j \\ 0 & \text{otherwise} \end{cases} \quad (5.2)$$

Figure 5.2 illustrates the pseudo code of the navigational control applied by each sensor i in the Semi-Flocking algorithm for computing the size and direction of vector $f_i^\gamma = u_i^\gamma$.

```

(1)  $u_i^\gamma = 0$ ;
(2) for target  $j=0$  to  $m$  do
(3)   if  $\|q_{tj} - q_i\|_\sigma \leq \theta_j$  then
(4)      $u_{i,tj}^\gamma = \frac{c_{1j}(q_{tj}-q_i)+c_{2j}(p_{tj}-p_i)}{n_{tj}}$ 
(5)     //where  $n_{tj}$  is the number of sensors already tracking target  $tj$ 
(6)      $u_i^\gamma = u_i^\gamma + u_{i,tj}^\gamma$ ;
(7)   end if
(8) end for
(9) if  $u_i^\gamma == 0$  then // the sensor is in searching mode
(10)   $q_{w,l}$  = center of adjacent areas that has least visited times
(11)   $u_i^\gamma = c \times (q_{w,l} - q_i)$ 
(12)  //toward the area that has longest time not being visited
(13) end if

```

Figure 5.2: Semi-Flocking navigational-control pseudo code for sensor i

As illustrated in Figure 5.2, all the targets whose Euclidean distance from the position of the sensor i is less than a threshold value θ_j (line 3 of Figure 5.2) participate in calculation of u_i^γ . The threshold value for each target (θ_j) depends on the number of sensors around that. At the beginning of the algorithm θ_j is set to a default value which depends on the total number of participating sensors and target. This value is same for all the targets. Later the default value will be corrected for each target based on the number of sensors that are supporting that target. The higher the number of supporters the lower the value of θ_j and vice versa. The contribution of each target t_j is calculated by Equation 5.3:

$$u_{i,t_j}^\gamma = \frac{c_{1j}(q_{t_j} - q_i) + c_{2j}(p_{t_j} - p_i)}{n_{t_j}} \quad (5.3)$$

The size of the vector u_{i,t_j}^γ is inversely proportional to the size of the flock around each target. The larger the size of the flock, the smaller the navigational vector. The multi-target navigational feedback concept is represented in Figure 5.3 In this figure f_i represents the summation of three vectors: $f_{i1}^\gamma(u_{i,t1}^\gamma)$, $f_{i2}^\gamma(u_{i,t2}^\gamma)$, $f_{i3}^\gamma(u_{i,t3}^\gamma)$ that are the control functions applied to a sample sensor.

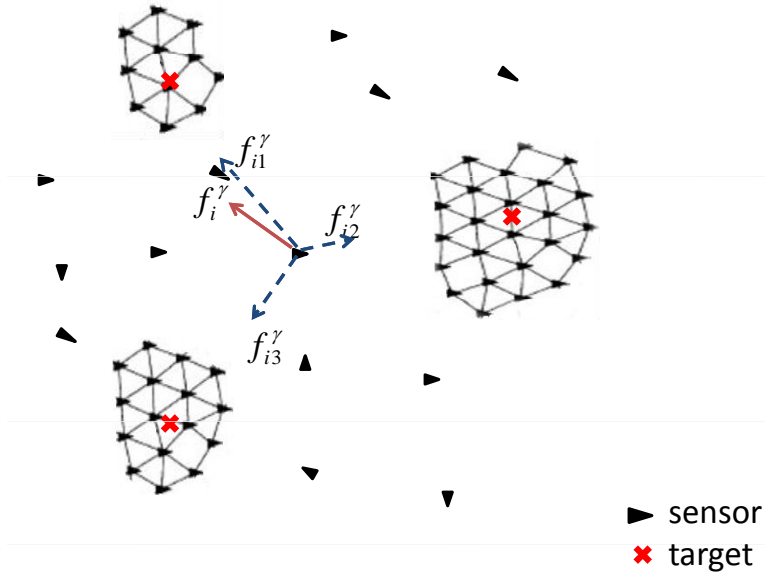


Figure 5.3: Multi-target navigational-control in Semi-Flocking algorithm

In the navigational part of the Semi-Flocking algorithm as represented in Figure 5.4, if no target exists close to sensor i (within distance θ), the sensor searches the AOI to detect incoming targets. In this part, each sensor, instead of doing a random search, moves toward the surrounding area that has the longest time not being visited. Assuming the targets have a uniform distribution, it is more likely to detect a target in such an area.

To find the least visited adjacent area for each sensor, a counter counts the number of times each area is covered by sensors. Suppose that $q_{w,l}$ represents the center of the adjacent area that has the least visited times, then $q_{w,l} - q_i$ represents a vector along the line connecting the current position of the sensor to the center of the least visited area. In the Semi-Flocking algorithm, if a sensor is not selected to track a target, then it will be attracted to the least visited area by calculating the u_i^γ vector using Equation 5.4. In this equation c is a positive constant value that adjusts the size of the vector.

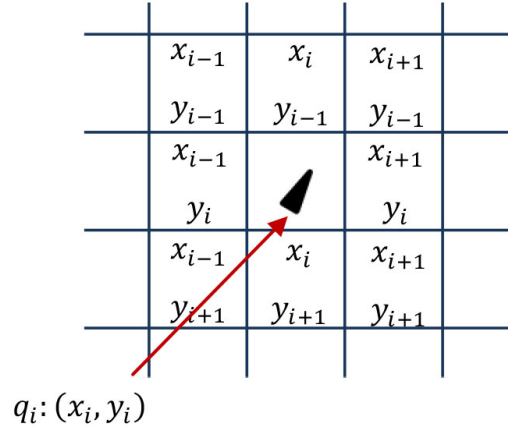


Figure 5.4: Moving toward least visited area in navigational-control of Semi-Flocking algorithm

$$u_i^\gamma = c \times (q_{w,l} - q_i) \quad (5.4)$$

5.3 Experiments and Discussion

5.3.1 Evaluation Parameters

Sensor management in the context of a surveillance application has a twofold objective. First, it should demonstrate reliable target coverage, and second, it should be able to obtain acceptable dynamic area coverage. Based on these requirements, I defined four parameters: Target Coverage (TC) , Target Detection Time (TDT) , Percentage of Non-detected Targets (PNDT) and Cumulative Area Coverage (AC) as the main evaluation parameters. These parameters are very similar to the ones applied for evaluation of the Anti-Flocking algorithm [131] and can be categorized into two groups: the first three parameters evaluate algorithms over target coverage and the last one evaluates their area coverage.

- **Target Coverage:** Target coverage for a target is the percentage of its lifetime in which it is covered by at least k sensors. Assume that k is the minimum number of sensors required for each target's complete coverage. Equation 5.5 represents the formula for computing target coverage of target i .

$$TC_i = \frac{\sum t_i}{Lifetime_i} \times 100 \quad (5.5)$$

Where i represents a target, each t_i is a fraction of time that target i is covered at least by k sensors, and $lifetime_i$ is the time that target i exists in the area of interest.

- **Target Detection Time:** The detection time of a target is defined as the time passing until the target is covered by at least k sensors.
- **Percentage of Non-detected Targets:** This parameter represents the percentage of targets that have never been covered by at least k sensors during their lifetime.
- **Cumulative Area Coverage [146]:** The cumulative area coverage for time interval $[0, t)$ in a surveillance system is the fraction of the AOI that is covered at least once by at least one sensor within time interval $[0, t)$.

5.3.2 Experimental Setup

I implemented a Java version of the Flocking algorithm in the framework presented by Olfati-Saber [5]. Semi-Flocking and Anti-Flocking algorithms are also implemented in the same framework. The following parameters remained fixed through the implementation of all three Flocking-based algorithms: $d = 20$, $r = 1.2d$, $\epsilon = 0.1$ (for σ -norm), $a = b = 5$ for $\varphi(z)$, $h = 0.2$ for the bump function of $\varphi_\alpha(z)$ and the step-size in all simulations is 0.02 seconds.

5.3.2.1 Random Moving Targets

In this experiment the AOI of the surveillance system is a 1250×665 rectangle. There are 150 mobile sensors in the system, and the detection radius of each sensor is 30. The number of critical targets that need to be tracked by the sensors varied from 0 to 9 to show low to high density problems. For each number of targets, 10 random instances were generated. The reported results are the average over these instances. All the targets are mobile and have a constant speed. In all the instances, the initial position, entrance time, and lifetime of each target were selected randomly by a uniform distribution. The initial position and velocity of all sensors are also selected using a uniform random distribution. The same random seeds are used to generate the same instances in the three algorithms: Flocking, Anti-Flocking and Semi-Flocking. For each instance, the monitoring time is continued for 360 sec (6 minutes). After entering, a target moves randomly around until the end of its lifetime. It is assumed that a target is covered if it is in the field of view of at least 3 sensors ($k = 3$). This assumption matches with the requirement of many applications of sensor networks [147, 148].

5.3.2.2 Walking Pedestrian Dataset

In the second experiment a standard dataset is used which is collected from digital video sequences of actual walking pedestrians in a busy scenario from a bird eye view. The scene used for data collection was filmed from the 4th floor of a hotel in Zurich in 2009 [149].

This dataset contains around 18061 position and velocity observations for 420 individuals which are manually annotated. Figure 5.5 represents a sample frame of this dataset.



Figure 5.5: Sample frame from the walking pedestrian dataset

5.3.3 Simulation Results and Analysis

5.3.3.1 Random Moving Targets

Figure 5.6 shows a snapshot of the proximity structure for 150 sensors applying the Semi-Flocking algorithm after a few seconds of starting the algorithm. The sensors are supposed to track 5 mobile targets in this example. As this figure demonstrates a small flock is formed and maintained around each target. At the same time there are some free sensors exploring the AOI. The observations are in close agreement with our expectation of the behaviour of the Semi-Flocking algorithm depicted in Figure 5.1

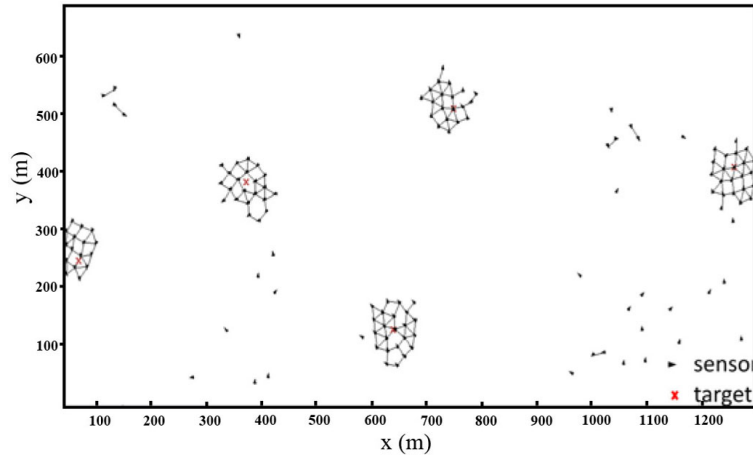


Figure 5.6: Snapshot of executing the Semi-Flocking algorithm by 150 sensors

The result of evaluation of the Semi-Flocking algorithm and its comparison with Flocking and Anti-Flocking algorithms with respect to the parameters introduced in Section 5.3.1 are as follows:

- **a) Target Coverage:** Figure 5.7 shows the average of 10 runs of target coverage for three Flocking-based approaches. Each point in the graph represents the average TC over the number of targets. For example if the number of targets is 2, then the diagram shows the value of $(TC_1 + TC_2)/2$.

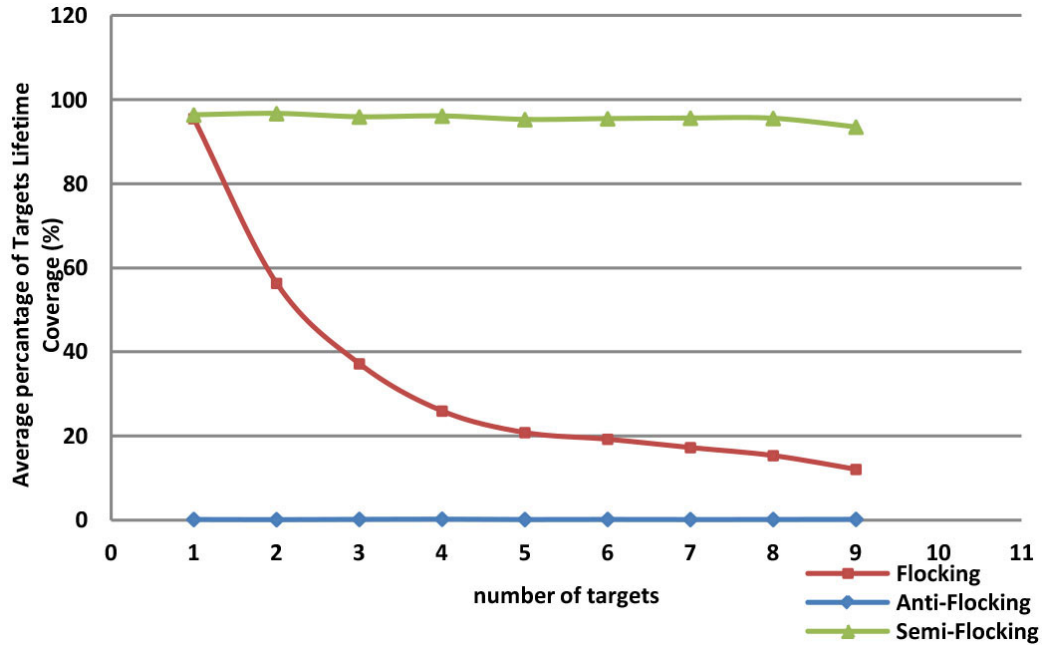


Figure 5.7: Average percentage of Target Coverage (TC) in three Flocking-based algorithms for random moving targets

As Figure 5.7 shows, the Semi-Flocking algorithm demonstrates higher target coverage than the other two algorithms for all numbers of targets, and it is the most interesting aspect of this algorithm. The Semi-Flocking algorithm creates smaller flocks than the Flocking algorithm, but it forms a flock around each target. If the number of targets increase, the size of flock around each target decreases automatically. This reduction slightly increase the chance of missing a target and, as a result, average target coverage decreases by increasing the number of targets.

The Flocking algorithm works well for a small number of targets, but as the number of targets increase the average target coverage drops rapidly. This behaviour is due to the creation of a flock (containing all the sensors) around the first target by the Flocking algorithm. Therefore, its target coverage for one target is perfect. However, it does not create a flock around the other targets (Figure 2.9 represents this problem). Although other targets may be covered, by chance, when they are placed inside the

flock that is tracking the first target, there is no plan in this algorithm for their coverage. The result is a high value for TC_1 and low values for TC_2 to TC_10 . Thus, for the Flocking algorithm, the average value of TC decreases rapidly by increasing the number of targets.

The Anti-Flocking algorithm does not demonstrate acceptable target coverage and this matter is not relevant to the number of targets. This behaviour can be explained by highlighting the fact that this algorithm does not create any flocks around targets. This behaviour decreases the chance of a target being covered by at least k sensors in its lifetime; and this chance does not change by increasing the number of targets.

- **b) Target Detection Time:** Figure 5.8 shows the average of 10 runs of target detection time for three Flocking-based approaches. Each point in the diagram represents the average of TDT over the number of targets. As Figure 5.8 illustrates, Semi-Flocking shows suitable results. The Flocking algorithm works well for the small number of targets but its results drop rapidly by increasing the number of targets. The Anti-Flocking algorithm fails to demonstrate acceptable average TDT for all the cases. The results are a reflection of the Flocking behaviour of these algorithms. The Flocking algorithm has the best TDT for the first target, but since it has no plan for detecting other targets, the TDT value for others is high. Thus by increasing the number of targets, the average of TDT increases rapidly. The Semi-Flocking algorithm demonstrates the best results. The results are excellent when there are only a few targets in the surveillance area. However, as the number of targets increase, the number of free sensors that search the AOI for the detection of new targets decreases slightly. As a result, the TDT for the next targets increases and, consequently, the average TDT increases a little. The Anti-Flocking algorithm has a high and almost constant TDT for all numbers of targets. Because this algorithm does not create a flock around each target and it takes a long time to k sensors be placed around each target simultaneously and eventually be able to detect the target.

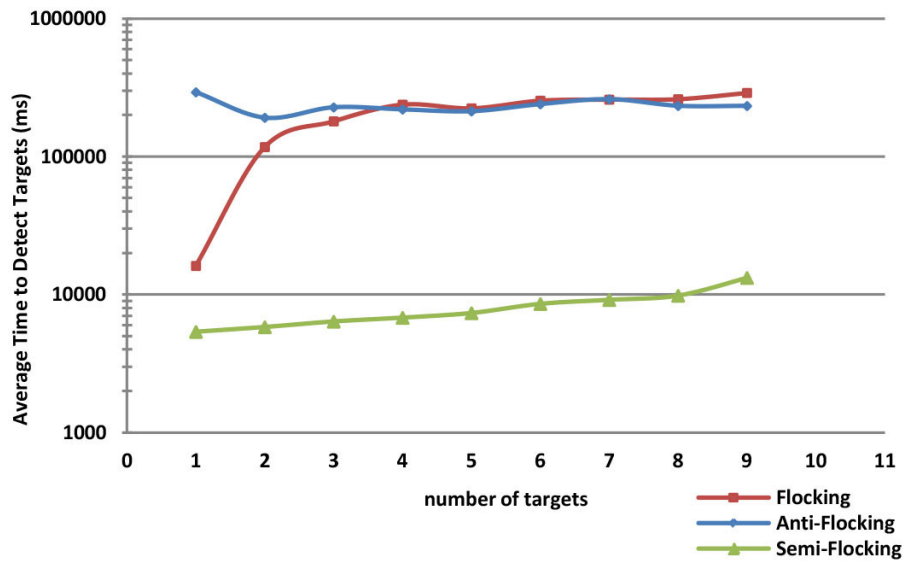


Figure 5.8: Average Target Detection Time (TDT) in three Flocking-based algorithms for random moving targets

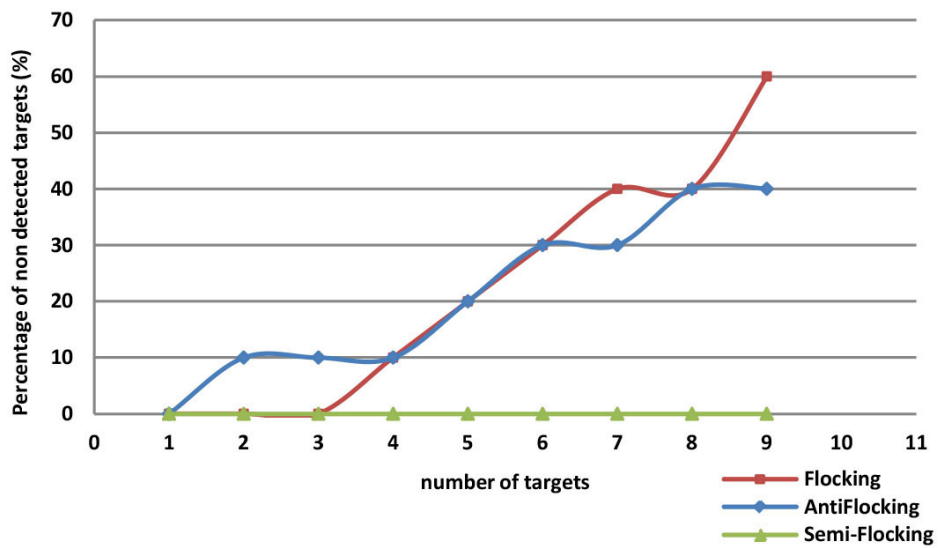


Figure 5.9: Percentage of Non Detected Targets (PNDT) in three Flocking-based algorithms for random moving targets

- **c) Number of Non-detected Targets:** Figure 5.9 shows the average percentage of non-detected targets in three Flocking-based approaches. Each point in the diagram represents the average of PNDT over the number of targets. As illustrated in Figure 5.9, the Semi-Flocking algorithm miss less than 5 percent of targets on average for all numbers of targets. The Flocking and Anti-Flocking algorithms miss more targets. The same reasons presented to explain TDT results (Figure 5.8) can also explain PNDT results, since both are related to the target detection parameter.
- **d) Cumulative Area Coverage:** Figure 5.10 shows the average of 10 runs of area coverage for three Flocking-based approaches in a time interval $[0, 360000)$ milliseconds.

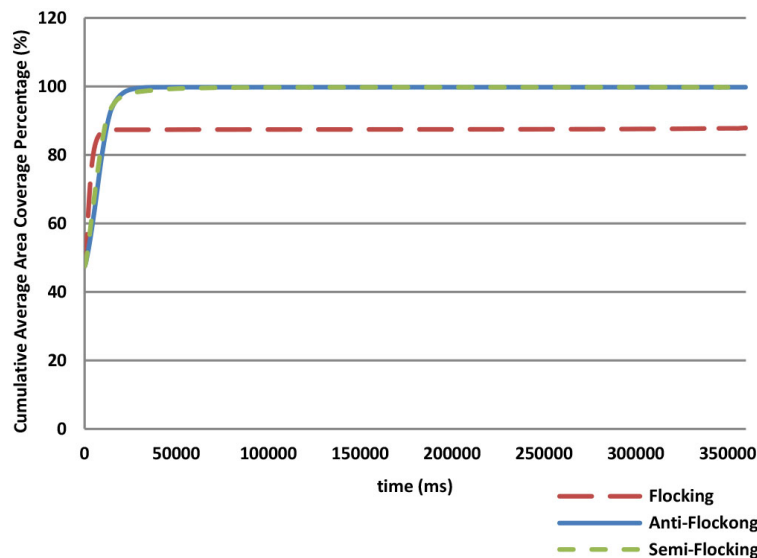


Figure 5.10: Cumulative Area Coverage (AC) in time interval $[0, 360000)$ (ms) in three Flocking-based algorithms for random moving targets

As shown in this figure, the cumulative area coverage increases over time for all three algorithms. However, the area coverage of the Anti-Flocking algorithm is higher than that of the two other algorithms most times. This occurs because the Anti-Flocking algorithm is almost a search algorithm and only emphasizes on increasing

area coverage. However the Semi-Flocking algorithm aims to strike a balance between the reliable area coverage and target coverage. Furthermore, as Figure 5.10 shows, the result of the Semi-Flocking algorithm, in area coverage, is extremely close to the result of the Anti-Flocking algorithm. The Flocking algorithm represents lower area-coverage results because it applies all the sensors for the creation of a flock and does not let any of them free to explore the AOI. Secondly, because it creates a big flock, in which most of the sensors revisit areas previously observed by their front sensors.

5.3.3.2 Walking Pedestrian Dataset

The results obtained from this part of the test are more reliable because this experiment is conducted over real data so the speed, position and the number of pedestrians (targets) in the scene varies more realistically. Table 5.1 illustrates the result of applying three Flocking-based algorithms on the walking pedestrian dataset [149]. This results include: average percentage of target coverage (TC), average target detection time (TDT), percentage of non-detected targets (PNDT). As represented in this table, the Semi-Flocking algorithm shows the highest percentage of target coverage, the lowest target detection time and fewer non-detected targets. These results soundly confirm previous results on random moving targets.

Figure 5.11 shows the result of area coverage for three Flocking-based approaches. As this figure illustrates, Anti-Flocking shows the best results, and the results of Semi-Flocking is very close to Anti-Flocking, and the Flocking algorithm has the worst performance over this parameter. These results are very similar to the results obtained for the random walking target data as represented in Figure 5.10.

Table 5.1: Results of walking pedestrians dataset

parameter \ algorithm	Semi-Flocking	Flocking	Anti-Flocking
Average percentage of target coverage(TC)	85.65	18.43	1.91
Average target detection time (TDT)	8317.73 ms	104751.52 ms	17466.37 ms
Percentage of non-detected targets (PNDT)	0	41.64	18.76

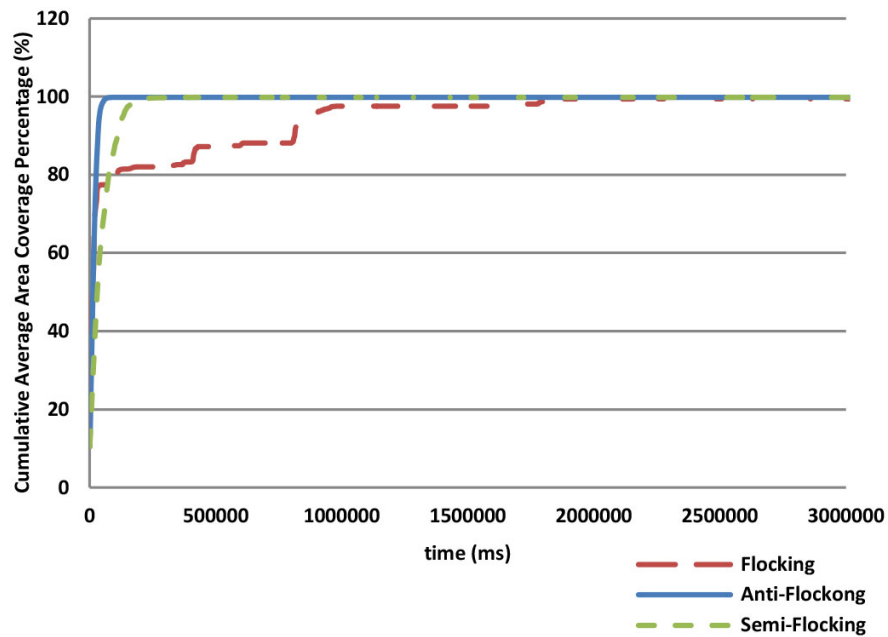


Figure 5.11: Cumulative Area Coverage (AC) in time interval $[0, 360000)$ (ms) in three Flocking-based algorithms for walking pedestrian dataset

Summarizing the results of the three Flocking-based algorithms over the four evaluated parameters shows that, in target coverage, the Semi-Flocking algorithm demonstrates un-

rivalled results, especially when the number of targets increase. In the area coverage, the results of the Semi-Flocking algorithm are very close to the results of the Anti-Flocking algorithm. Considering the objective function of surveillance systems which is optimizing both target and area coverage, the Semi-Flocking algorithm is the most suitable algorithm among Flocking-based algorithms for the management of sensors in this application.

5.4 Summary

This chapter introduced the Semi-Flocking algorithm, an approach for controlling the movement of mobile sensors in surveillance applications. The Semi-Flocking algorithm combines features of the Flocking and Anti-Flocking algorithms, and therefore inhabits a position between these two extremes. Next, Flocking, Anti-Flocking and Semi-Flocking algorithms were examined as guidance strategies, once for a set of mobile sensors tracking randomly moving targets in the surveillance system and once for a standard walking pedestrian dataset. It has been found that the target coverage, detection effectiveness and area coverage of these mobile sensors vary with the mobility-control algorithm used. The proposed Semi-Flocking algorithm exhibits outstanding performance in meeting the objectives of surveillance application in both test cases.

Chapter 6

Constrained Clustering for Flocking-based Tracking in Maneuverable Target Environments

Tracking maneuverable targets is one of the most challenging problems in self-organizing sensor networks. Although, as represented in the previous chapter, Flocking-based algorithms have demonstrated promising performance in tracking linear target(s), they have deficiencies in tracking maneuverable targets.

This chapter introduces a constrained clustering approach that uses a novel extension of K-means algorithm to provide better coverage over maneuverable targets. This extension clusters the sensors based on certain background knowledge, then uses the information about the clusters to improve coverage over maneuverable targets. The performance of Flocking-based algorithms, both with and without applying the proposed approach, are examined in tracking both linear and maneuverable targets. Experimental results demonstrate how constrained clustering yields better tracking of maneuverable targets, and how applying constraints on the clustering process improves the quality of clustering and increases the speed of convergence.

The outline of the chapter is as follows. Section 6.1 briefly introduces the problem of tracking maneuverable targets. Section 6.2 highlights the drawbacks of Flocking-based

methods for tracking maneuvering targets and introduces a constrained clustering based approach that addresses these drawbacks. Section 6.3 introduces the evaluation criteria, the experimental setup, and the simulation results and analysis. Finally, concluding remarks are given in Section 6.4.

6.1 Introduction

In a surveillance application, targets are normally classified into two classes based on their motion type: maneuvering (non-linear) and non-maneuvering (linear). A non-maneuvering target has a constant velocity. All other targets (those with non-constant velocity) are categorized as maneuvering targets [150]. One key challenge in large-scale surveillance systems is mobility control and coordination, which deals with the optimal movement of a set of mobile sensors. This problem is even more challenging when sensors are dealing with maneuverable targets that change their speed and direction frequently and suddenly [151, 22]. Extensive research has focused on this problem in recent years [152, 153, 146, 154]. The Flocking algorithm [152] is a well-cited example of this research work [5, 130, 131, 132, 133, 134]. Although Flocking-based algorithms have demonstrated promising performance in tracking mobile targets, they are not able to cover maneuvering targets as well as non-maneuvering ones, particularly when there is a small flock around a maneuvering target.

This chapter discusses the effectiveness of two Flocking-based algorithms, namely, Flocking [5] and Semi-Flocking (introduced in Chapter 5) in tracking maneuvering and non-maneuvering targets. It then argues the deficiencies of both algorithms when it comes to their ability to maintain reliable target coverage over maneuvering targets, then presents a novel constrained clustering approach that facilitates improved target coverage performance under complex target maneuvering conditions.

Constrained clustering is an approach that can be applied in applications in which some background knowledge about data sets is available. Traditional clustering approaches make no use of this information even if it does exist [155]. This prior information provides increased evidence as to which instances should or should not be placed in the same cluster. This information provides indispensable insight for forming more precise clusters and/or

increasing the rate of convergence in a clustering algorithm. It is maintained that cluster precision and clustering convergence are key requirements for dynamic multi-target tracking.

6.2 Tracking Maneuverable Targets by Flocking-based Algorithms

As represented in Chapter 5, the Semi-Flocking algorithm demonstrates good target coverage for non-maneuvering targets; however it is less effective in covering maneuvering targets. As mentioned, a maneuvering target changes its speed frequently and, as a result, it can escape from the covered area. Figure 6.1 shows this problem, illustrating that, in some cases the flock is behind the maneuvering target and in some cases it is forward. This problem is more challenging when a small flock of sensors is tracking a target. Bigger flocks better tolerate maneuvers of maneuvering targets. Although this problem is more sophisticated in the Semi-Flocking algorithm (because of the small flocks), the Flocking algorithm still suffers from the same problem.

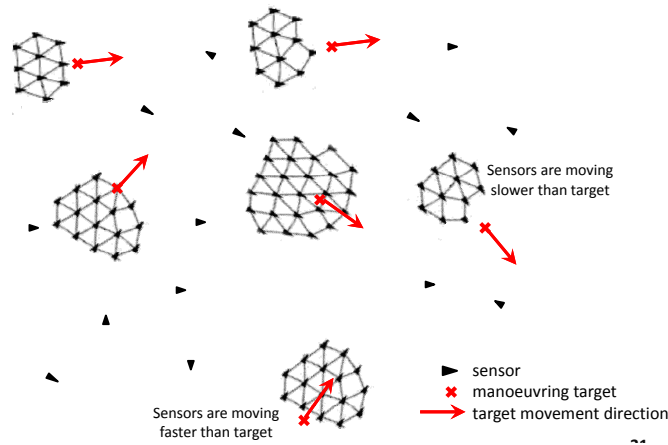


Figure 6.1: Mismatch of sensor speeds with maneuvering target speeds in the Semi-Flocking algorithm

6.2.1 Constraints Definition

This chapter presents a novel constrained clustering approach to tackle the problem of mismatch-speeds depicted in Figure 6.1. In this approach, a constrained clustering algorithm is applied to cluster the sensors based on their locations, using information available about the number, size and place of clusters to guide the clustering algorithm to converge faster, with less computational effort and with more precise clustering results.

An example of this background knowledge in Flocking-based methods is the number of clusters, which is equal to the number of targets in the Semi-Flocking algorithm, and is equal to 1 in the Flocking algorithm. We are not interested in empty clusters in this application. Furthermore, based on the number of sensors, there is an estimation of flock size boundaries, and these boundaries also restrict the size of clusters. These examples and other similar information create rich background knowledge about the clusters, and their size and place in the two-dimensional environment. This research aims to benefit from this knowledge for defining some constraints to be added to the clustering approach. The constraints should be defined such that all of the sensors of each flock are placed in the same cluster. Figure 6.2 represents a list of constraints that should be satisfied in this clustering.

As illustrated in this figure, six constraints are defined (lines 3-8 of Figure 6.2). The first constraint adjusts the number of clusters to the number of targets (line 3 of Figure 6.2). The second constraint restricts the initial cluster centers to target positions (line 4 of Figure 6.2). The third and fourth constraints restrict the size of each cluster to between zero and a threshold value α (lines 5-6 of Figure 6.2). Finally, the fifth and sixth constraints restrict the distance between the members of each cluster and the cluster center (lines 7-8 of Figure 6.2).

6.2.2 Constrained Clustering Approach

After clustering the sensors, the center of each cluster is computed (line 10 of Figure 6.2) and then the speeds of the sensors are adjusted based on the distance between the center of their cluster and their tracking target (lines 11-16 of Figure 6.2).

```

(1) Constrained Clustering procedure {
(2)   cluster the sensors based on following constraints:
(3)     1. number of clusters should be exactly the same as number of targets
(4)     2. Initial cluster centers should be target positions instead of random
        choices
(5)     3. none of the clusters should be empty
(6)     4. size of each cluster should be less than threshold  $\alpha$ 
(7)     5. elements of each cluster should be closest point to the center of the
        cluster
(8)     6. elements of each cluster should not be further than threshold  $\rho$ 
        from the center of the cluster (If the distance between a sensor and all
        the targets is greater than  $\rho$ , the sensor should be considered as an
        outlier and should not be included in any cluster)

(9)   for (each cluster i) do
(10)     $C_i =$  the center of cluster i
(11)     $\vec{v}_i = \overrightarrow{target_i - C_i}$  //targeti represents position of target i
(12)   end for
(13)   for (all the sensors in cluster i) do
(14)     $q_i = q_i + \vec{v}_i$ 
(15)    //adjust the position of each sensor to match the centre of cluster
        with the current position of the maneuver target
(16)   end do
(17) }

```

Figure 6.2: Constrained clustering approach to match the speed of sensors with a maneuvering target

As the sensors and targets are mobile and the configuration of sensor network changes over time, the clustering approach should be called once at predefined time intervals. The selected time intervals should not be so small as to impose high computation overhead on

the algorithm and not so large as to let the maneuvering target escape from the tracking sensors.

6.2.3 Constrained K-means Clustering

Although the constrained clustering approach proposed in this chapter is a general one, the K-means clustering algorithm [156] is selected and defined constraints are applied on that to obtain constrained clustering. This algorithm was selected because it is a well-known, effective and simple algorithm for large-scale clustering problems.

Given a data set $D = \{x^i\}_{i=1}^n$ of n points, and knowing that they should be partitioned into k clusters, the K-means clustering problem is as follows: find cluster centres C^1, C^2, \dots, C^k such that summation of distances between each point and its nearest cluster center C^h are minimized. This objective is represented in Equation 6.1 [157].

$$\min_{C^1, \dots, C^k} \sum_{i=1}^n \min_{h=1, \dots, k} \left(\frac{1}{2} \|x^i - C^h\|_2^2 \right) \quad (6.1)$$

It is mathematically provable [157] that Equation 6.1 is equivalent to the problem of Equation 6.2:

$$\min_{C, T} \sum_{i=1}^n \sum_{h=1}^k T_{i,h} \cdot \left(\frac{1}{2} \|x^i - C^h\|_2^2 \right) \quad (6.2)$$

Subject to:

$$\sum_{h=1}^k T_{i,h} = 1, \quad i = 1, \dots, n$$

$$T_{i,h} \geq 0 \quad i = 1, \dots, n, \quad h = 1, \dots, k$$

$T_{i,h}$ is equal to one if x^i is the closest point to C^h and is equal to zero otherwise. This problem is solved by the K-means algorithm iteratively.

Given the set of n data points D and cluster centers $C^{1,t}, C^{2,t}, \dots, C^{k,t}$ at iteration t , the K-means algorithm is employed to compute the cluster centers for iteration $t + 1$ ($C^{1,t+1}, C^{2,t+1}, \dots, C^{k,t+1}$) as follows:

- Each instance x^i is assigned to its closest cluster center h .
- Each cluster center $C^{h,t+1}$ is updated to be the mean of all instances assigned to cluster h .

The algorithm terminates when the following condition is satisfied:

$$C^{h,t} = C^{h,t+1}, \quad \text{for all } h = 1, \dots, k.$$

In other words, the algorithm terminates when all cluster centers become invariant in two consecutive iterations.

Based on this formulation, in the standard K-means algorithm, all the data points should be assigned to one and only one cluster (constraint $\sum_{h=1}^k T_{i,h} = 1 \quad i = 1, \dots, n$). In some applications, some data points are outliers and should not be assigned to any cluster. In Semi-Flocking algorithm, the sensors searching the AOI and not tracking any target should be considered as the outliers in the clustering algorithm and, as a result, should not be assigned to any of the clusters. In addition the standard K-means formulation does not handle the constraints defined in Section 6.2.1.

To avoid the described drawbacks, I propose to modify the constraints in Equation 6.2 so as to yield the following constrained K-means problem:

$$\min_{C,T} \sum_{i=1}^n \sum_{h=1}^k T_{i,h} \cdot \left(\frac{1}{2} \|x^i - C^h\|_2^2 \right) \quad (6.3)$$

Subject to:

$$\sum_{h=1}^k T_{i,h} \leq 1, \quad i = 1, \dots, n$$

$$T_{i,h} \geq 0 \quad i = 1, \dots, n, \quad h = 1, \dots, k$$

$$k = m \quad //\text{Constraint 1}$$

$$C^{h,1} = q_{th}, \quad h = 1, \dots, k \quad //\text{Constraint 2}$$

$$1 \leq \sum_{i=1}^n T_{i,h} \leq \alpha, \quad h = 1, \dots, k \quad //\text{Constraint 3,4}$$

$$\text{where } k \cdot \alpha \leq n$$

$$\text{if } T_{i,h} = 1 \quad \text{and} \quad T_{j,h} = 0 \quad \text{then} \quad \|x^i - C^h\|_2^2 < \|x^j - C^h\|_2^2,$$

$$i = 1, \dots, n, \quad j = 1, \dots, n, \quad h = 1, \dots, k \quad //\text{Constraint 5}$$

$$\text{if } T_{i,h} = 1 \quad \text{then} \quad \|x^i - C^h\|_2^2 < \rho \quad i = 1, \dots, n, \quad h = 1, \dots, k \quad //\text{Constraint 6}$$

Similar to the traditional K-means approach, the constrained K-means clustering algorithm iterates between assignment and update phases; however, the objective function and constraints in constrained K-means clustering (Equation 6.3) are different from those in traditional K-means (Equation 6.2).

Adding constraints to the K-means algorithm results in its faster convergence and also results in more precise clusters. Experimental results, presented in Section 6.3, confirm this claim.

6.3 Experiments and Discussion

6.3.1 Evaluation Parameters

The main objective of the proposed constrained clustering approach is to increase the target coverage in Flocking-based algorithms. Based on this objective, a Target Coverage (TC) parameter is applied to evaluate the results.

- **Target Coverage:** This parameter is calculated as described in Section 5.3.1 by Equation 5.5.

In addition to evaluating of TC, we need to know how adding defined constraints to the clustering algorithm increases the precision of clustering and also how it reduces the computations required in the clustering algorithm. To compare the proposed constrained K-means clustering approach with the traditional K-means algorithm in terms of optimality of clustering and computational effort, this chapter uses the following two evaluation parameters:

- **Davies-Bouldin Index (DBI):** DaviesBouldin index [158] is a metric for the evaluation of clustering algorithms. This metric aims to identify whether clusters are well separated and compact. For any partition $X : X_1 \cdots \cup X_i \cdots \cup X_k$, where X_i represents the i^{th} cluster of such partition, the DaviesBouldin index is defined as in Equation 6.4.

$$DBI(X) = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} \frac{\Delta(X_i) + \Delta(X_j)}{\delta(X_i, X_j)} \quad (6.4)$$

where k is the number of clusters, $\delta(X_i, X_j)$ represents the distance (in this research Euclidean distance) between cluster i and cluster j (inter-cluster distance), and $\Delta(X_i)$ represents the intra-cluster distance for cluster X_i . The smaller values for DBI represent clusters that are more compact and more separated from each other. Therefore, the smaller the value of DBI, the more optimal the clusters and the more efficient the clustering method. [158].

- **Number of Iterations to Convergence (NITC):** To evaluate this reduction, a parameter called Number of Iterations to Convergence (NITC) is defined. NITC shows the number of iterations that a clustering algorithm takes to converge. The higher the number of iterations, the higher the computation load of the algorithm.

6.3.2 Experimental Setup

A Java version of the Flocking and Semi-Flocking algorithms are implemented in the framework presented by Olfati-Saber [5]. This framework has been used to examine the efficiency of the proposed constrained clustering approach. The following parameters remained fixed through the implementation of all three Flocking-based algorithms: $d = 20$, $r = 1.2d$, $\epsilon = 0.1$ (for σ -norm), $a = b = 5$ for $\varphi(z)$, $h = 0.2$ for the bump function of $\varphi_\alpha(z)$, and 0.02 seconds the step-size in all simulations.

Maneuverable targets are implemented with the non-linear dynamics represented in Equation 6.5 [7]:

$$x(k+1) = A(x(k))x(k) + Bw(k) \quad (6.5)$$

where $x(k) = (q_1(k), p_1(k), q_2(k), p_2(k))^T$ denotes the state of the target at time k , $q = (q_1(k), q_2(k))^T$ represents the position of the target in a two-dimensional environment at time k and $p = (p_1(k), p_2(k))^T$ illustrates the velocity of a target in two dimensions. The target moves inside of a square field of k width and l height. The matrix $A(x)$ is defined as represented in Equation 6.6.

$$A(x) = M(x) \otimes F_1 + (I_2 - M(x)) \otimes F_2 \quad (6.6)$$

$$F_1 = \begin{bmatrix} 1 & \epsilon \\ 0 & 1 \end{bmatrix}$$

$$F_2 = \begin{bmatrix} 1 & \epsilon \\ -\epsilon c_1 & 1 - \epsilon c_2 \end{bmatrix}$$

$$M = \begin{bmatrix} \mu(x_1) & \epsilon \\ 0 & \mu(x_3) \end{bmatrix}$$

F_1 and F_2 determines the dynamicity of the target inside and outside of the region, respectively and \otimes represents the Kronecker product. $\mu(z)$ is a function for switching between F_1 and F_2 depending on target's position.

$$\mu(z) = \frac{\sigma(a+z) + \sigma(a-z)}{2}$$

$$\sigma(z) = \begin{cases} 1, & z \geq 0 \\ -1, & z < 0 \end{cases}$$

Matrix B determines the intensity of noise and is given by

$$B = I_2 \otimes G, \quad G = \begin{bmatrix} \frac{\epsilon^2 \sigma_0}{2} \\ \epsilon \sigma_0 \end{bmatrix}$$

where $\epsilon = 0.02$ is the step-size, a, c_1, c_2 are the parameters of the PD controller, which depends on the length and width of the field (k and l). The elements of $w(k)$ are normal zero-mean Gaussian noise with $Q = 100I_2$. To create various dynamicity levels, the value of σ_0 is varied from 0 to 40 in four unit steps $\sigma_0 \in [0, 40)$. The higher the value of σ_0 , the higher the maneuverability of the target. At the beginning ($\sigma_0 = 0$), the target has near to linear dynamics. Figure 6.3 represents the trajectory of targets with various levels of dynamicity in a 6-minute time interval.

In this experiment, the AOI of the surveillance system is a 1250×665 rectangle. There are 150 mobile sensor nodes in the system, and the detection radius of each sensor is $r = 30$. There are three maneuvering targets with the same maneuverability levels in the AOI. For each maneuverability level, 10 random instances were generated. The reported results are the average over these instances. All the targets are mobile. In all the instances, the initial position, entrance time, and lifetime of each target were selected randomly. The same random seeds were used to generate the same instances in all the algorithms. For

each instance, the monitoring time is continued for 360 seconds. After entering, a target moves randomly around until the end of its lifetime. It is assumed that a target is covered if it is in the field of view of at least 3 sensors ($k = 3$).

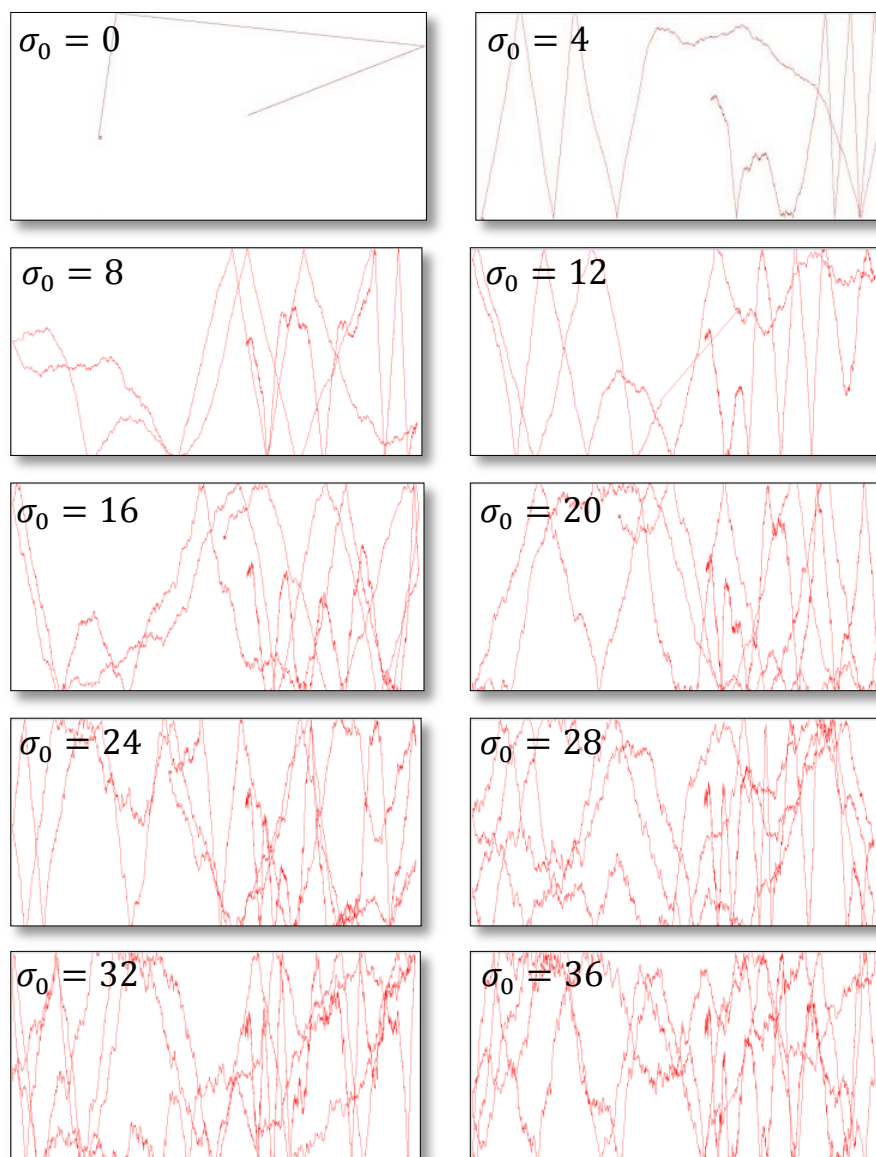


Figure 6.3: A maneuvering target trajectory with various levels of dynamicity ($\sigma_0 \in [0, 40)$ in four unit steps) in a time interval of $[0, 360000)$ milliseconds.

A constrained version of K-means algorithm is implemented in Java and called five times in each second of execution of a Flocking-based algorithm (Flocking or Semi-Flocking).

6.3.3 Simulation Results and Analysis

1. **Target Coverage:** Figure 6.4 shows the average of 10 runs of target coverage results in Flocking and Semi-Flocking algorithms. In this figure, all the targets are maneuvering and their maneuverability increases over time. This figure compares Flocking and Semi-Flocking algorithms with their improved versions, including the constrained clustering approach.

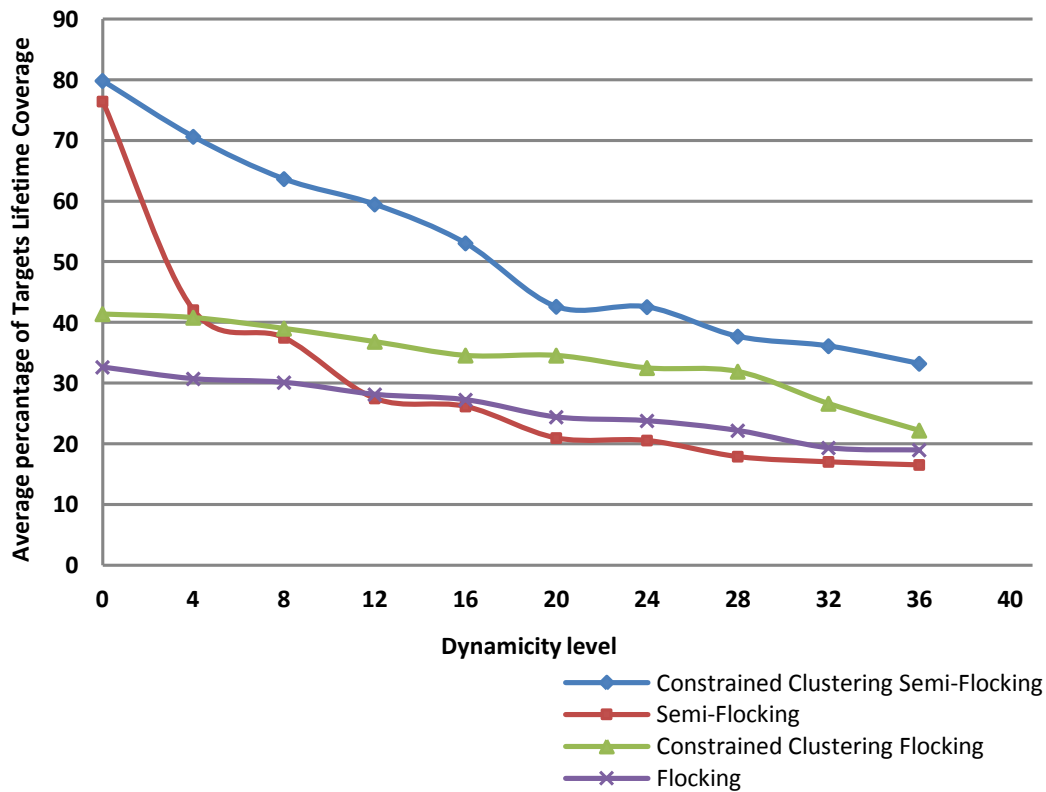


Figure 6.4: Average Target Coverage (TC) in Flocking and Semi-Flocking algorithms (with and without constrained clustering) for maneuvering targets.

As illustrated in Figure 6.4, both Flocking and Semi-Flocking algorithms fail to show acceptable results when targets are maneuvering, especially when the maneuverability level increases. In both of the algorithms, adding the proposed constrained clustering approach to these basic algorithms results in higher target coverage. In some cases, this improvement is more than 100 percent. These results clearly show the effectiveness of applying the constrained clustering approach to the Flocking and Semi-Flocking algorithms.

The average TC of maneuvering targets drops more using the Semi-Flocking algorithm than in the Flocking algorithm. This result is due to the difference between the flock sizes in Flocking and Semi-Flocking algorithms. The Flocking algorithm creates a big flock around just one of the targets, so even if the target is a maneuverable one it cannot escape from the sensors easily.

2. **Davies-Bouldin Index (DBI):** Figure 6.5 shows the average of 20 runs of DBI resulting from clustering the sensors in Semi-Flocking algorithm by increasing the number of targets. This figure compares the DBI of simple K-means clustering (clustering without considering the constraints defined in Section 6.2) and DBI of constrained K-means clustering (clustering considering the constraints defined in Section 6.2). As illustrated in this figure, the value of DBI in constrained K-means is much smaller than its value in simple K-means. The smaller values of DBI in constrained clustering approach shows that clusters created by this method are more separated and more compact than simple clustering ones. These results reveal the positive impact of adding constraints to the traditional K-means clustering algorithm to create better and more-precise clusters.

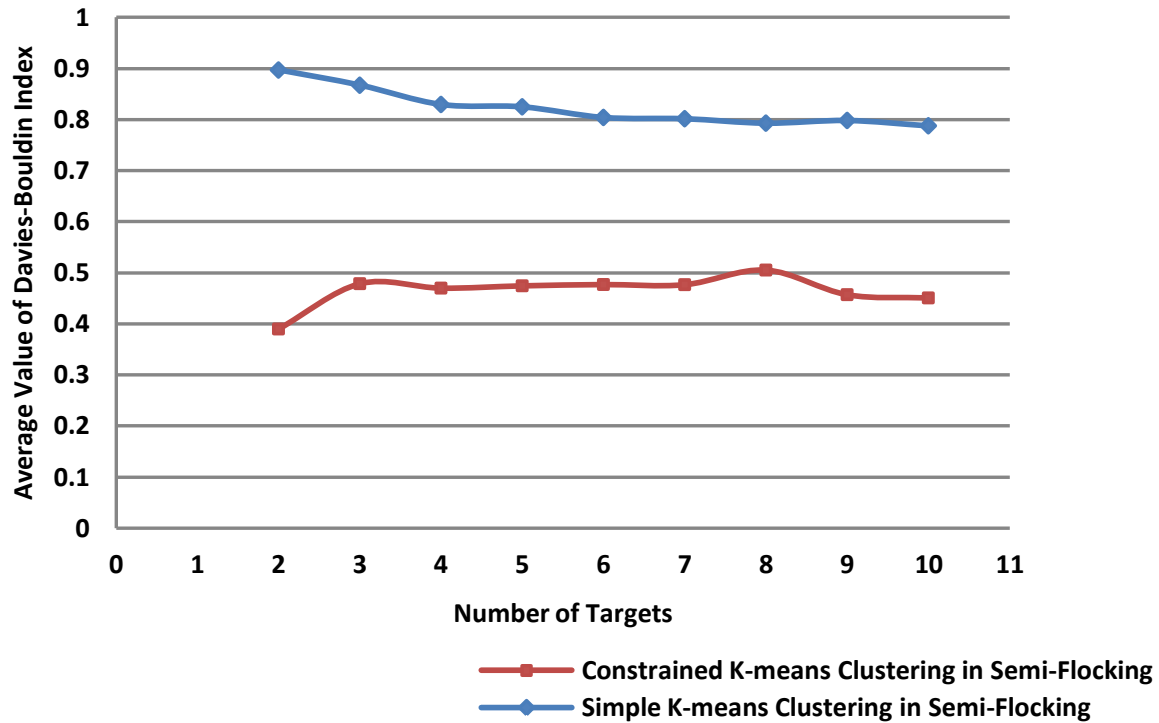


Figure 6.5: Davies-Bouldin Index of constrained and simple K-means clustering in Semi-Flocking approach.

- 3. Number of Iteration to Converge (NITC):** Figure 6.6 shows the average of 20 runs of NITC resulting from the Semi-Flocking algorithm. Also this figure compares the NITC of simple K-means clustering (clustering without considering constraints defined in Section 6.2) and NITC of constrained K-means clustering (clustering with considering the constraints defined in Section 6.2) by increasing the number of targets. As illustrated in this figure, in both of the clustering methods (constrained/simple), as the number of clusters increases, the number of iterations to converge K-means algorithm increases because the problem becomes more complex and requires higher computational effort.

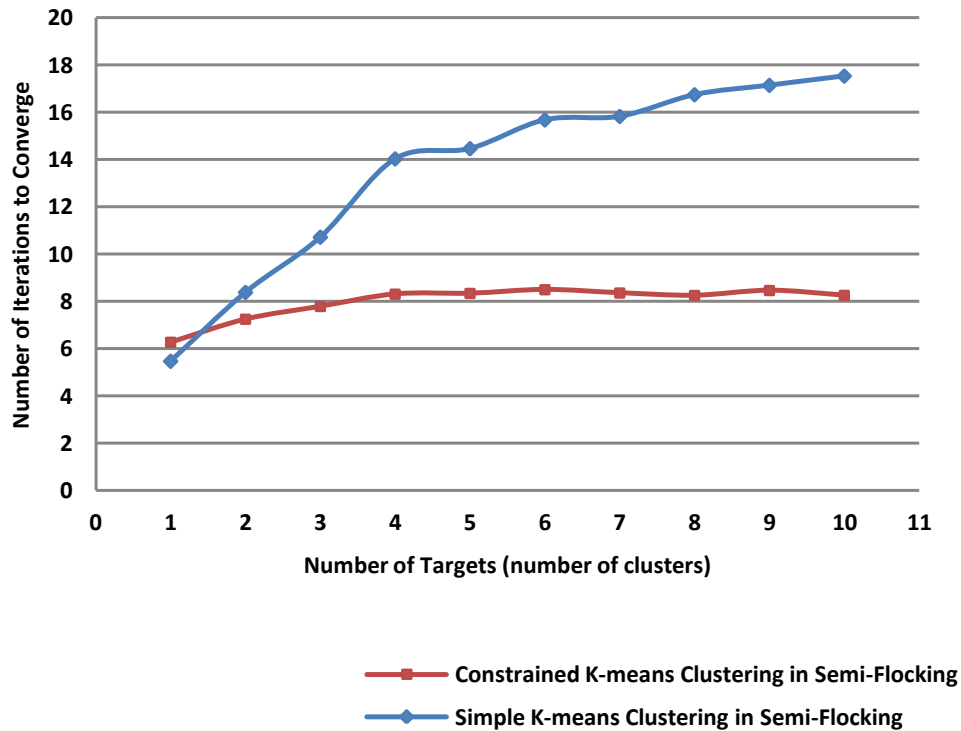


Figure 6.6: Number of iterations of K-means algorithm to converge when it is applied for constrained clustering of sensors in the Semi-Flocking approach.

A significant advantage of this approach is that using constraints greatly decreases the computational expose of clustering: defined constraints help the K-mean algorithm to find the clusters in fewer iterations.

6.4 Summary

This chapter has discussed the effectiveness of Flocking and Semi-Flocking algorithms in tracking maneuverable targets in a surveillance application. Evaluations demonstrated that Flocking-based algorithms are not able to track maneuverable targets perfectly, because maneuverable targets change their speed and direction frequently and Flocking sensors do not have adequate time to adjust their speed accordingly. A novel approach based on

constrained clustering is proposed to tackle this problem. This method clusters the sensors in predefined time slots and then adjusts the speed of the sensors based on the difference between the center of the cluster and the actual position of the tracking target.

The K-means algorithm, as a well-known clustering technique, was selected for applying the constrained clustering method. Then a set of constraints related to size, location and number of clusters were defined and added to the K-means algorithm to create a constrained K-means clustering method. Both (constrained and simple) clustering techniques were added to the Flocking and Semi-Flocking algorithms for tracking both simple and maneuvering targets. It has been found that, first, the proposed clustering approach greatly increases target coverage in Flocking-based methods, especially for maneuverable targets. Second, adding constraints to a K-means clustering approach creates much more-precise, more-compact and more-separate clusters. Third, adding constraints decreases the convergence time and iterations of the K-means algorithm. These results confirm the impact of appropriate definition of constraints in increasing the quality and decreasing the computational cost of clustering.

Chapter 7

Reliable Collaborative Multi-target Tracking Using Semi-Flocking Sensor Networks

An important concern in tracking multiple targets in sensor networks is the ability of sensors to self-organize so as to reliably adaptively cope with target dynamics, environmental changes, and measurement conditions. To address the problem of robust multiple target tracking using a sensor network, a coupled distributed estimation and motion control approach is required. As discussed in previous chapters, biologically inspired approaches, especially Flocking-based ones, have demonstrated successful performance as a tool to address complex motion tracking and control. The Flocking algorithm can be combined with distributed estimation approaches such as the well-celebrated continuous Kalman-Consensus Filter (KCF). Generally speaking, although this combination has demonstrated promising performance in single target applications, it is incapable of dealing with multi-target situations.

This chapter proposes a framework in which the Semi-Flocking algorithm is employed for multi-target motion control, and KCF for motion estimation. The Semi-Flocking algorithm approaches the problem by assigning a small flock of sensors to a single target, while at the same time leaving some sensors free to explore the environment to discover

new or uncovered targets. In the proposed coupled approach, each small group of flocking sensors (a Semi-Flock) applies a separate KCF algorithm to estimate the position of its target. This approach allows sensors to collectively provide reliable target engagement and comprehensive area coverage. This chapter reports experimental results to demonstrate how this distributed estimation-control algorithm can successfully track multiple targets under a wide range of target dynamics in a noisy sensor network.

The outline of the chapter is as follows. A brief introduction to the problem is presented in Section 7.1. Section 7.2 focuses on distributed tracking of targets in sensor networks under a noisy environment and also discusses current solutions and highlights their limitations. Section 8.2 describes the proposed coupled Semi-Flocking-KCF algorithm. Section 7.4 presents the performance-evaluation parameters, experimental setup, simulation results and analysis. Finally, concluding remarks and recommendations for future work are given in Section 7.5.

7.1 Introduction

In a surveillance system, self-organizing mobile sensors cooperate and coordinate their activities to collect information about targets (events) in a given volume of interest (VOI); they then fuse this multi sensor information to obtain a complete picture of the environment. An important feature, in such applications, is that the sensory information is inherently noisy, and it is imperative that this noise be reduced. Most of the research in this area has focused on centralized algorithms. Centralized Kalman Filtering is one of the most successful algorithms to attract significant attention [159]. As with most of the centralized algorithms, data is transmitted to a central sink, causing issues such as data congestion, limited scalability, and poor reliability. A number of distributed versions of this algorithm have emerged to mitigate these issues [160, 161, 162, 163, 164]. KCF is one such version. This algorithm was introduced in [163] and was subsequently used to track multiple targets using a network of cameras [164].

Another important challenge in self-organizing surveillance systems is the control and coordination of sensor mobility. This problem concerns the optimal movement of a set of

mobile sensors to achieve maximum area and (or) target coverage [19]. Previous chapters have demonstrated the successfulness of Flocking-based algorithms in the field of sensor control and coordination in sensor networks [137, 129, 5, 130, 131, 161, 133]. Various modes of flocking have been reported, including simple Flocking, Anti-Flocking and Semi-Flocking. These algorithms possess a biologically inspired behaviour that embodies a form of cooperation among a large number of autonomous interacting agents, the goal of which is to achieve coordinated group behaviour. To conceive a Flocking-based behaviour, each agent follows a set of rules and maintains some sort of communication with its neighbouring agents. Self-organization and local communication requirements of the Flocking-based algorithms provide an outstanding behaviour in the management of sensors in mobile sensor networks. The Semi-Flocking algorithm has been introduced in Chapter 5 as an alternative to the simple Flocking-based algorithms. This algorithm has demonstrated effective self-organizing capabilities in multi-target mobile sensor networks and, as a result, has reliable surveillance performance.

This chapter proposes combining KCF with Semi-Flocking to achieve distributed coupling of target motion estimation and motion control of the sensors. Noisy sensor measurements and multiple targets in the area to be under surveillance system are assumed. In [7], Olfati-Saber combines KCF with simple Flocking for the purpose of target tracking. Although, this coupled approach demonstrated success in single target tracking situations, it is incapable of tracking multiple targets. This limitation is inherently due to the Flocking algorithm used in this coupled approach. In contrast to Flocking, Semi-Flocking is designed to create small flocks of sensors around each target, hence its ability to achieve reliable target coverage in a multi-target situation, while still leaving some sensors free to search the surveillance area for uncovered new targets to achieve dynamic area coverage. It is conceivable, therefore, to combine this algorithm with KCF to realize coupled multi-target tracking and sensor motion control.

7.2 Distributed Tracking of Targets in Sensor Networks Under Noisy Environment

Although Flocking-based algorithms have demonstrated promising results in the motion control of sensors, they lack the ability to handle noisy measurements. Consequently, the sensor network is unable to accurately estimate the motion parameters of the target to determine its location and velocity. Furthermore, the sensors cannot take advantage of each other's measurement to minimize the impact of noise on their sensing performance. This limitation hinders their applicability in implementing sensor motion control strategies.

Most of the research work reported in this field uses centralized algorithms for reliable target tracking [159]. Centralized Kalman Filtering plays a crucial role in such target tracking algorithms. Centralized algorithms work in some applications, however, distributed algorithms have several advantages over centralized ones in distributed environments. Scalability, efficiency, robustness, reliability and autonomy are just a few examples of these advantages. Several studies have been done in the field of distributed target tracking [160, 132, 162, 164]. The KCF algorithm is one of the most successful completely distributed versions of the Centralized Kalman Filtering [163]. In this algorithm, each sensor uses its local information to compute a local estimate of the position of the target, then the local estimates will be forwarded to nearby sensors, Finally, the receiving sensors fuse the data and update their local estimates. It is proved that using this algorithm guarantees that all mobile sensors involved in estimating the state of a target reach a consensus.

In [7], the KCF algorithm and the Flocking are combined to form an estimation-control algorithm. This combination has been proven effective. The flock of sensors produced by the Flocking algorithm constitutes an appropriate connected network of sensors, in the sense that it enables the KCF algorithm to converge rapidly. In addition, the Flocking algorithm avoids collisions amongst sensors. A framework is proposed for analysing the performance of mobile sensor networks employing the Flocking algorithm as the mobility control model and the KCF algorithm for collaborative tracking of a single target [7].

This algorithm suffers from the limitation that it is applicable only to one-target situ-

ations. This restriction is mainly due to the Flocking algorithm’s inability to track more than one target. As described in Section 2.4.2.1, the Flocking algorithm creates a large flock of sensors around one target and leaves other targets uncovered. Therefore, when the Flocking algorithm is combined with the KCF, the combination inherits this limitation from the Flocking algorithm.

This chapter proposes a new combination of Flocking-based and KCF algorithms to address the problem of tracking of multiple targets. Furthermore, the development of this strategy is carried out based on noisy sensor measurements. The main idea behind this new combination is to use the Semi-Flocking algorithm in conjunction with the KCF algorithm to overcome the limitation of the Flocking algorithm in tracking more than one target. The Semi-Flocking algorithm, by its nature, is able to track multiple targets. Therefore, it is conceivable that combining the Semi-Flocking algorithm with the KCF algorithm will enable the sensors to track multiple targets. Section 8.2 describes this new combination.

7.3 Coupled Semi-Flocking and KCF Algorithm

This section describes how the Semi-Flocking algorithm can be combined with KCF [163]. Figure 7.1 shows the discrete-time version of KCF algorithm that has been used in this combination.

In this algorithm, each sensor makes a local estimate of the state of the target and broadcasts the estimate to its neighbours. The receivers fuse the received information, compute the Kalman-Consensus state estimate, and then update their micro-filters. This algorithm builds upon converging to a consensus estimate calculated by distributed local Kalman Filters rather than distributed averaging-based Kalman Filtering. It is mathematically provable that all estimators asymptotically will reach a consensus, i.e., $\hat{x}_1 = \dots = \hat{x}_n = x$ [7].

Algorithm 1 Kalman-Consensus Filter (message passing during one cycle at time index k for node i)

Given P_i, \bar{x}_i , and messages $m_j = \{w_j, W_j, \bar{x}_j\}, \forall j \in J_i = N_i \cup \{i\}$,

1. Obtain measurement z_i with covariance R_i .
2. Compute information vector and matrix of node i .

$$w_i = H_i^T R_i^{-1} z_i$$

$$W_i = H_i^T R_i^{-1} H_i$$

3. Broadcast message $m_i = \{u_i, U_i, \bar{x}_i\}$ to neighbors.
4. Receive messages from all neighbors.
5. Fuse information matrices and vectors

$$y_i = \sum_{j \in J_i} w_j, \quad S_i = \sum_{j \in J_i} W_j$$

6. Compute the Kalman-Consensus state estimate

$$M_i = (P_i^{-1} + S_i)^{-1},$$

$$\hat{x}_i = \bar{x}_i + M_i(y_i - S_i \bar{x}_i) + \mu F_i G_i \sum_{j \in N_i} (\bar{x}_j - \bar{x}_i)$$

$$\mu = \epsilon / (1 + \|F_i G_i\|), \quad \|X\| = \text{tr}(X^T X)^{1/2}$$

$$F_i = I - M_i S_i,$$

$$G_i = A M_i A^T + B Q B^T + P_1 S_1 P_1$$

7. Update the state of the Microfilter (x^+ is the updated x)

$$P_i^+ = A M_i A^T + B Q B^T$$

$$\bar{x}_i^+ = A \hat{x}_i$$

Figure 7.1: Kalman-Consensus Filter [7]

Coupled Semi-Flocking and KCF Algorithm (for a multi-target problem with m targets): Let $\hat{x}_{i,j} = \text{col}(\hat{q}_{i,j}, \hat{p}_{i,j})$ be the estimation of the state of target j by mobile sensor i using the KCF algorithm represented in Figure 7.1. Note that in this coupled algorithm, $\hat{x}_{i,j}$ will be calculated only for the targets that are within distance θ from the sensor i . Therefore as represented in Figure 7.2, sensors use multiple KCFs - one for each target. In this example, the sensors are tracking three targets. The Semi-Flocking algorithm creates three flocks of sensors, each around one target. Each flock uses its special KCF algorithm to

estimate the position of a related target and continues KCF until a consensus is reached on the position of that target. Figure 7.3 shows how the Semi-Flocking algorithm is combined with the KCF algorithm. In this figure, S_i shows the set of sensors tracking target i , and \hat{T}_i represents the new estimates of the state of the target by the KCF algorithm.

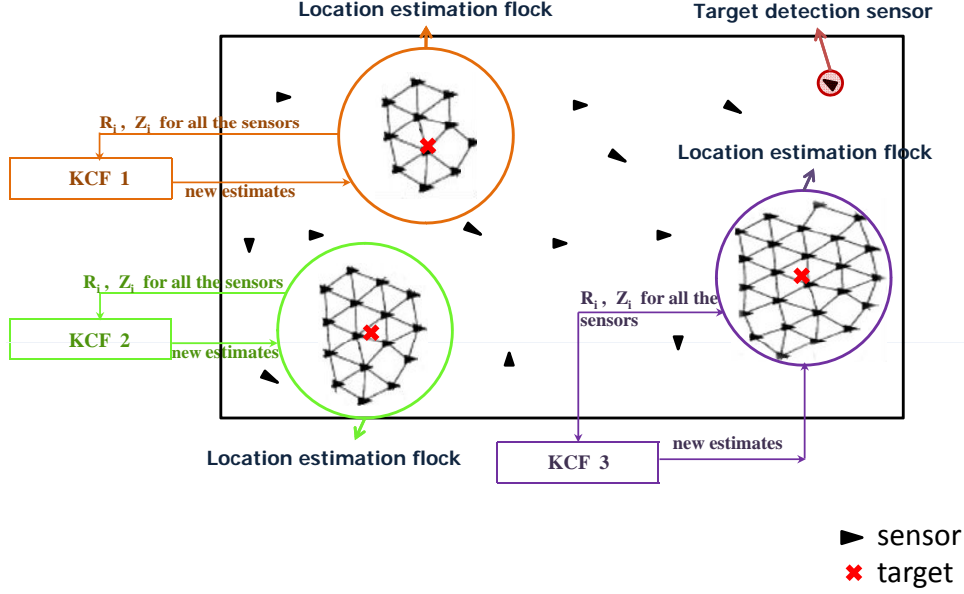


Figure 7.2: Coupled Semi-Flocking and KCF

More formally, in the coupled Semi-Flocking and KCF algorithm, each sensing agent α_i , with the dynamics represented in Equation 3.1, applies a control input vector, *i.e.*, $u_i = f_i^g + f_i^d + f_i^\gamma$, in which the first two terms are related to the three Reynolds rules: flock entering, collision avoidance and velocity matching, and the third term (f_i^γ) is navigational feedback. This control input function is represented in Equation 7.1.

$$u_i = \sum_{j \in N_i} \phi_\alpha(\|q_j - q_i\|_\alpha) n_{ij} + \sum_{j \in N_i} a_{ij}(q)(p_j - p_i) + f_i^\gamma \quad (7.1)$$

where f_i^γ is a navigational feedback that attracts sensors toward one of the surrounding target estimates (\hat{x}_{ij}), or alternatively, frees them to search the AOI to look for new targets. Sensors that are closer than a threshold θ to the estimate of that sensor from a target (\hat{x}_{ij})

apply Equation 7.2 as the navigational part. Refer to [5] for the definition of n_{ij} , ϕ_α and a_{ij} .

$$f_i^\gamma(q_i, p_i, \hat{q}_{t1}, \hat{q}_{t2}, \dots, \hat{q}_{tm}) = \sum_{j=1}^m \frac{c_{1j}(\hat{q}_{tj} - q_i)}{n_{tj}} + \sum_{j=1}^m \frac{c_{2j}(\hat{p}_{tj} - p_i)}{n_{tj}} = \sum_{j=1}^m \frac{c_{1j}(\hat{q}_{tj} - q_i) + c_{2j}(\hat{p}_{tj} - p_i)}{n_{tj}} \quad (7.2)$$

where c_{1j}, c_{2j} are positive constant values, and n_{tj} represents the number of sensors currently tracking target tj . On the other hand, if none of the target estimates are close to sensor i (within distance θ), the sensor searches the AOI to detect incoming targets using Equation 7.3.

$$u_i^\gamma = c \times (q_{w,l} - q_i) \quad (7.3)$$

where c is a positive constant value that adjusts the size of the vector, and $q_{w,l}$ represents the center of the adjacent area that has been least visited.

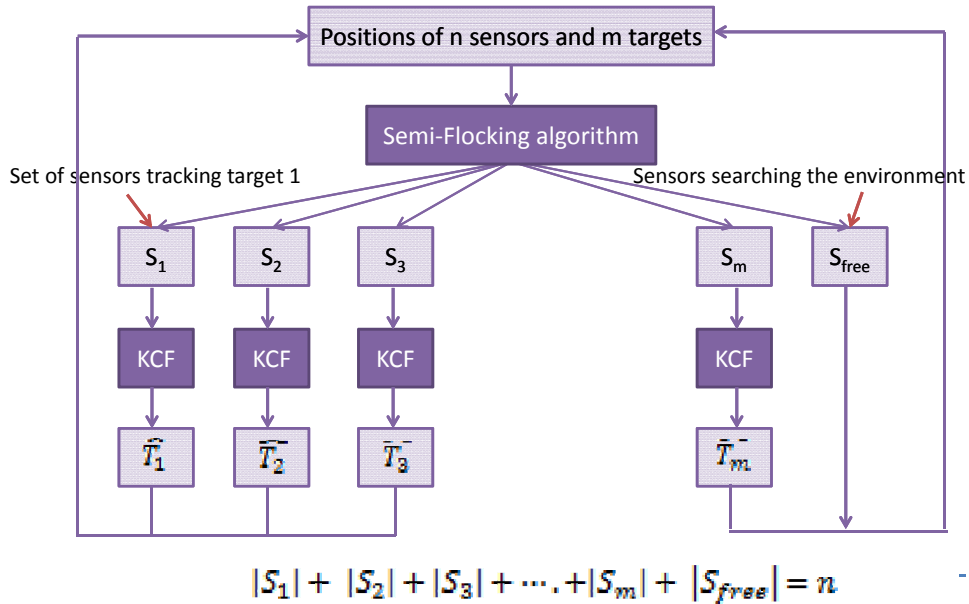


Figure 7.3: Combination of Semi-Flocking and KCF algorithms

It is important to notice that in this coupled algorithm the set of sensors that are participating in each KCF is dynamic. Thus, the participating sensors may change with time. As the targets and sensors are mobile, they will change their position which in turn changes the distance between them. As the role of sensors (tracking/searching) in the Semi-Flocking algorithm strongly depends on the distances between the sensors and targets, they may change their role based on the new positions of the sensors and targets. Therefore, the group of sensors tracking a target may change with time. New members may be added to the group or may be removed from it. This fact is referred to as the dynamicity of the group members in the Semi-Flocking algorithm.

In the coupled algorithm, this dynamicity results in changing the set of sensors cooperating in KCF algorithm. New members may be added to the group or may be leave that. Experimental results show that the few number of changes occur in semi-flock's memberships do not have a significant side effect on the convergence of KCF algorithm. Because most of the group members remain unchanged during continuous cycles and they are enough to converge the estimates. The new members start estimating the position of the target from scratch, but after a few rounds their estimate will be converged to the group consensus estimate.

7.4 Experiments and Discussion

7.4.1 Experimental Setup

In this section the proposed estimation-control algorithm (Semi-Flocking+KCF) for tracking two types of targets in a multi-target application is examined. In the experimental setup of this research, the area of interest (AOI) is a 1250×665 rectangle and all the targets and sensors remain in this area for all time $t \geq 0$.

Both parts of the estimation-control algorithm (Semi-Flocking and KCF [7]) are implemented in Java. The parameters of the Semi-Flocking algorithm are set as follows: $d = 70$, $r = 1.2d$, $\epsilon = 0.1$ (for σ - norm), $a = 8b$, $b = 1$ for $\varphi(z)$, $h = 0.2$ for the bump function of $\phi_\alpha(z)$. For the KCF algorithm, $P_0 = 100I_4$, $x_0 \sim (0, \sigma^2 I_4)$ with $\sigma = 60$, and $Q = 100I_2$.

There are 150 mobile sensors and 3 targets in the system, and the detection radius of each sensor is 30. The results reported are the average over 10 random generated instances. All the targets are mobile. In all instances, the initial position and lifetime of each target were selected randomly from a uniform distribution. For each instance, the monitoring time is continued for 300 sec (5 minutes).

The dynamics of targets for the case in which they are within the AOI (and are not on the borders) is represented in Equation 7.4. The parameters of the targets' dynamics and sensor measurements are kept the same as those in the experimental work reported in [7]. Doing so allows us to compare the results of the proposed coupled algorithm for tracking multiple targets with the results of the coupled algorithm presented in [7] for tracking a single target.

$$x(k+1) = Ax(k) + Bw(k) \tag{7.4}$$

with

$$A = \begin{bmatrix} 1 & 0 & \epsilon & 0 \\ 0 & 1 & 0 & \epsilon \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{\epsilon^2}{2} & 0 \\ 0 & \frac{\epsilon^2}{2} \\ \epsilon & 0 \\ 0 & \epsilon \end{bmatrix}$$

where $x(k) = (q_1(k), q_2(k), p_1(k), p_2(k))^T$ denotes the state of a target at time k , and $\epsilon = 0.02$ seconds is the step-size. To keep a target within the AOI, the target reverses the proper direction at the borders. The sensors within a suitable distance from each target make noisy measurements of the position of targets, *i.e.*

$$z_i(k) = H_i(k)x(k) + v_i(k);$$

$$H_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Both $w(k)$ and $v_i(k)$ are zero-mean Gaussian signal noises with the following statistics:

$$E[w(k)w(l)^T] = Q_k \delta_{kl}$$

$$E[v_i(k)v_j(l)^T] = R_i(k) \delta_{kl} \delta_{ij}$$

where $\delta_{kl} = 1$ if $k = l$, and $\delta_{kl} = 0$, otherwise. The measurement error covariance matrix for sensor i ($R_i = \frac{2}{f(\rho_i)} I_2$) is calculated based on the model of information value in [132]. Where $f(\rho_i)$ is the information value function,

$$I_i = f(\rho_i) = 2I_0 \left(a + b + (a - b) \frac{\rho_i - l}{\sqrt{1 + (\rho_i - l)^2}} \right)^{-1} \quad (7.5)$$

where $\rho_i = \|H_i \bar{x}_i - q_i\|$, $I_0 = 0.001$, $a = 8b$ and $b = 1$.

7.4.2 Evaluation Parameters

To evaluate the proposed method, three parameters were selected and used in this experiment. These parameters are standard statistic metrics that have been used in previous papers to evaluate similar algorithms.

- **MSE:** represents the mean square error between the fused estimated position of the target and its real position. MSE is an appropriate metric for evaluation of the success of the proposed method because it shows how accurate the estimates in the proposed algorithm are and, as a result, how successful it is in tracking the target.
- **Average info value:** represents the average of information value of all the sensors participating in tracking a target. The information value for each sensor is calculated by Equation 7.5.

- **Algebraic connectivity:** The algebraic connectivity of a graph G is the second-smallest eigenvalue of the Laplacian matrix of G . This eigenvalue is greater than 0 if and only if G is a connected graph. In this application the connectivity of the location estimate flock of sensors around each target is evaluated.

7.4.3 Simulation Results and Analysis

Figure 7.4 shows a snapshot of the proximity structure for 150 sensors applying the coupled Semi-Flocking and KCF algorithm after a few seconds of starting the algorithm.

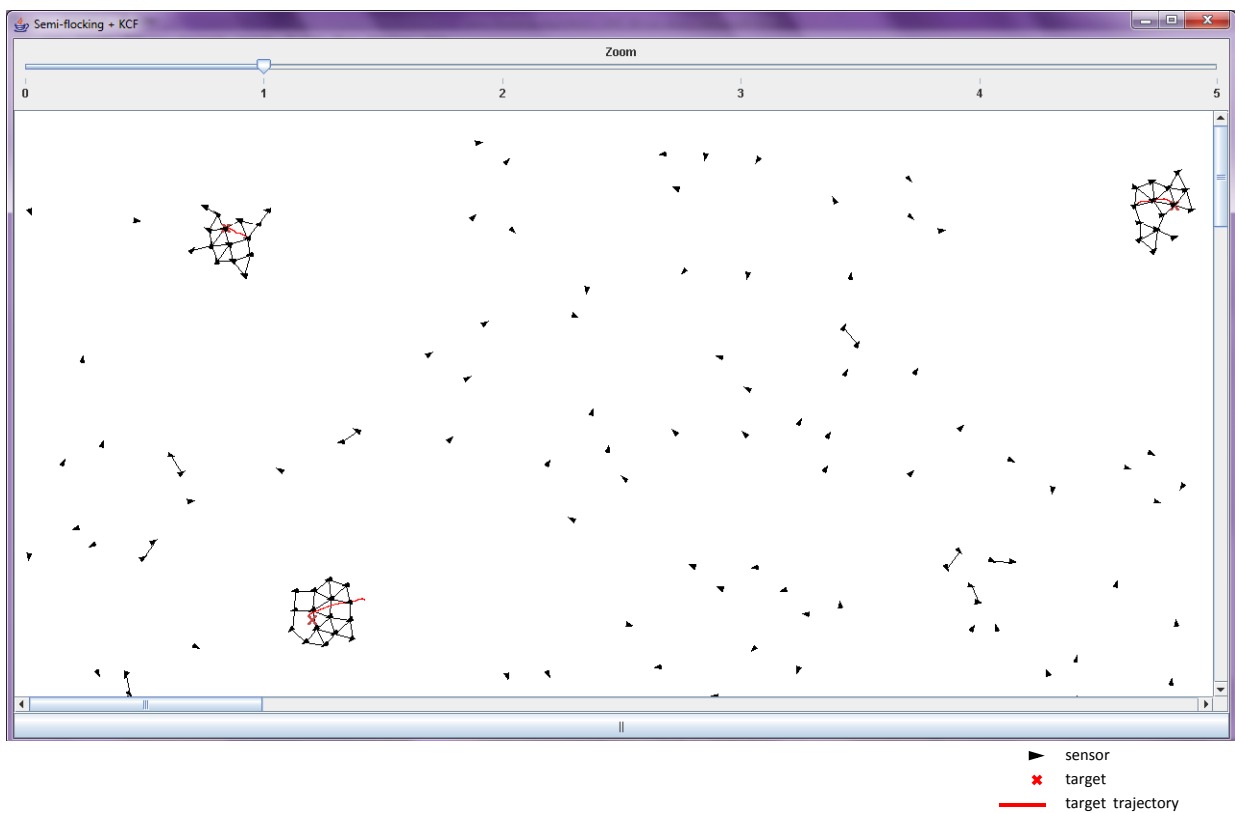


Figure 7.4: Snapshot of 150 sensors executing the coupled Semi-Flocking and KCF algorithm

The sensors are supposed to track 3 mobile targets in this example. As this figure demonstrates, a small flock is formed and maintained around each target. At the same time, some free sensors are exploring the AOI. The observations are in close agreement with our expectation of the behaviour of this coupled algorithm, which is depicted in Figure 7.2.

The results of evaluating the proposed algorithm are as follows:

1. **MSE:** Figure 7.5 represents the average mean square error of 10 runs of the proposed estimation-control algorithm. The results of the MSE are represented for three tracking targets in this figure and show that, for all the targets, the mean square error decreases with time. Thus in all of the KCFs used in semi-flocks, as time goes on, the accuracy of estimates increases and the group is able to create better estimates. The better the estimates in the KCF part, the better the tracking quality in the Semi-Flocking part of the coupled algorithm. The results for three targets in this experiment match with the results reported in [7] for one target.

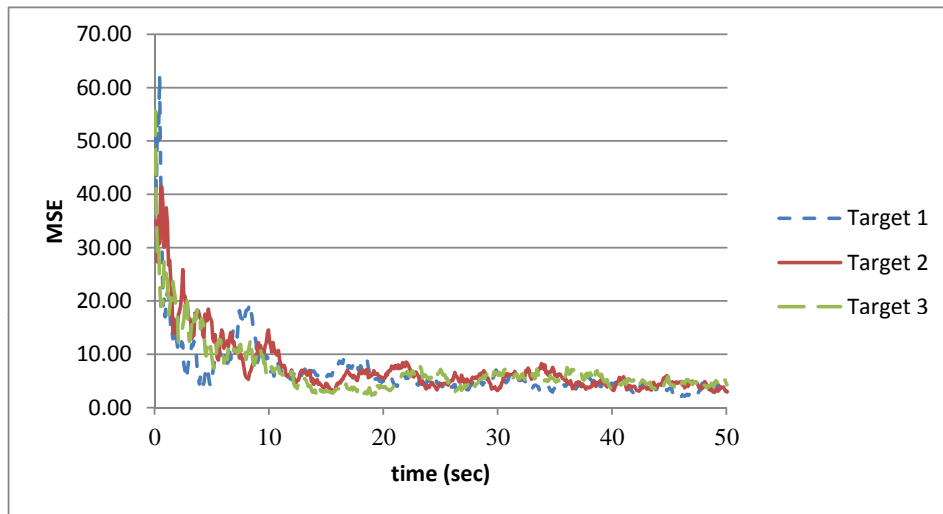


Figure 7.5: Average mean square error in tracking three targets in the estimation-control Semi-Flocking and KCF algorithm

2. **Average info value:** Figure 7.6 represents the average information value of the Flocking sensors in 10 runs of the proposed estimation-control algorithm. There are

two kinds of sensors: location estimate and target detection sensors. Only the first group participates in tracking an already detected target, creates a semi-flock, and applies the KCF algorithm. The information value is only meaningful for location-estimation sensors that apply the KCF algorithm. Therefore, the results reported show the average information value among all of the Flocking (location estimate) sensors.

The results show that the information value among the Flocking sensors increases over time, so the combination of Semi-Flocking and KCF algorithms is successful in increasing the information about the targets. The results also show that the average information value converges to $I_0 = 0.001$. These results exactly match the results reported in [7] for one target, in which the average information value converges to the value that was selected for I_0 in that paper ($I_0 = 0.1$).

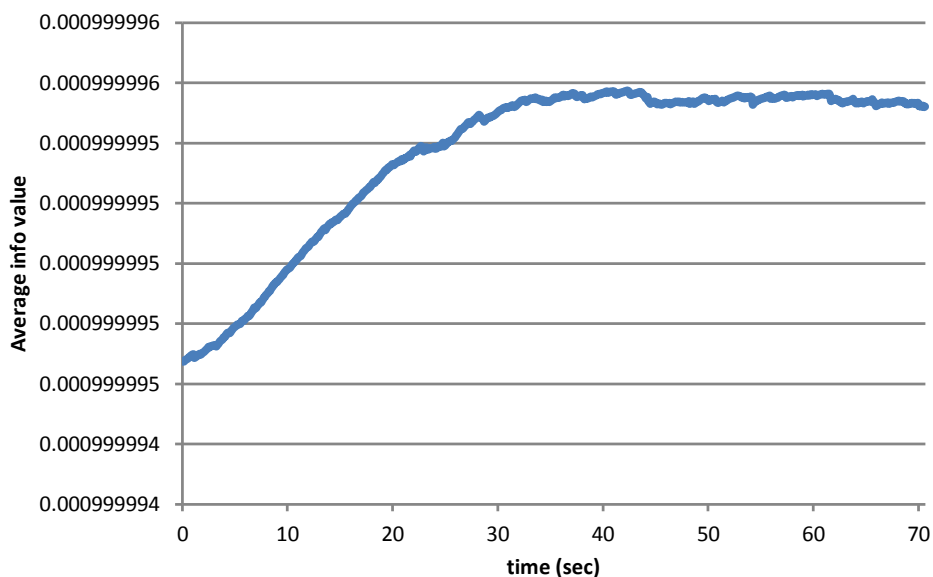


Figure 7.6: Average information value of Flocking (location estimate) sensors in tracking three targets in the estimation-control Semi-Flocking and KCF algorithm

- Algebraic connectivity:** Figure 7.7 represents the average algebraic connectivity of three networks of sensors tracking three targets in ten runs of the proposed

estimation-control algorithm.

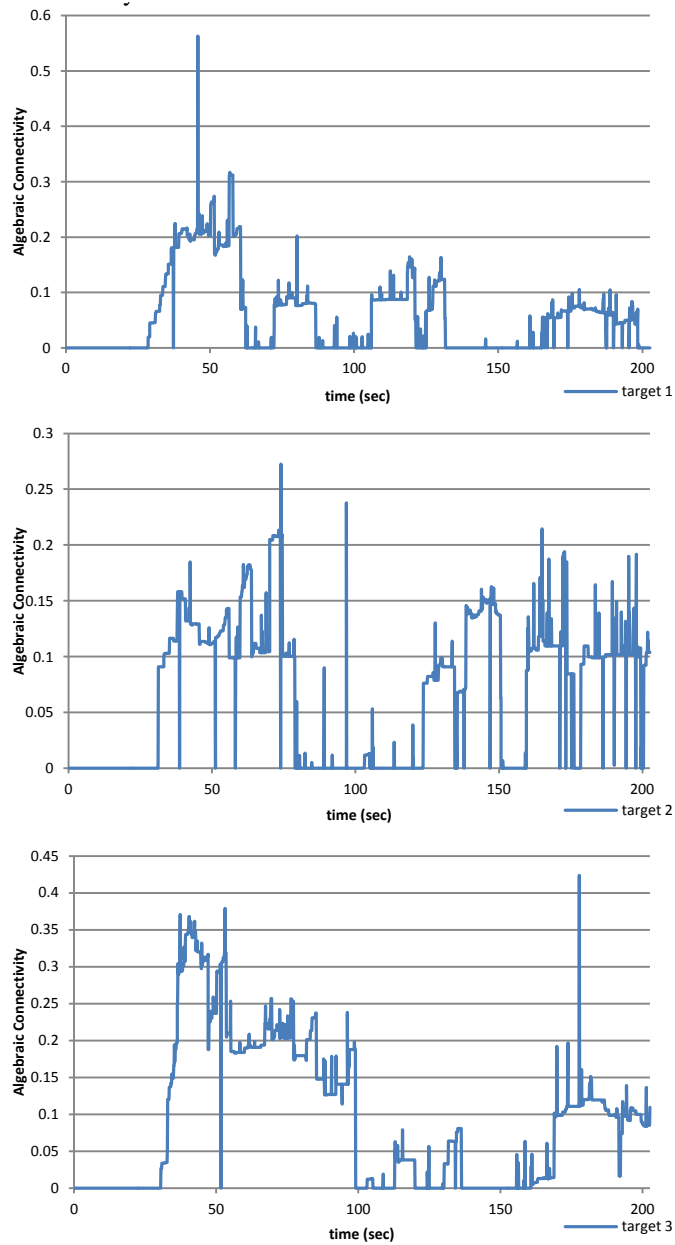


Figure 7.7: Average algebraic connectivity of three networks of sensors tracking three targets in the estimation-control Semi-Flocking and KCF algorithm

The algebraic connectivity for each network shows how effectively the nodes of the network are connected to one another. This value for a network G is greater than 0 if and only if G is a connected graph. Therefore, if just one node of the network is not connected to the rest, the algebraic connectivity of that network would be 0 or less. As Figure 7.7 shows, for each group of sensors flocking around a target, the algebraic connectivity increases at the beginning but then falls several times and again increases. In contrast, in the results reported in [7], the algebraic connectivity always increases and never falls. This behaviour of the estimation-control Semi-Flocking and the KCF algorithm is mostly related to the dynamicity of the group of sensors that are flocking around each target. By dynamicity, I mean changing of the members of the group over time. In the Flocking algorithm, as all the targets join with each other to create a big and connected group of tracking sensors, the algebraic connectivity of the group always increases with time. But in the Semi-Flocking algorithm, sensors are always switching between the searching and tracking modes, based on their distance to each target and the number of trackers involved. This algorithm does not dedicate all the sensors to tracking. On the other hand, the targets are mobile, so while a group of sensors always exists around each target, some new members may join this group or leave it. This behavior of the Semi-Flocking algorithm, which is called dynamicity in the group membership of the semi-flocks, results in alternative increases and decreases in algebraic connectivity.

Although algebraic connectivity of the sensor network around each target alternatively increases and decreases, the results reported in Section 5.3, and also reported for the MSE and information value in this research, confirms that the group is still connected enough to track the target and apply the KCF algorithm successfully. Figure 7.8 shows this fact directly. This figure compares the MSE and algebraic connectivity for Target 1 within the time interval $[0, 200]$ seconds. As this figure shows, while the algebraic connectivity alternatively increases and decreases, the MSE remains low. Thus, the algebraic connectivity is high enough to let the KCF perform well in fusing sensory data. The results for the other two targets are almost similar to the results for Target 1.

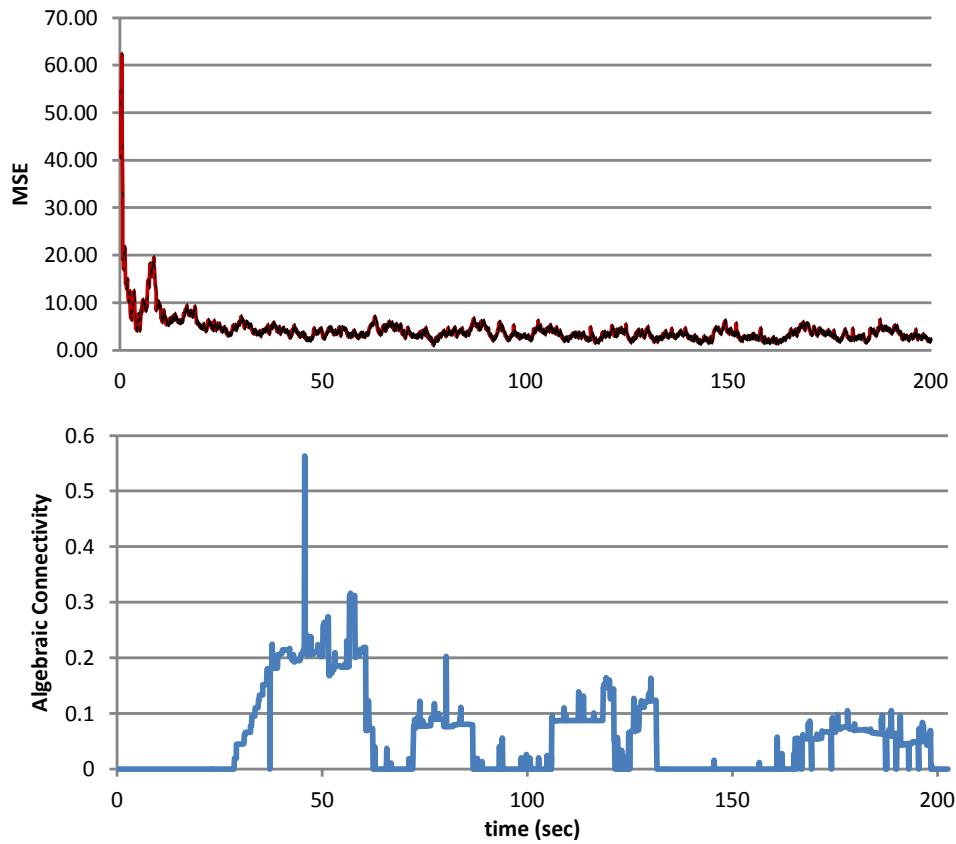


Figure 7.8: Comparison of MSE and algebraic connectivity for Target 1 within the time interval $[0, 200]$

Figure 7.9 shows the target trajectory of three targets and fused estimates of semi-flocks around each target. This figure shows how accurately KCF can estimate the position of the target. The target trajectory and its estimate strongly match for all the targets, thus showing the effectiveness of the proposed estimation-control algorithm.

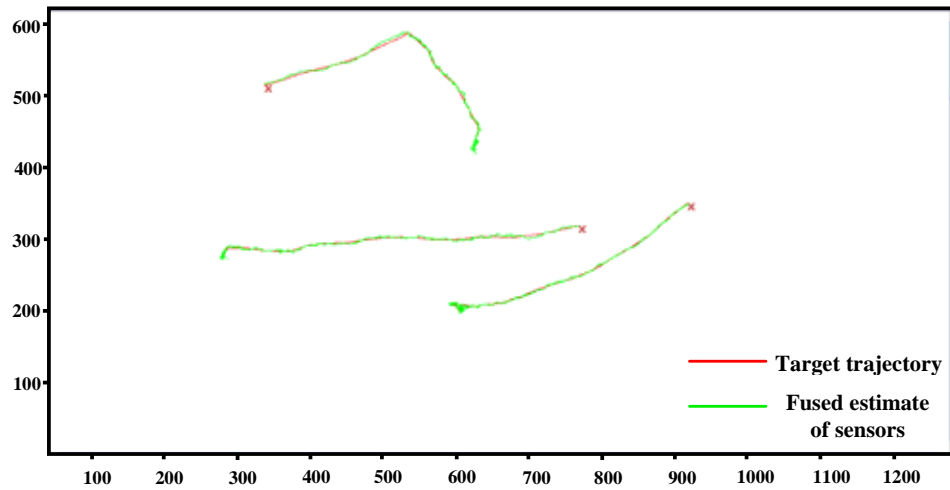


Figure 7.9: Target trajectory and fused estimates of three networks of sensors tracking three targets in estimation-control Semi-Flocking and KCF algorithm

7.5 Summary

This chapter introduced a reliable estimation-control framework for tracking multiple targets in a noisy sensor network. In this framework Semi-Flocking algorithm plays the role of controller, and KCF acts as the estimator. Use of the Semi-Flocking, instead of the simple Flocking algorithm, in this combination allows coordinated tracking of multiple targets instead of engaging all the sensors to cover just one target. In addition, combining KCF algorithm with each group of Semi-Flocking sensors reduces their fused noise and provides a reliable target coverage. The proposed algorithm has been tested with several parameters, including MSE, average information value and average algebraic connectivity. It has been found that the proposed algorithm for tracking multiple targets represents almost similar results to the results reported in [7] for tracking only a single target. These results show that the proposed algorithm can successfully obtain reliable target coverage under a wide range of target dynamics and under noisy conditions in a sensor network.

Chapter 8

Multi-Target Engagement in complex Mobile Surveillance Sensor Networks

Efficient use of the network's resources to collect information about objects (events) in a given volume of interest (VOI) is a key challenge in large-scale sensor networks. Sensor management techniques in such networks aim to maximize performance while minimizing cost. Multi-sensor multi-target tracking in surveillance applications is an example where the network's success in tracking targets, efficiently and effectively, hinges significantly on the network's ability to allocate the right set of sensors to the right set of targets so as to achieve optimal performance. This task can be even more complicated when both sensors and targets are mobile.

To ensure the timely tracking of mobile targets, the surveillance sensor network needs to maintain continuous engagement with all targets in its working area. Thus, the network must be able to perform the following tasks in real-time: 1) target-to-sensor allocation; 2) sensor mobility control and coordination. This chapter presents a combination of the Semi-Flocking algorithm (presented in Chapter 5), as a multi-target motion control and coordination approach, and the hierarchical DCOP modelling algorithm (presented in Chapter 4), as an allocation approach, to tackle the target engagement problem in mobile multi-target multi-sensor surveillance systems. The Semi-Flocking algorithm approaches the problem by assigning a small flock of sensors to each target, while at the same time

leaving some sensors free to explore the environment. On the other hand, hierarchical DCOP modelling is an efficient and fundamental approach that can be elegantly combined with the Semi-Flocking algorithm to achieve a simultaneous control and allocation approach. The experimental results show that this algorithmic combination can successfully engage multiple sensors with multiple mobile targets such that the number of uncovered targets is minimized and the sensors' utilization factor is maximized.

The outline of the chapter is as follows. Section 8.1 provides a brief introduction to the target engagement problem. Section 8.2 describes the proposed sensor engagement formulation. The pseudo code presented to solve the problem is represented in Section 8.3. Section 8.4 presents experimental work conducted to evaluate the proposed formulation. Finally, concluding remarks are given in Section 8.5.

8.1 Introduction

Sensor networks typically employ a large number of sensory nodes that are capable of collecting massively diverse information about their operating environment. This capability has granted them in recent years an important status in the research community and in a wide range of commercial applications. Their importance is expected to increase as sensor networks continue to advance in capabilities. An important challenge in large-scale sensor networks is the efficient use of the network's resources to collect information about a given volume of interest (VOI). To address this challenge, sensor networks employ resource management schemes intended to maximize information gathering at a minimum cost. Often, this goal morphs into a problem of managing a multi-criteria program of conflicting objectives. Achieving such objectives can prove hard, particularly in mission-critical applications such as multi-sensor multi-target surveillance applications where the network's success in tracking targets in a given VOI, efficiently and effectively, hinges significantly on the network's ability to allocate the right set of sensors to the right set of targets so as to achieve optimal performance. This task can be even more complicated if the surveillance application requires the sensors and targets to be mobile.

The surveillance sensor network must maintain engagement with all mobile targets in a

given VOI, to ensure timely target tracking. Thus, the network must perform the following real-time tasks alternately at predefined time intervals: 1) combinatorial target-to-sensor allocation; 2) sensor mobility control and coordination. Given the tasks involved, the sensor-target engagement is obviously an NP-hard problem. Thus, for sensor networks to succeed in such an application, an efficient approach that can tackle this NP-hard problem in real-time is needed.

The Target-to-sensor problem is addressed in Chapter 4. As mentioned in Chapter 4 the target-to-sensor allocation problem is concerned with a selection process that involves assigning targets to a set of sensors. This selection process is a combinatorial optimization problem in which some sensors, targets and constraints are given and the goal is to compute an allocation scheme that minimizes constraint violation costs. This NP-hard problem requires major computational and communication efforts, especially in large and dynamic sensor networks. Optimization constraints can be used to capture sensor restrictions in tracking a target, for example, to signify the inability of the sensor to track more than one target simultaneously, or the requirement of more than one sensor to track a target.

Generally speaking, surveillance applications imply that targets are mobile, and in this case, the problem definition changes over time. Therefore, even after the sensors find a configuration that enables them to track all targets, the expected dynamism of the environment may quickly invalidate this configuration. Thus, the process of finding the best assignment of sensors to targets with minimum cost has to adapt to this dynamism by evolving as the environment changes in time. This fact highlights the need for a dynamic sensing strategy that can enable sensor optimal mobility control and coordination so as to achieve maximum area and target coverage with minimum cost. The strategy should enable the sensors to continuously plan and coordinate their locations in order to avoid collisions and conflicts. This problem is addressed through Chapters 5 to 7.

This chapter proposes a formulation for sensor-to-target engagement in surveillance applications. The formulation captures the issues of sensor-to-target allocation, sensor mobility control, and sensor coordination as intertwined problems. For the sensor-to-target allocation problem, the distributed constraint optimization (DCOP) modelling based approach (introduced in Chapter 4) is employed. Subsequently, the Semi-Flocking algorithm (introduced in Chapters 5), a Flocking-based approach, is employed for control and coor-

dination.

This chapter presents a combination of the Semi-Flocking algorithm, as a multi-target motion control and coordination approach, and a hierarchical DCOP modelling algorithm, as an allocation approach, to tackle the target engagement problem in mobile multi-target multi-sensor surveillance systems. The Semi-Flocking algorithm approaches the problem by assigning a small flock of sensors to each target, while at the same time leaving some sensors free to explore the environment. Flocking-based approaches have recently gained significant attention as a tool for addressing motion control problems. On the other hand, hierarchical DCOP modelling is an efficient and fundamental approach that can be elegantly combined with the Semi-Flocking algorithm to create a simultaneous control and allocation approach. The experimental results show that this algorithmic combination can successfully engage multiple sensors to multiple mobile targets such that the number of uncovered targets is minimized and the sensors' utilization factor is maximized.

8.2 Coupled Allocation and Control Algorithm

Chapter 4 introduced an efficient approach to solve the target-to-sensor allocation problem by modelling the problem as a hierarchical Distributed Constraint Optimization Problem (HDCOP). On the other hand, Chapter 5 introduced Semi-Flocking, a biologically inspired algorithm that benefits from key characteristics of other Flocking-based approaches. The Semi-Flocking algorithm became more powerful through Chapters 6 and 7.

Although both hierarchical DCOP modelling and Semi-Flocking are successful approaches, neither may be able to address target engagement individually in complex problems. The efficiency of each method strongly depends on the complexity of the problem to be solved. This Chapter introduces three complexity levels for the surveillance application:

- **Level 1:** only static sensors (or mobile sensors that are moving randomly and out of any control) are available in the sensor network, and the number of targets is high (static sensors/crowded area)

- **Level 2:** mobile sensors (that are moving under a control strategy) are available, but the number of targets is low (mobile sensors/non-crowded area)
- **Level 3:** mobile sensors (that are moving under a control strategy) are available and the number of targets is high (mobile sensors/crowded area)

Figure 8.1 depicts these three scenarios in a surveillance system that is used for monitoring a small size airport (i.e., Waterloo Regional Airport). As illustrated in this figure, this surveillance system includes some static and mobile sensors in addition to some critical and non-critical mobile targets.

In all the scenarios the goal is to manage the sensors to maximize (critical) target and area coverage. As represented in this figure, in the first level there are a high number of critical targets, thus it is a crowded and dense area. Also there are a number of static (or mobile but non-controllable) sensors. In this scenario, the hierarchical DCOP modelling target-to-sensor allocation approach can be applied to find the optimal assignment between targets and available sensors. Chapter 4 represented how this assignment can be done optimally. As the sensors are static or non-controllable, there is no requirement for a control and coordination strategy. In the second level, there are both mobile (controllable) and static sensors and also both critical and non-critical targets, but the number of critical targets is very low. Thus, this area can be categorized as a non-crowded area. As there are sufficient sensors, the target engagement problem can be solved using an appropriate control and coordination algorithm, and there is no need for a target-to-sensor allocation approach. As represented in Chapter 5, the Semi-Flocking algorithm works for such applications.

Finally, the third level describes the most complex scenario, in which there are both mobile (controllable) and static sensors, so there is an essential need for a control and coordination strategy. On the other hand, there are a high number of critical targets, thus there is a high demand on the limited available sensors and we need to apply an appropriate target-to-sensor allocation approach to find the optimal allocation between the targets and sensors.

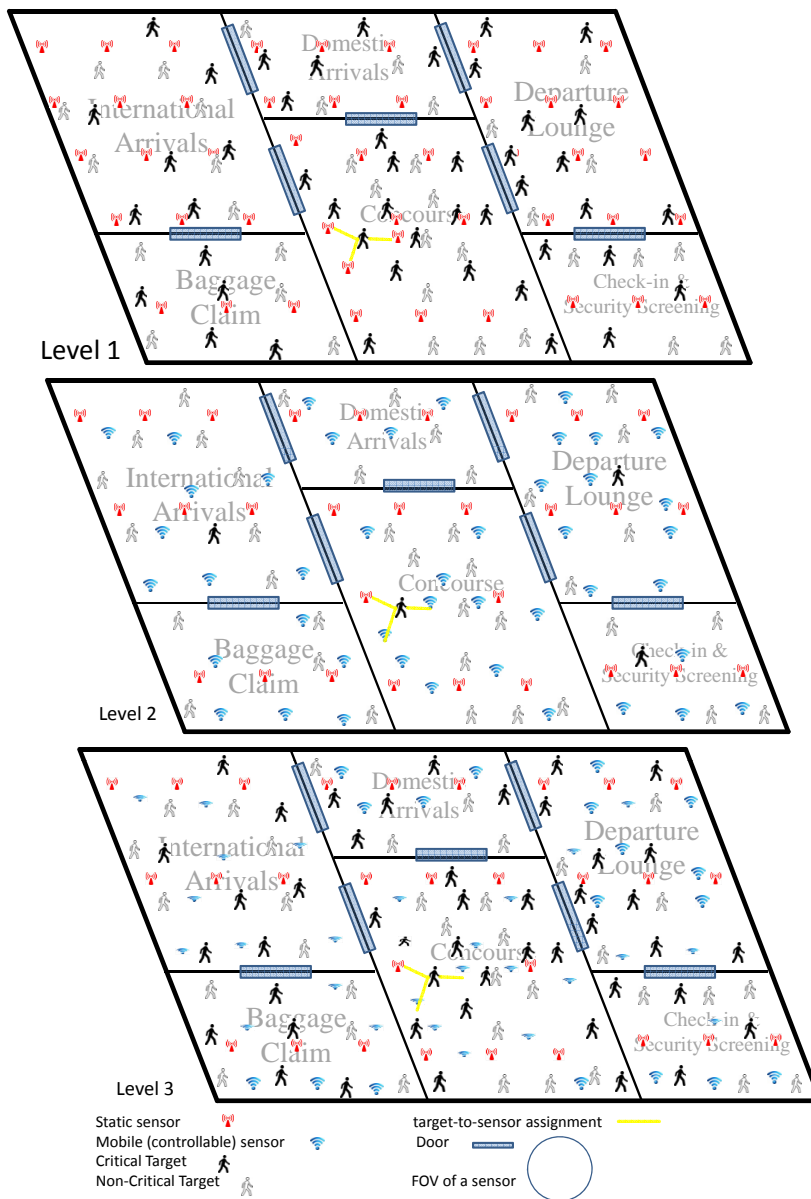


Figure 8.1: Target engagement problem in an airport surveillance application. Level 1) non-controllable sensors/crowded area, Level 2) controllable sensors/non-crowded area, Level 3) controllable sensors/crowded area

Target engagement in such a complex set-up is a twofold problem that should handle

control and coordination of sensors and optimal assignment of sensors to targets actively. In addition as targets are mobile an optimal assignment between sensors and targets may not work over time. Thus, we need to alternately switch between control and allocation strategies. Figure 8.1 represents target engagement as a twofold problem where target engagement actively switches between two sub-problems. This figure also illustrates selected approaches to deal with each of the sub-problems.

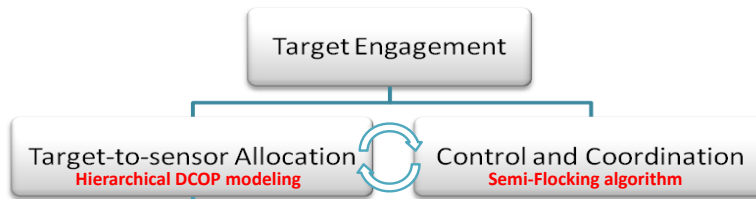


Figure 8.2: Target engagement problem as a twofold problem

To have a successful engagement between sensors and targets in the described complex surveillance system, this chapter proposes a solution strategy in which the following steps should be repeatedly done:

1. Move the sensors toward their optimal positions based on the Semi-Flocking algorithm
2. Find the optimal assignment between sensors and targets based on the hierarchical DCOP modelling technique

Figure 8.3 represents two steps of the proposed approach for solving the target engagement problem in the airport example. The three parts of this figure represent the state of the problem before and after applying each of the mentioned steps. In the first step (Figure 8.3 - b), the Semi-Flocking algorithm is used to navigate sensors to create a small flock around each target. It also leaves some of the sensors free to search the surveillance area for new targets. If one or more targets is close to another, the algorithm creates a larger flock to cover all of these targets.

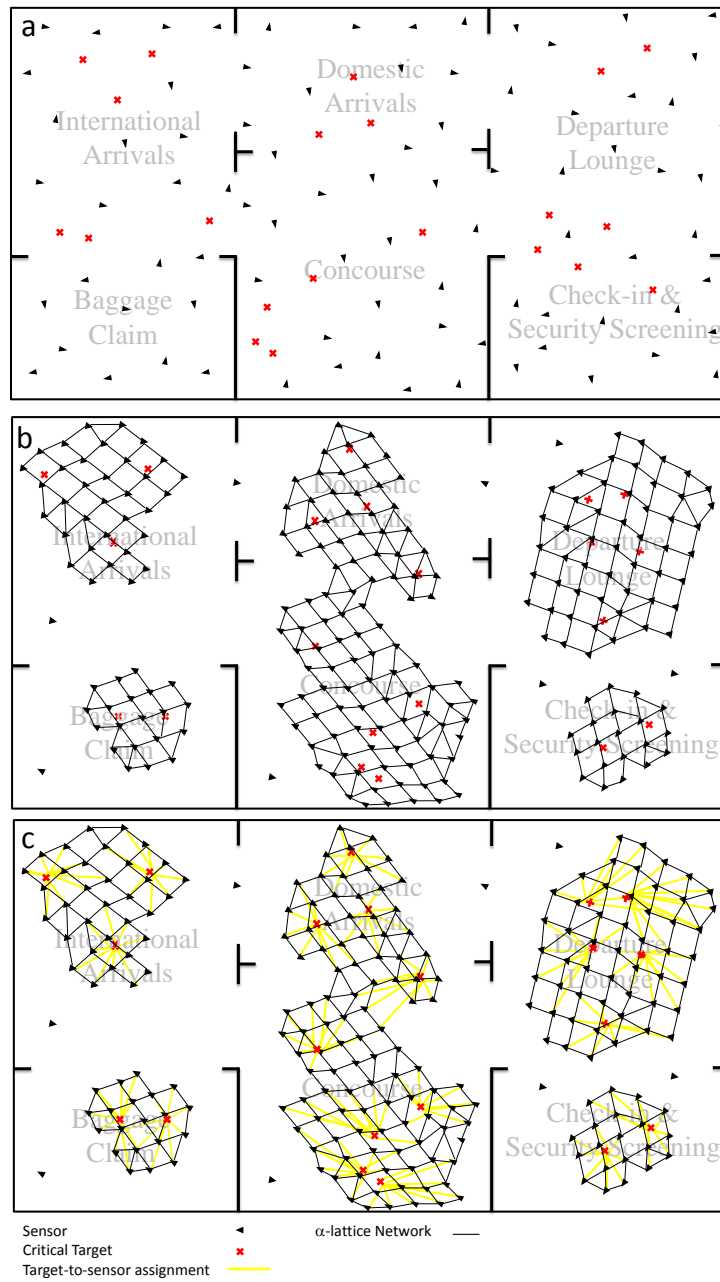


Figure 8.3: Example of target engagement using Semi-Flocking and hierarchical DCOP modeling approaches, a) before applying control and allocation algorithms, b) after applying the Semi-Flocking algorithm, c) after applying the hierarchical DCOP modeling (assignment of 10 sensors to each target)

In the second step (Figure 8.3 - c), the hierarchical DCOP modelling technique is used to find the optimal assignment of 10 sensors to each target based on their current positions. As described in Chapter 4 it first partitions the main problem into sub-problems and next models each part as a separate non-binary variable DCOP. Then uses a DCOP algorithm to find the optimal assignment for covering the maximum number of targets.

After finding a suitable assignment using the allocation technique, we need to switch again to the control strategy because of the dynamicity of the engagement problem, which includes the movement of current targets, entering new targets or leaving current targets. If the targets change their position, then the Semi-Flocking algorithm changes the position of the formed networks of sensors to cover the target in its new position. If a new target arrives, the Semi-Flocking algorithm shrinks the other groups providing some sensors for the new target. Finally, if a target leaves the AOI, then the Semi-Flocking algorithm scatters sensors allocated to that target between other groups in higher demand, or applies them to search the AOI for new targets. Switching between allocation and control strategies continues at predefined time intervals until the end of the monitoring time.

8.3 Pseudo Code for Solving Complex Target Engagement Problem

Figure 8.4 represents the pseudo code presented in this research for solving the described complex target engagement problem. This pseudo code includes two parts. In the first part the Semi-Flocking algorithm is applied to compute u_i for each sensor i , and then the algorithm changes the speed and position of each sensor based on the calculated value. The next step of this pseudo code describes the hierarchical DCOP modelling which models the problem based on the new position of the sensors and targets to a hierarchy of DCOPs and then solves the modelled problem using a DCOP solver such as the distributed breakout algorithm (DBA) [141]. The DBA algorithm is selected in this research because it uses an iterative improvement strategy that is more suitable for real-time tasks such as the problem of this research. This fact has been shown in Chapter 4.

```

While (monitoring time is finished)
(1) for sensor  $i = 0 \rightarrow n$  do
(2)    $u_i^y = 0$ ;
(3)   for target  $tj = 0 \rightarrow m$  do
(4)     if  $\|q_{tj} - q_i\|_\sigma \leq \theta_j$  then
(5)        $u_{i,tj}^y = \frac{c_{1j}(q_{tj}-q_i)+c_{2j}(p_{tj}-p_i)}{n_{tj}}$ 
(6)       //where  $n_{tj}$  is the number of sensors already tracking tj
(7)        $u_i^y = u_i^y + u_{i,t}^y$ ;
(8)     end if
(9)   end for
(10)  if  $u_i^y == 0$  then // the sensor is in searching mode
(11)     $q_{w,l}$  = center of adjacent areas that has least visited times
(12)     $u_i^y = c \times (q_{w,l} - q_i)$ 
(13)    //toward the area that has longest time not being visited
(14)  end if
(15)   $u_i = u_i^\alpha + u_i^y$ 
(16)   $p_i = p_i + u_i \times \Delta t$  // $\Delta t$  is the updating time intervals
(17)   $q_i = q_i + p_i \times \Delta t$  // $\Delta t$  is the updating time intervals
(18) end for

```

Control and coordination by semi-flocking algorithm

Figure 8.4: Pseudo code for solving complex target engagement problem

Target-to-sensor allocation by hierarchical DCOP modeling

```

(19) Start of modeling
(20)  $L \leftarrow 1$ 
(21) while  $|P_L| > X$ 
(22)   partition  $P_L$  into a set of sub-problems:  $\{P_{L,1}, P_{L,1}, \dots, P_{L,n}\}$ 
      such that
(23)   a.  $\forall i: 1 \rightarrow n: |P_{L,i}| \leq X$ 
(24)   b.  $\forall i, j: i, j: 1 \rightarrow n: \min |inter - constraint(P_{L,i}, P_{L,j})|$ 
(25)   c.  $\forall i: 1 \rightarrow n: \max |intra - constraint(P_{L,i})|$ 
(26)   d.  $\forall i, j: i, j: 1 \rightarrow n: P_{L,i} \cap P_{L,j} = \emptyset$ 
(27)   e.  $Sensor_L = \bigcup_{\forall i} Sensor_{L,i}$ 
(28)   f.  $Target_L = \bigcup_{\forall i} Target_{L,i}$ 
(29)   g.  $Constraint(P_L) = (\bigcup_{\forall i} intra - constraint(P_{L,i}) \cup$ 
       $(\bigcup_{\forall i, j} inter - constraint(P_{L,i}, P_{L,j})))$ 
(30)   for all  $P_{L,i} i: 1 \rightarrow n$  do
(31)     model  $P_{L,i}$  as a DCOP as described in [1]:
(32)     a.  $Sensor_{L,i} (/Target_{L,i}) \rightarrow$  variables
(33)     b.  $Target_{L,i} (/Sensor_{L,i})$  in the FOV of each sensor ( $/$ 
      target)  $\rightarrow$  Domain
(34)     c.  $Intra - constraint(P_{L,i}) \rightarrow$  constraints
(35)   end for
(36)    $L \leftarrow L + 1$ 
(37)   //all targets that are within FOV of sensors of more than one
      region participate in the set of targets of next layer
(38)   for all  $t_i \in Target_{L-1}$  do
(39)     if  $\exists P_{L-1,i}, P_{L-1,j} (FOV(t_i) \cap P_{L-1,i} \neq \emptyset) \wedge (FOV(t_i) \cap$ 
       $P_{L-1,j} \neq \emptyset)$  then
(40)       add  $t_i$  to  $Target_L$ 
(41)     end if
(42)   end for
(43)   //all the sensors that their FOV covers more than one region

```

Figure 8.4: Pseudo code for solving complex target engagement problem (continue)

```

(44) for all  $s_i \in \text{Sensor}_{L-1}$  do
(45)   if  $\exists P_{L-1,i}, P_{L-1,j} \mid (FOV(s_i) \cap P_{L-1,i} \neq \emptyset) \wedge (FOV(s_i) \cap$ 
       $P_{L-1,i} \neq \emptyset)$  then
(46)     add  $s_i$  to  $\text{Sensor}_L$ 
(47)   end if
(48) end for
(49) //inter region constraint of previous level constructs constraints of
      this level
(50)  $\text{constraint}(P_L) = \text{inter-constraint}(P_{L-1})$ 
(51) end while
(52) model the last level of hierarchy
(53) model  $P_L$  as a DCOP as described in [1]:
(54)   a.  $\text{Sensor}_L$  (/Target $_L$ )  $\rightarrow$  variables
(55)   b. Target $_L$  (/Sensor $_L$ ) in the FOV of each sensor (/
      target)  $\rightarrow$  Domain
(56)   c.  $\text{constraint}(P_L) \rightarrow$  constraints
(57) End of modeling
(58) Solve the modeled problem using DBA [28] algorithm
(59) End while

```

allocation by hierarchical DCOP modeling

Figure 8.4: Pseudo code for solving complex target engagement problem (continue)

As represented in Figure 8.4, switching between Semi-Flocking and DCOP modelling continues until the end of the monitoring time. This figure uses the notations listed below:

- L : a variable that shows the level of the hierarchy.
- X : the size of the largest manageable problem.
- P_L : the problem in Level L .
- $P_{L,i}$: the sub-problem i in Level L .
- $\text{Target}_{L,i}$: targets in sub-problem i in Level L .
- $\text{Sensor}_{L,i}$: sensors in sub-problem i in Level L .

- constraint (P_L): total constraints in level L .
- inter-constraint ($P_{L,i}, P_{L,j}$): inter region constraints between sub-problems $P_{L,i}$ and $P_{L,j}$ in level L .
- intra-constraint ($P_{L,i}$): intra region constraints of $P_{L,i}$ in level L .

8.4 Experiments and Discussion

8.4.1 Experimental Setup

This section describes the experimental setup of the proposed coupled algorithm (Semi-Flocking + hierarchical DCOP modelling) for tracking targets in a multi-target environment similar to the small airport described in Section 8.2. In this airport, the area of interest (AOI) is a 1250×665 rectangle and all the targets and sensors remain in this area all the time $t \geq 0$.

Both parts of the coupled target engagement algorithm, Semi-Flocking and hierarchical DCOP modelling (including DBA algorithm), are implemented in Java. These two algorithmic components are executed in predefined time intervals. The parameters of Semi-Flocking algorithm are set as follows: $d = 140$, $r = 1.2d$, $\epsilon = 0.1(\text{for}\sigma - \text{norm})$, $a = 8b$, $b = 1$ for $\phi(z)$, $h = 0.2$ for the bump function of $\phi_\alpha(z)$. The hierarchical DCOP modelling and the DBA algorithm try to find the optimal assignment of three sensors to each target.

In this experiment the number of targets varies from 15 to 25 and the number of sensors are selected proportional to the number of targets (number of sensors $\cong 6.5 \times$ number of targets). The greater the number of targets, the greater the number of sensors deployed in the environment. All the sensors are mobile, and their initial positions are selected randomly.

The result reported for each point is the average over 10 randomly generated instances. All targets are mobile. In all instances, the initial position and lifetime of each target were

selected randomly from a uniform distribution. For each instance, the monitoring time is continued for 300 sec (5 minutes).

The dynamics of targets when they are within the AOI (and are not on the borders) is represented in Equation 8.1.

$$x(k+1) = Ax(k) + Bw(k) \quad (8.1)$$

with with

$$A = \begin{bmatrix} 1 & 0 & \epsilon & 0 \\ 0 & 1 & 0 & \epsilon \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{\epsilon^2}{2} & 0 \\ 0 & \frac{\epsilon^2}{2} \\ \epsilon & 0 \\ 0 & \epsilon \end{bmatrix}$$

where $x(k) = (q_1(k), q_2(k), p_1(k), p_2(k))^T$ denotes the state of a target at time k , and $\epsilon = 0.02$ seconds is the step-size. To keep a target within the AOI, the target reverses their previous direction at the borders.

8.4.2 Evaluation Parameters

Three parameters were selected to evaluate the proposed coupled algorithm in this experiment:

- **The number of conflicts:** represents the number of constraint violations (conflicts) occurring in the optimal assignment between sensors and targets at each moment of time. A constraint is violated if one of the following conditions happens:
 - The number of sensors assigned to a target is less than the required value (in this research the value of three).

- A sensor is assigned to more than one target.

Actually this latter parameter shows how many targets are not perfectly covered by the allocation algorithm.

- **Average of utilization factor:** the utilization factor for each sensor is defined as the ratio of the time that the sensor has been allocated to a target to the total time that has passed since its appearance. Equation 8.2 represents how this fraction is calculated for sensor i . The average of the utilization factor represents the average of the utilization factor over all of the sensors at a specified time.

$$UtilizationFactor(S_i) = \frac{\text{time that } S_i \text{ is allocated to a target}}{\text{current time} - \text{time of the appearance of } S_i \text{ in the FOV}} \quad (8.2)$$

- **Variance of utilization factor:** represents the variance of the utilization factor over all of the sensors at a specified time.

8.4.3 Simulation Results and Analysis

Figure 8.5 illustrates a snapshot of the engagement of 98 sensors to 15 targets, using the coupled target engagement algorithm proposed in this chapter. In this example, each target is considered to be covered if it is assigned to three sensors at the same time. Yellow lines represent the assignment of a sensor to a target. In this example, the surveillance area is a dense one and both sensors and targets are mobile. Thus, the coupled algorithm is applied to solve the problem. As this figure demonstrates, small flocks are formed and maintained around the targets by the control part (Semi-Flocking algorithm). If two or more targets are close to each other, then a bigger flock is created to cover all of them. At the same time, some free sensors explore the AOI. The allocation part (hierarchical DCOP modelling) is assigned three sensors for each target. This assignment is such that the maximum number of targets is covered by the available sensors around them. Note that only sensors within a predefined distance from each target can be so assigned. The observations are in close

agreement with our expectations of the behaviour of this coupled algorithm, depicted in Figure 8.3.

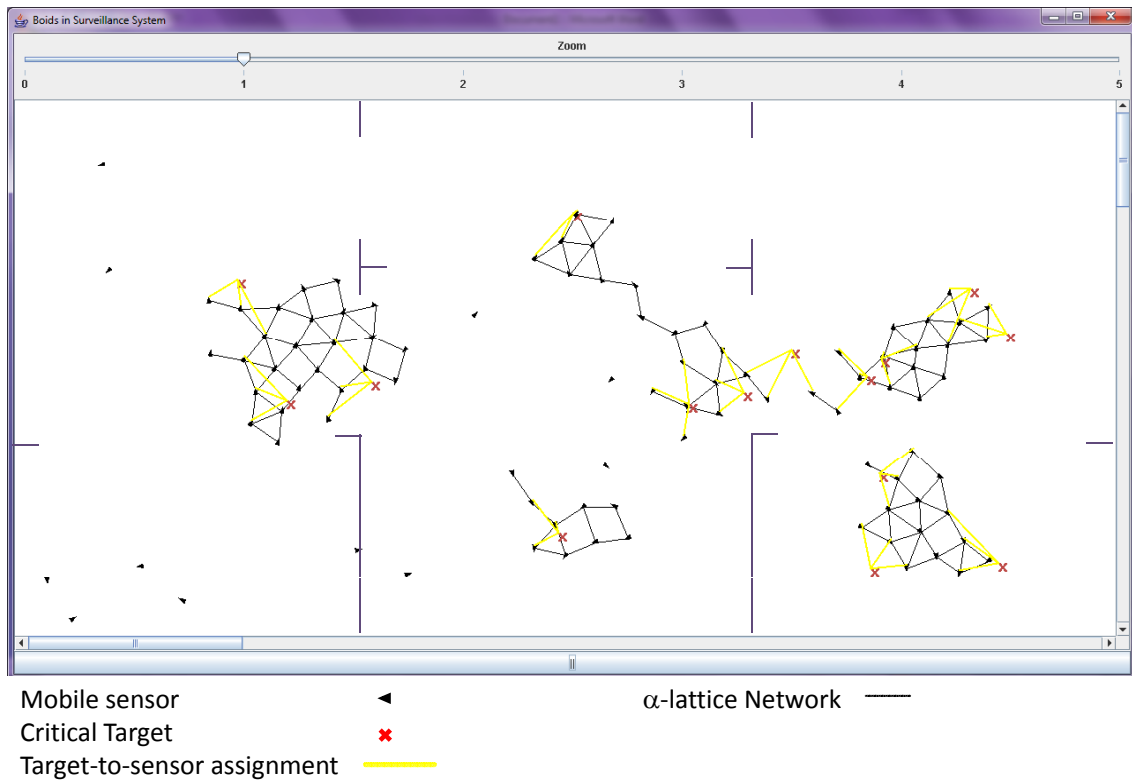


Figure 8.5: Snapshot of executing coupled Semi-Flocking and hierarchical DCOP modelling approach by 15 targets and 98 sensors

The results of evaluating the proposed algorithm are as follows:

1. **The number of conflicts:** Figure 8.6 represents the number of constraint violations at each moment of time after running the coupled Semi-Flocking and hierarchical DCOP modelling algorithm. This figure demonstrates the results for the problems with 15 to 25 targets. The results show that for all numbers of targets, the optimal assignment found at the beginning of the algorithm contains a high number of conflicts, but as time goes on, the number of conflicts reduces to less than one conflict.

This reduction is the result of applying the Semi-Flocking algorithm to direct sensors to better positions in which they can be assigned to more targets. In other words, at the beginning, as the sensors are deployed randomly, there are not enough sensors around each target to be assigned to that target by the allocation algorithm; thus, in first efforts, even a perfect and optimal allocation algorithm cannot find a suitable assignment between sensors and targets due to the lack of sensors in the required positions. Over time, the sensors change their positions based on the Semi-Flocking algorithm to the places with higher demand. As a result, a suitable allocation algorithm, such as the one selected in this chapter, is able to assign sensors to targets, in a way that avoids most of conflicts. The reduction of the conflicts continues to the end of the monitoring time (300 Sec).

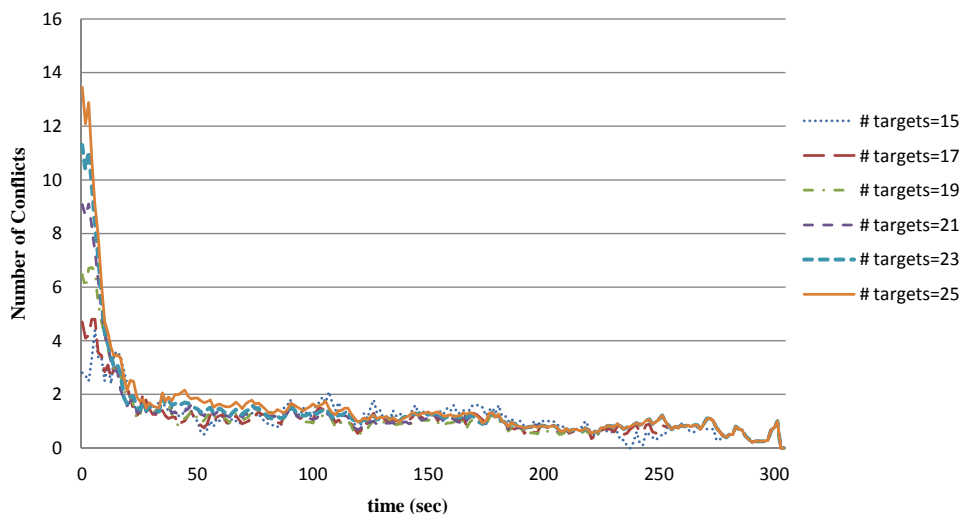


Figure 8.6: Number of violated constraints (conflicts) in the coupled Semi-Flocking and hierarchical DCOP modelling algorithm

2. **Average of utilization factor:** Figure 8.7 represents the average of the utilization factor over all the sensors at each moment of time for experiments with 15 to 25 targets. As described in Section 8.4.2, the utilization factor for a sensor represents the ratio of the time that the sensors have been allocated to a target, to the total time passed since its appearance. At the beginning of the algorithm, none of the sensors

are allocated to any of the targets so their utilization factors are zero. Over time, the number of allocations (due to the changing sensor positions) and the amount of time that they are allocated, increases, and as a result, the average of utilization factor increases. The highest average for the utilization factor was detected around 50 seconds after starting the algorithm, but this value does not remain high due to the mobility of the targets, which results in losing some of the assignments and establishing new ones. The utilization factor average remains between the ranges of 0.65 to 0.8 from the 150th sec to the end of monitoring time for almost all the experiments (various numbers of targets).

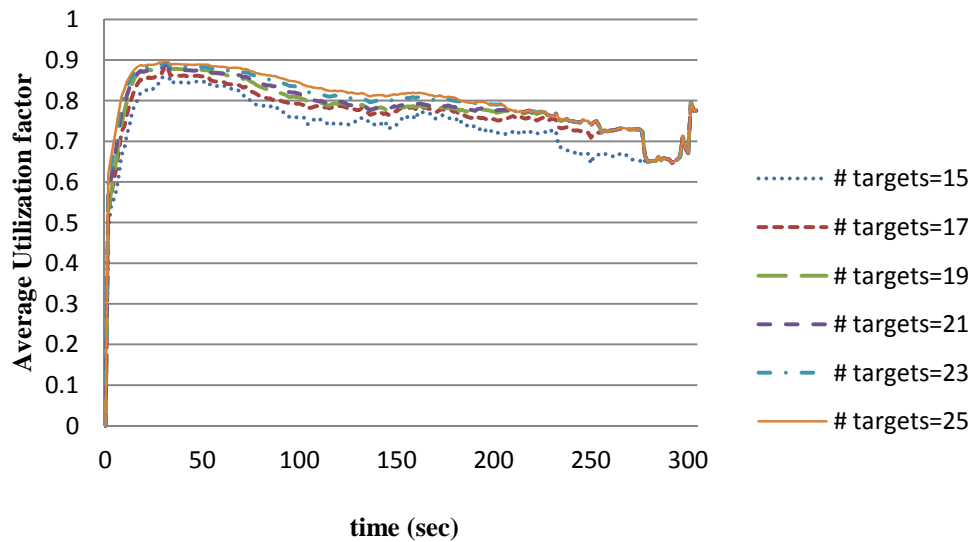


Figure 8.7: Average of the utilization factor over all of the sensors in the coupled Semi-Flocking and hierarchical DCOP modelling algorithm

- Variance of utilization factor:** Figure 8.8 represents the variance of the utilization factor over all the sensors at each moment of time for experiments with 15 to 25 targets. At the beginning, the variance is high as only a few sensors are allocated to a target and the rest are unallocated. As time goes on, a higher number of sensors will be allocated to the targets due to changing of the sensors' positions. As a result, the variance will be decreased. This reduction continues to the end of the monitoring

time. A low value of the utilization factor variance in addition to the high value of the utilization factor average represents the fact that numerous targets are allocated to the sensors.

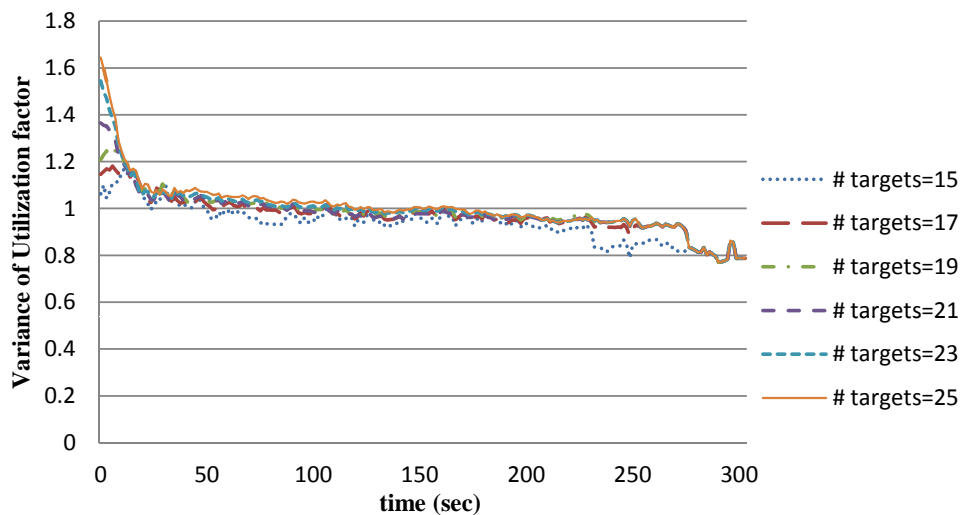


Figure 8.8: Figure 11. Variance of the utilization factor over all of the sensors in the coupled Semi-Flocking and hierarchical DCOP modelling algorithm

8.5 Summary

This chapter introduced a reliable coupled allocation and control algorithm for multiple target engagement, especially in dense surveillance areas in which controllable sensors are available. In this coupled algorithm, the Semi-Flocking algorithm plays the role of controller, and the hierarchical DCOP modelling approach (in addition to the DBA algorithm) acts as the allocator. It is shown that although each of these algorithms are successful in less-complicated surveillance systems, neither is able to work separately in complex scenarios. The presented pseudo code of the coupled algorithm shows how these two algorithms match with each other and can be applied at predefined time intervals to solve the engagement problem. The proposed algorithm is evaluated over several parameters including the number of constraint violations which shows the number of non-perfectly covered targets,

and the average and variance of the utilization factor. It has been found that the proposed coupled algorithm engages multiple sensors to multiple mobile targets such that the number of uncovered targets decreases efficiency and the sensors' mean utilization factor is increased. These results show that the proposed algorithm has successfully engaged the targets and sensors in dense surveillance systems with controllable sensors and mobile targets.

Chapter 9

Conclusion and Future Work

This chapter provides a brief summary of the contributions of this thesis and present some suggestions for future work. Section 9.1 concludes this thesis, highlighting the main contributions. Section 9.2 presents some recommendations for future direction of this research.

9.1 Conclusion

Sensor networks have demonstrated noticeable success in various applications including but not limited to mobile surveillance. Intelligent management of large number of sensor nodes and process the data that these sensors collect is a key challenge in such Large-scale sensor networks. This thesis describes the development of an intelligent sensor-target engagement framework for use in large-scale surveillance applications. Five primary challenges have been addressed in this work:

- **Simultaneous reliable dynamic area and target coverage:** Providing simultaneous coverage over all the targets in the VOI and also acceptable coverage over the Area of Interest (AOI) to detect new targets are two inherently conflicting objectives that are the main requirements in many dynamic surveillance applications.

- **Target-to-sensor allocation:** This selection process, which involves optimal assignment of targets to a set of sensors, is an NP-hard problem. Thus, for sensor networks to succeed in such applications, an efficient approach that can tackle this NP-hard problem in real-time is desperately needed.
- **Sensor mobility control and coordination:** Optimal movement of mobile sensors in such a way as to achieve a certain objective is one of the essential requirements of any sensor-management framework that has mobile and controllable sensors. An appropriate mobility control mechanism will greatly enhance the quality of the final solution.
- **Tracking both manoeuvring and non-manoevring targets:** The need to track manoeuvring targets in addition to non-manoevring ones has been recognized in the majority of the real-world tracking scenarios. Human motion tracking or military/civilian surveillance is an example of these applications in which the targets often move with high manoeuvring ability, and require more-advanced tracking approaches.
- **Reliable Target Tracking:** An important concern in tracking multiple targets in sensor networks is the ability of sensors to track targets in noisy measurement conditions, which is the case in many real-world scenarios. The success of a network to track targets in such environments hinges significantly on the sensors' ability to reach a consensus value on their measurement of targets' status. This value must have the minimum error possible.

To address these challenges, the research work in this thesis offers solutions in seven aspects, as discussed next. Figure 9.1 provides an overview of the steps that have been taken in this research work to present a solution for the target engagement problem.

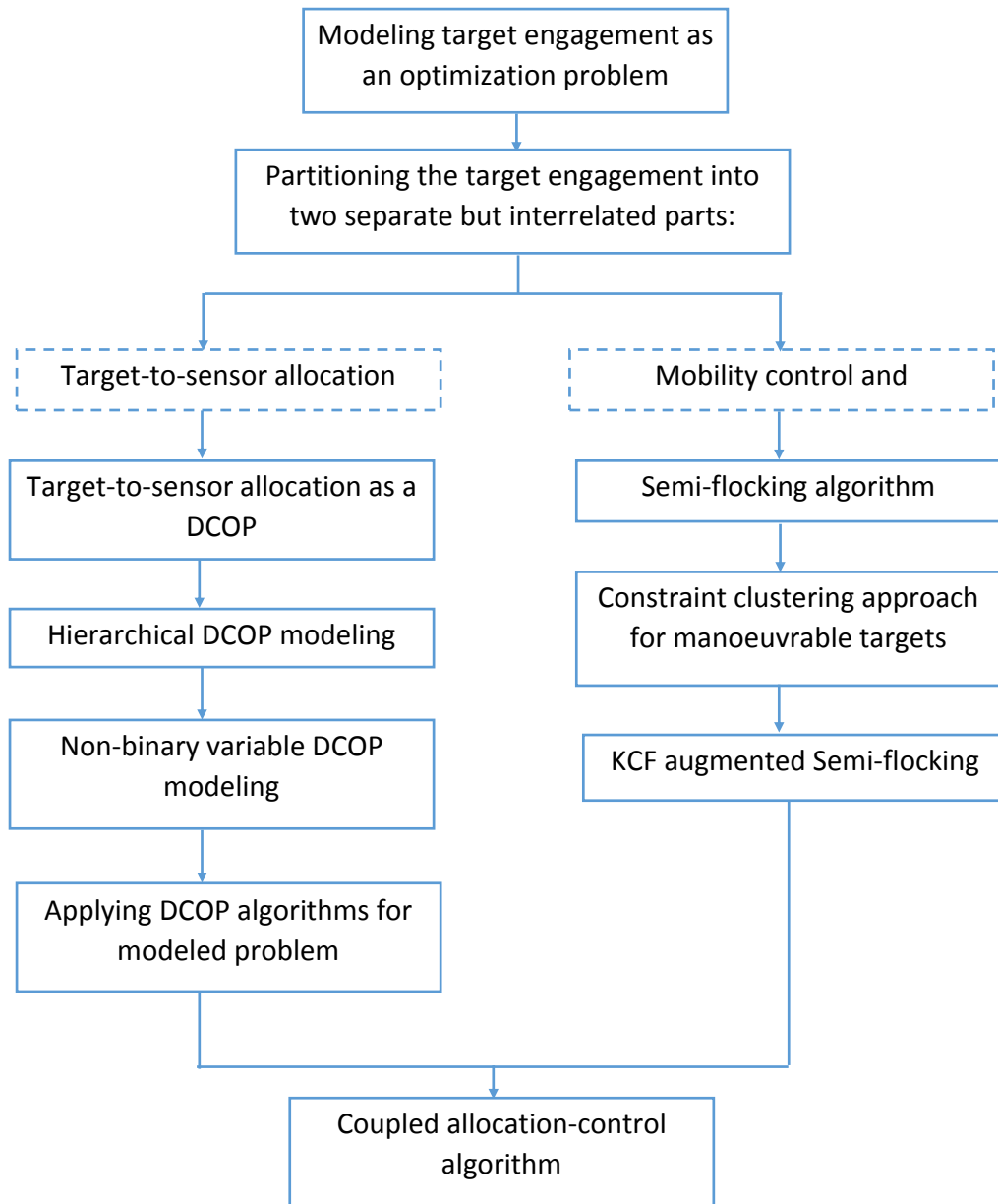


Figure 9.1: An Overview of the proposed target-engagement solution steps.

- **Formulation of the target engagement problem:** This work presents a formal formulation of the target engagement problem as an optimization problem subject to some constraints. The distributed nature of the problem and also the mobility of both sensors and targets are considered in this formulation.
- **Hierarchical DCOP modelling:** This work proposes a technique to solve the target-to-sensor allocation problem by modelling the problem as a hierarchical Distributed Constraint Optimization Problem (DCOP). DCOPs tend to be computationally expensive and often intractable, particularly in large problem spaces such as Wireless Sensor Networks (WSNs). To address this challenge, I propose changing the sensor-to-target allocation to a hierarchical set of smaller DCOPs with a shared system of constraints, avoiding significant computational and communication costs.
- **Non-binary variable DCOP modelling:** In contrast to other DCOP modelling methods, this research presents a non-binary variable model for reducing the number of variables and the number of intra-agent constraints, and consequently reducing the communication cost.
- **Semi-Flocking algorithm:** This work presents Semi-Flocking, a biologically inspired algorithm that benefits from key characteristics of reported Flocking-based algorithms. The Semi-Flocking algorithm approaches the control and coordination problem in sensor networks by assigning a small flock of sensors to each target, while at the same time leaving some sensors free to explore the environment. This approach allows the algorithm to strike a balance between reliable area coverage and target coverage. This balance is facilitated via flock-sensor coordination.
- **Constrained clustering for tracking manoeuvrable targets:** This research presents a constrained clustering approach to be combined with Flocking-based algorithms to provide better coverage over manoeuvrable targets. To perform the constrained clustering, a novel extension of K-means algorithm is presented and applied to cluster the sensors. This extension clusters the sensors based on certain background knowledge. Then the information about the clusters is used to improve coverage over manoeuvrable targets.

- **KCF augmented Semi-Flocking algorithm:** Addressing the problem of reliable multiple target tracking using a sensor network requires a coupled distributed estimation and motion-control approach. This work proposes a framework wherein Semi-Flocking algorithm is employed for the purpose of multi-target motion control, and a Kalman-Consensus Filter (KCF) for the purpose of motion estimation. In the proposed coupled approach, each small group of Flocking sensors (semi-flock) applies a separate KCF algorithm to estimate the position of its target. Doing so allows sensors to collectively provide reliable target engagement and comprehensive area coverage.
- **Coupled allocation-control algorithm:** To ensure timely tracking of mobile targets, the surveillance sensor network needs to maintain continuous engagement with all targets. Thus, the network must be able to perform the following real-time tasks: 1) target-to-sensor allocation; 2) sensor mobility control and coordination. This work presents a combination of the Semi-Flocking algorithm, as a multi-target motion control and coordination approach, and a hierarchical DCOP modelling algorithm, as an allocation approach, to tackle target-engagement problems in mobile multi-target multi-sensor surveillance systems.

9.2 Future Work

Intelligent target engagement is an active area of research. New issues arise in this field as sensor networks develop and new applications and technologies emerge. There are a number of issues that should be investigated in the future:

- **Prediction:** By increasing the number of targets, the chance of missing already detected targets increases in any engagement approach including the one presented in this research. Applying prediction techniques for predicting the next positions of targets and then guide sensors toward such positions has a great impact on decreasing the chance of missing already detected targets.

- **Smart targets:** A smart target is a target that can react in a manner that makes future surveillance more difficult. As smart targets are able to respond to sensing activities, their detecting and tracking in surveillance application is very challenging. The sensors must trade among several modalities to most quickly and effectively detect and track the targets. In this research the problem of tracking linear and manoeuvrable target are addressed. Tracking smart targets is a challenging topic that can be considered as the future direction of this research.
- **Optimal Number of Sensors:** Finding the optimal number of sensors for the specific number of targets to be deployed in the surveillance area is a fundamental important research area in target engagement. Theoretical and experimental analyses are required to find this optimal number. Considering the high cost of sensors, finding the optimal number of required sensors is an important parameter especially in industrial projects as it has a great impact in minimizing the project costs.
- **Security:** Due to the distributed nature of wireless sensor network and their deployment in remote areas, they always incur various types of security threats, especially when they are used in military or surveillance applications. Due to the small size of the sensors and consequently their limitations in processing power and memory, sensor networks are not able to apply traditional security mechanisms with high computation and communication overheads. Therefore, a target engagement framework must take into consideration the required security measures and deal with various types of security threats.

References

- [1] P. J. Modi, H. Jung, M. Tambe, W.-M. Shen, and S. Kulkarni, “A dynamic distributed constraint satisfaction approach to resource allocation,” in *Principles and Practice of Constraint Programming, CP 2001*, pp. 685–700, Springer, 2001.
- [2] C. Zhu, L. Shu, T. Hara, L. Wang, S. Nishio, and L. T. Yang, “A survey on communication and data management issues in mobile sensor networks,” *Wireless Communications and Mobile Computing*, vol. 14, no. 1, pp. 19–36, 2014.
- [3] Y. Cai, W. Lou, M. Li, and M. Li, “Target-oriented scheduling in directional sensor networks,” in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 1550–1558, IEEE, 2007.
- [4] P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo, “Adopt: asynchronous distributed constraint optimization with quality guarantees,” *Artificial Intelligence*, vol. 161, no. 1-2, pp. 149 – 180, 2005.
- [5] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: Algorithms and theory,” *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401–420, 2006.
- [6] W. Yeoh, A. Felner, and S. Koenig, “Bnb-adopt: An asynchronous branch-and-bound dcop algorithm,” in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, vol. 2, pp. 591–598, International Foundation for Autonomous Agents and Multiagent Systems, 2008.

- [7] R. Olfati-Saber and P. Jalalkamali, “Collaborative target tracking using distributed kalman filtering on mobile sensor networks,” in *American Control Conference (ACC), 2011*, pp. 1100–1105, IEEE, 2011.
- [8] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [9] G. J. Pottie and W. J. Kaiser, “Wireless integrated network sensors,” *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.
- [10] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, “Habitat monitoring with sensor networks,” *Communications of the ACM*, vol. 47, no. 6, pp. 34–40, 2004.
- [11] V. C. Gungor and G. P. Hancke, “Industrial wireless sensor networks: Challenges, design principles, and technical approaches,” *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 10, pp. 4258–4265, 2009.
- [12] C.-Y. Chong and S. P. Kumar, “Sensor networks: evolution, opportunities, and challenges,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.
- [13] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *Communications magazine, IEEE*, vol. 40, no. 8, pp. 102–114, 2002.
- [14] Y.-C. Tseng, Y.-C. Wang, K.-Y. Cheng, and Y.-Y. Hsieh, “imouse: an integrated mobile surveillance and wireless sensor system,” *Computer*, vol. 40, no. 6, pp. 60–66, 2007.
- [15] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, “Energy-efficient surveillance system using wireless sensor networks,” in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pp. 270–283, ACM, 2004.
- [16] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, “Senseye: a multi-tier camera sensor network,” in *Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 229–238, ACM, 2005.

- [17] F. Dressler, *Self-organization in sensor and actor networks*. John Wiley & Sons, 2008.
- [18] M. Vinyals, J. A. Rodriguez-Aguilar, and J. Cerquides, “A survey on sensor networks from a multiagent perspective,” *The Computer Journal*, p. bxq018, 2010.
- [19] L. Hodge and M. Kamel, “An agent-based approach to multisensor coordination,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 33, no. 5, pp. 648–661, 2003.
- [20] M. A. Batalin and G. S. Sukhatme, “Spreading out: A local approach to multi-robot coverage,” in *Distributed autonomous robotic systems 5*, pp. 373–382, Springer, 2002.
- [21] A. Howard, M. J. Matarić, and G. S. Sukhatme, “Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem,” in *Distributed autonomous robotic systems 5*, pp. 299–308, Springer, 2002.
- [22] S. Li, C. Xu, W. Pan, and Y. Pan, “Sensor deployment optimization for detecting maneuvering targets,” in *Information Fusion, 2005 8th International Conference on*, vol. 2, pp. 2–7, IEEE, 2005.
- [23] S. K. Singh, M. Singh, and D. Singh, “Routing protocols in wireless sensor networks—a survey,” *International Journal of Computer Science & Engineering Survey (IJCSES) Vol*, vol. 1, pp. 63–83, 2010.
- [24] M. A. Adnan, M. A. Razzaque, I. Ahmed, and I. F. Isnin, “Bio-mimic optimization strategies in wireless sensor networks: A survey,” *Sensors*, vol. 14, no. 1, pp. 299–345, 2013.
- [25] I. Amundson and X. D. Koutsoukos, “A survey on localization for mobile wireless sensor networks,” in *Mobile Entity Localization and Tracking in GPS-less Environments*, pp. 235–254, Springer, 2009.
- [26] N. Xiong and P. Svensson, “Multi-sensor management for information fusion: issues and approaches,” *Information fusion*, vol. 3, no. 2, pp. 163–186, 2002.

- [27] P. Modi, P. Scerri, W. Shen, and M. Tambe, “Distributed resource allocation: A distributed constraint reasoning approach,” *Distributed Sensor Networks: A Multiagent Perspective*, Kluwer Academic Publishers, 2003.
- [28] T. Ibaraki and N. Katoh, “Resource allocation problems: algorithmic approaches,” 1988.
- [29] A. R. Benaskeur and R. Defence, *Holonic approach for control and coordination of distributed sensors*. Defence R&D Canada-Valcartier, 2008.
- [30] Q. Ling, Y. Fu, and Z. Tian, “Localized sensor management for multi-target tracking in wireless sensor networks,” *Information Fusion*, vol. 12, no. 3, pp. 194–201, 2011.
- [31] Z. Tang and Ü. Özgüner, “Cooperative sensor deployment for multi-target monitoring,” *International Journal of Robust and Nonlinear Control*, vol. 18, no. 2, pp. 196–217, 2008.
- [32] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, “Robomote: enabling mobility in sensor networks,” in *Proceedings of the 4th international symposium on Information processing in sensor networks*, p. 55, IEEE Press, 2005.
- [33] E. Natalizio and V. Loscrí, “Controlled mobility in mobile sensor networks: advantages, issues and challenges,” *Telecommunication Systems*, vol. 52, no. 4, pp. 2411–2418, 2013.
- [34] Z. Butler and D. Rus, “Event-based motion control for mobile-sensor networks,” *Pervasive Computing, IEEE*, vol. 2, no. 4, pp. 34–42, 2003.
- [35] E. R. Carney, “Apparatus and method for multiple target engagement,” Aug. 27 1991. US Patent 5,042,743.
- [36] C. Schumacher, “Ground moving target engagement by cooperative uavs,” in *American Control Conference, 2005. Proceedings of the 2005*, pp. 4502–4505, IEEE, 2005.
- [37] S. J. Rasmussen, J. W. Mitchell, P. R. Chandler, C. J. Schumacher, and A. L. Smith, “Introduction to the multiuav2 simulation and its application to cooperative control

- research,” in *American Control Conference, 2005. Proceedings of the 2005*, pp. 4490–4501, IEEE, 2005.
- [38] H. Chen, Z. Liu, Y. Sun, and Y. Li, “Particle swarm optimization based on genetic operators for sensor-weapon-target assignment,” in *Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on*, vol. 2, pp. 170–173, IEEE, 2012.
- [39] Y. Bar-Shalom, R. Larson, and M. Grossberg, “Application of stochastic control theory to resource allocation under uncertainty,” *Automatic Control, IEEE Transactions on*, vol. 19, no. 1, pp. 1–7, 1974.
- [40] Y. Bar-Shalom, “Performance adaptive resource allocation in a time-varying environment,” in *Decision and Control including the 15th Symposium on Adaptive Processes, 1976 IEEE Conference on*, vol. 15, pp. 1075–1080, IEEE, 1976.
- [41] J. M. Nash, “Optimal allocation of tracking resources,” in *IEEE Conference on Decision Control*, vol. 16, pp. 1177–1180, IEEE, 1977.
- [42] V. Lesser, C. L. Ortiz Jr, and M. Tambe, *Distributed sensor networks: A multiagent perspective*, vol. 9. Springer Netherlands, 2003.
- [43] L. Song, D. Niyato, Z. Han, and E. Hossain, “Game-theoretic resource allocation methods for device-to-device (d2d) communication,” *arXiv preprint arXiv:1403.5723*, 2014.
- [44] M. Haghghi, “Market-based resource allocation for energy-efficient execution of multiple concurrent applications in wireless sensor networks,” in *Mobile, Ubiquitous, and Intelligent Computing*, pp. 173–178, Springer, 2014.
- [45] X. Lu, P. Wang, D. Niyato, and Z. Han, “Resource allocation in wireless networks with rf energy harvesting and transfer,” *arXiv preprint arXiv:1405.5630*, 2014.
- [46] R. Hecht-Nielsen, “Theory of the backpropagation neural network,” in *Neural Networks, 1989. IJCNN., International Joint Conference on*, pp. 593–605, IEEE, 1989.

- [47] J. Figuera-Nazuno, *Neural Networks: A Comprehensive Foundation*. No. 2nd ed, Jul. 998.
- [48] K. Obermayer and T. J. Sejnowski, *Self-organizing map formation: foundations of neural computation*, vol. 93. Mit Press, 2001.
- [49] T. Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
- [50] H. Hatime, R. Pendse, and J. M. Watkins, “Comparative study of task allocation strategies in multirobot systems,” *Sensors Journal, IEEE*, vol. 13, no. 1, pp. 253–262, 2013.
- [51] M. Liu, H. Li, Y. Shen, J. Fan, and S. Huang, “Elastic neural network method for multi-target tracking task allocation in wireless sensor network,” *Computers & Mathematics with Applications*, vol. 57, no. 11, pp. 1822–1828, 2009.
- [52] L. Haihao, L. Mei, S. Yi, and Q. Deli, “Research on task allocation technique for multi-target tracking in wireless sensor network,” in *International Conference on Mechatronics and Automation*, pp. 360–365, IEEE, 2007.
- [53] S. Shams, “Neural network optimization for multi-target multi-sensor passive tracking,” *Proceedings of the IEEE*, vol. 84, no. 10, pp. 1442–1457, 1996.
- [54] S. Kraus, “Automated negotiation and decision making in multiagent environments,” in *Multi-agent systems and applications*, pp. 150–172, Springer, 2001.
- [55] P. Sujit, A. Sinha, and D. Ghose, “Multiple uav task allocation using negotiation,” in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 471–478, ACM, 2006.
- [56] M. J. Matarić, G. S. Sukhatme, and E. H. Østergaard, “Multi-robot task allocation in uncertain environments,” *Autonomous Robots*, vol. 14, no. 2-3, pp. 255–263, 2003.
- [57] D. Goldberg, V. Ciciello, M. B. Dias, R. Simmons, S. Smith, and A. Stentz, “Task allocation using a distributed market-based planning mechanism,” in *Proceedings*

of the second international joint conference on autonomous agents and multiagent systems, pp. 996–997, ACM, 2003.

- [58] N. Edalat, W. Xiao, M. Motani, N. Roy, and S. K. Das, “Auction-based task allocation with trust management for shared sensor networks,” *Security and Communication Networks*, vol. 5, no. 11, pp. 1223–1234, 2012.
- [59] A. Charlish, K. Woodbridge, and H. Griffiths, “Multi-target tracking control using continuous double auction parameter selection,” in *Information Fusion (FUSION), 2012 15th International Conference on*, pp. 1269–1276, IEEE, 2012.
- [60] D. Shen, G. Chen, K. Pham, and E. Blasch, “A trust-based sensor allocation algorithm in cooperative space search problems,” in *SPIE Defense, Security, and Sensing*, pp. 80440C–80440C, International Society for Optics and Photonics, 2011.
- [61] A. Rubinstein, “Perfect equilibrium in a bargaining model,” *Econometrica: Journal of the Econometric Society*, pp. 97–109, 1982.
- [62] C. L. Ortiz Jr, T. W. Rauenbusch, E. Hsu, and R. Vincent, “Dynamic resource-bounded negotiation in non-additive domains,” *Distributed sensor networks: a multiagent perspective*, 2003.
- [63] D. Cieslik, “Two classical optimization problems,” *Shortest Connectivity: An Introduction with Applications in Phylogeny*, pp. 1–20, 2005.
- [64] K. Macarthur, *Multi-agent coordination for dynamic decentralised task allocation*. PhD thesis, University of Southampton, 2011.
- [65] A. Farinelli, A. Rogers, and N. R. Jennings, “Agent-based decentralised coordination for sensor networks using the max-sum algorithm,” *Autonomous agents and multiagent systems*, vol. 28, no. 3, pp. 337–380, 2014.
- [66] R. Zivan, H. Yedidsion, S. Okamoto, R. Grinton, and K. Sycara, “Distributed constraint optimization for teams of mobile sensing agents,” *Autonomous Agents and Multi-Agent Systems*, pp. 1–42, 2014.

- [67] M. Cardei, M. T. Thai, Y. Li, and W. Wu, “Energy-efficient target coverage in wireless sensor networks,” in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, pp. 1976–1984, IEEE, 2005.
- [68] H. Liu, P. Wan, C.-W. Yi, X. Jia, S. Makki, and N. Pissinou, “Maximal lifetime scheduling in sensor surveillance networks,” in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4, pp. 2482–2491, IEEE, 2005.
- [69] X. Shen and P. K. Varshney, “Sensor selection based on generalized information gain for target tracking in large sensor networks,” 2013.
- [70] O. Yadgar, S. Kraus, and C. Ortiz, “Hierarchical information combination in large-scale multi agent resource management,” *Communication in Multiagent Systems*, pp. 129–145, 2003.
- [71] B. Horling, R. Mailler, M. Sims, and V. Lesser, “Using and maintaining organization in a large-scale distributed sensor network,” in *Proceedings of the Workshop on Autonomy, Delegation, and Control*, 2003.
- [72] L. R. Foulds, *Optimization techniques*. Springer, 1981.
- [73] L. A. Wolsey and G. L. Nemhauser, *Integer and combinatorial optimization*, vol. 18. Wiley New York, 1988.
- [74] G. Sierksma, *Linear and integer programming: theory and practice*. CRC Press, 2001.
- [75] M. Jain, M. Taylor, M. Tambe, and M. Yokoo, “Dcops meet the real world: Exploring unknown reward matrices with applications to mobile sensor networks,” *in practice*, vol. 20, no. 15, p. 19, 2009.
- [76] K. Mizuno, Y. Fukui, and S. Nishihara, “Urban traffic signal control based on distributed constraint satisfaction,” in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, pp. 65–75, 2008.

- [77] D. de Oliveira, A. L. Bazzan, and V. Lesser, “Using cooperative mediation to coordinate traffic lights: a case study,” in *AAMAS’05*, pp. 463–470, 2005.
- [78] S. Tadokoro, H. Kitano, T. Takahashi, I. Noda, H. Matsubara, A. Shinjoh, T. Koto, K. Takeuchi, F. Matsuno, and M. Hatayama, “The robocup-rescue project: a robotic approach to the disaster mitigation problem,” in *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 4089–4094, 2000.
- [79] H. W. Chun, “Train timetable and route generation using a constraint-based approach,” in *Proceedings of the 10th international conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pp. 441–448, Goose Pond Press, 1997.
- [80] S. H. Semnani and K. Zamanifar, “Using maxcapo to solve comprehensive university timetabling problem considering intercollegiate courses,” in *Proceedings of the 3th joint congress on fuzzy and intelligent systems*, 2009.
- [81] K. Sycara, S. F. Roth, N. Sadeh, and M. S. Fox, “Distributed constrained heuristic search,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, pp. 1446–1461, 1991.
- [82] M. Yokoo, T. Ishida, E. H. Durfee, and K. Kuwabara, “Distributed constraint satisfaction for formalizing distributed problem solving,” pp. 614–621, 1992.
- [83] T. Schiex, H. Fargier, and G. Verfaillie, “Valued constraint satisfaction problems: Hard and easy problems,” in *IJCAI*, pp. 631–639, 1995.
- [84] S. Bistarelli, U. Montanari, and F. Rossi, “Semiring-based constraint solving and optimization,” *Journal of the ACM*, pp. 201–236, 1997.
- [85] R. Mailler, “Comparing two approaches to dynamic, distributed constraint satisfaction,” in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pp. 1049–1056, ACM, 2005.

- [86] C. Fernández, R. Béjar, B. Krishnamachari, and C. Gomes, “Communication and computation in distributed csp algorithms,” in *Eighth International Conference on Principles and Practice of Constraint Programming*, pp. 1–15, 2002.
- [87] R. Bejar, B. Krishnamachari, C. Gomes, and B. Selman, “Distributed constraint satisfaction in a wireless sensor tracking system,” in *Workshop on Distributed Constraint Reasoning, International Joint Conference on Artificial Intelligence, Seattle, Washington*, 2001.
- [88] R. Béjar, C. Domshlak, C. Fernández, C. Gomes, B. Krishnamachari, B. Selman, and M. Valls, “Sensor networks and distributed csp: communication, computation and complexity,” *Artificial Intelligence*, pp. 117 – 147, 2005.
- [89] R. Mailler and V. R. Lesser, “Asynchronous partial overlay: A new algorithm for solving distributed constraint satisfaction problems,” *Journal of Artificial Intelligence Research*, vol. 25, pp. 529–576, 2006.
- [90] M. Yokoo, “Asynchronous weak-commitment search for solving distributed constraint satisfaction problems,” in *Principles and Practice of Constraint Programming*, pp. 88–102, Springer, 1995.
- [91] B. Satzger, F. Bagci, F. Kluge, and T. Ungerer, “Towards lightweight self-configuration in wireless sensor networks,” in *Proceedings of the 2010 ACM Symposium on Applied Computing*, pp. 791–792, ACM, 2010.
- [92] T. Matsui and H. Matsuo, “A formalization for distributed cooperative sensor resource allocation,” in *Agent and Multi-Agent Systems: Technologies and Applications*, pp. 292–301, 2008.
- [93] K. Ota, T. Matsui, and H. Matsuo, “Layered distributed constraint optimization problem for resource allocation problem in distributed sensor networks,” *Principles of Practice in Multi-Agent Systems*, pp. 245–260, 2009.
- [94] T. Matsui and H. Matsuo, “A constraint based formalisation for distributed cooperative sensor resource allocation,” *International Journal of Intelligent Information and Database Systems*, vol. 4, pp. 307–321, 2010.

- [95] G.-Y. Zhang, J.-C. Zeng, and S.-D. Xue, “Research on task allocation of multi-target search with swarm robots,” *International Journal of Wireless and Mobile Computing*, vol. 7, no. 3, pp. 297–304, 2014.
- [96] A. Jevtic, A. Gutiérrez, D. Andina, and M. Jamshidi, “Distributed bees algorithm for task allocation in swarm of robots,” *Systems Journal, IEEE*, vol. 6, no. 2, pp. 296–304, 2012.
- [97] R. V. Kulkarni and G. K. Venayagamoorthy, “Particle swarm optimization in wireless-sensor networks: A brief survey,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 2, pp. 262–267, 2011.
- [98] X. Wang, J.-J. Ma, S. Wang, and D.-W. Bi, “Distributed particle swarm optimization and simulated annealing for energy-efficient coverage in wireless sensor networks,” *Sensors*, vol. 7, no. 5, pp. 628–648, 2007.
- [99] K. H. Low, W. K. Leow, and M. H. Ang Jr, “Task allocation via self-organizing swarm coalitions in distributed mobile sensor network,” in *AAAI*, vol. 4, pp. 28–33, 2004.
- [100] Y.-C. Tseng, S.-P. Kuo, H.-W. Lee, and C.-F. Huang, “Location tracking in a wireless sensor network by mobile agents and its data fusion strategies,” *Information Processing in Sensor Networks*, vol. 47, pp. 554–554, 2003.
- [101] J.-M. Gil and Y.-H. Han, “A target coverage scheduling scheme based on genetic algorithms in directional sensor networks,” *Sensors*, vol. 11, no. 2, pp. 1888–1906, 2011.
- [102] F. Zhou, “Energy-efficient coverage using sensors with continuously adjustable sensing ranges,” in *Natural Computation (ICNC), 2011 Seventh International Conference on*, vol. 1, pp. 109–113, IEEE, 2011.
- [103] A. Jevtić and A. Gutiérrez, “Distributed bees algorithm parameters optimization for a cost efficient target allocation in swarms of robots,” *Sensors*, vol. 11, no. 11, pp. 10880–10893, 2011.

- [104] L. Mei, H. Dao-ping, and X. Xiao-ling, “Node task allocation based on pso in wsn multi-target tracking,” *Advances in Information Sciences and Service Sciences*, vol. 2, no. 2, pp. 13–18, 2010.
- [105] J. Chen, C. Zang, W. Liang, and H. Yu, “Auction-based dynamic coalition for single target tracking in wireless sensor networks,” in *The Sixth World Congress on Intelligent Control and Automation*, pp. 94–98, IEEE, 2006.
- [106] D. Pei, P. Song, X. Han, and K. Li, “The cooperative hunting research of mobile wireless sensor network based on improved dynamic alliance,” *Intelligent Computing and Information Science*, pp. 599–605, 2011.
- [107] C. M. Bishop, *Pattern recognition and machine learning*, vol. 1. springer New York, 2006.
- [108] A. Campbell and A. S. Wu, “Learning and exploiting knowledge in multi-agent task allocation problems,” in *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, pp. 2637–2642, ACM, 2007.
- [109] C. Kreucher, D. Blatt, A. Hero, and K. Kastella, “Adaptive multi-modality sensor scheduling for detection and tracking of smart targets,” *Digital Signal Processing*, vol. 16, no. 5, pp. 546–567, 2006.
- [110] G. Mainland, D. C. Parkes, and M. Welsh, “Decentralized, adaptive resource allocation for sensor networks,” in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, pp. 315–328, USENIX Association, 2005.
- [111] M. Azarbayejani, A. El-Osery, K. Choi, and M. R. Taha, “A probabilistic approach for optimal sensor allocation in structural health monitoring,” *Smart Materials and Structures*, vol. 17, no. 5, p. 055019, 2008.
- [112] J. Butterfield, O. C. Jenkins, and B. Gerkey, “Multi-robot markov random fields,” in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, pp. 1211–1214, International Foundation for Autonomous Agents and Multiagent Systems, 2008.

- [113] Y. Bar-Shalom, T. Kirubarajan, and X. Lin, “Probabilistic data association techniques for target tracking with applications to sonar, radar and eo sensors,” *Aerospace and Electronic Systems Magazine, IEEE*, vol. 20, no. 8, pp. 37–56, 2005.
- [114] D. Wang, J. Liu, and Q. Zhang, “Probabilistic field coverage using a hybrid network of static and mobile sensors,” in *Quality of Service, 2007 Fifteenth IEEE International Workshop on*, pp. 56–64, IEEE, 2007.
- [115] P. Wang, B. D. Youn, and C. Hu, “A probabilistic detectability-based structural sensor network design methodology for prognostics and health management,” in *Annual conference of the prognostics and health management society*, 2010.
- [116] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization based on virtual forces,” in *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, pp. 1293–1303, 2003.
- [117] L. Bertuccelli, M. Alighanbari, and J. How, “Robust planning for coupled cooperative uav missions,” in *43rd IEEE Conference on Decision and Control*, vol. 3, pp. 2917–2922, IEEE, 2004.
- [118] R. Vidal, O. Shakernia, H. J. Kim, H. Shim, and S. Sastry, “Multi-agent probabilistic pursuit-evasion games with unmanned ground and aerial vehicles,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 662–669, 2002.
- [119] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, “Cooperative air and ground surveillance,” *Robotics & Automation Magazine, IEEE*, vol. 13, no. 3, pp. 16–25, 2006.
- [120] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [121] H. Shu and Q. Liang, “Fuzzy optimization for distributed sensor deployment,” in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 3, pp. 1903–1908, IEEE, 2005.

- [122] Z. Tang and U. Ozguner, “Motion planning for multitarget surveillance with mobile sensor agents,” *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 898–908, 2005.
- [123] J. S. Bellingham, M. Tillerson, M. Alighanbari, and J. P. How, “Cooperative path planning for multiple uavs in dynamic and uncertain environments,”
- [124] A. Petcu and B. Faltings, “A scalable method for multiagent constraint optimization,” in *International Joint Conference on Artificial Intelligence*, vol. 19, p. 266271, LAWRENCE ERLBAUM ASSOCIATES LTD, 2005.
- [125] J. S. Bellingham, M. Tillerson, M. Alighanbari, and J. P. How, “A region-based approach for cooperative multi-target tracking in a structured environment,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2764–2769, IEEE, 2002.
- [126] P. Ogren, E. Fiorelli, and N. E. Leonard, “Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment,” *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.
- [127] A. Richards and J. How, “Decentralized model predictive control of cooperating uavs,” in *43rd IEEE Conference on Decision and Control*, vol. 4, pp. 4286–4291, IEEE, 2004.
- [128] L. E. Parker and B. A. Emmons, “Cooperative multi-robot observation of multiple moving targets,” in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 3, pp. 2082–2089, IEEE, 1997.
- [129] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.
- [130] R. Olfati-Saber and P. Jalalkamali, “Coupled distributed estimation and control for mobile sensor networks,” *Automatic Control, IEEE Transactions on*, vol. 57, no. 10, pp. 2609–2614, 2012.

- [131] Y.-Q. Miao, A. Khamis, and M. S. Kamel, “Applying anti-flocking model in mobile surveillance systems,” in *Autonomous and Intelligent Systems (AIS), 2010 International Conference on*, pp. 1–6, IEEE, 2010.
- [132] R. Olfati-Saber, “Distributed tracking for mobile sensor networks with information-driven mobility,” in *American Control Conference, 2007. ACC’07*, pp. 4606–4612, IEEE, 2007.
- [133] P. Jalalkamali and R. Olfati-Saber, “Information-driven self-deployment and dynamic sensor coverage for mobile sensor networks,” in *American Control Conference (ACC), 2012*, pp. 4933–4938, IEEE, 2012.
- [134] H. M. La and W. Sheng, “Dynamic target tracking and observing in a mobile sensor network,” *Robotics and Autonomous Systems*, vol. 60, no. 7, pp. 996–1009, 2012.
- [135] S. A. Quintero, G. E. Collins, and J. P. Hespanha, “Flocking with fixed-wing uavs for distributed sensing: A stochastic optimal control approach,” in *American Control Conference (ACC), 2013*, pp. 2025–2031, IEEE, 2013.
- [136] Z. Wang and D. Gu, “Cooperative target tracking control of multiple robots,” *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 8, pp. 3232–3240, 2012.
- [137] S. H. Semnani and O. A. Basir, “Semi-flocking algorithm for motion control of mobile sensors in large-scale surveillance systems,” vol. 45, no. 1, pp. 129–137, 2015.
- [138] M. Hannebauer and S. Müller, “Distributed constraint optimization for medical appointment scheduling,” in *Proceedings of the fifth international conference on Autonomous agents*, pp. 139–140, ACM, 2001.
- [139] R. Junges and A. L. Bazzan, “Evaluating the performance of dcop algorithms in a real world, dynamic problem,” in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pp. 599–606, International Foundation for Autonomous Agents and Multiagent Systems, 2008.
- [140] R. Mailler and V. Lesser, “Using cooperative mediation to solve distributed constraint satisfaction problems,” in *Proceedings of the Third International Joint Conference on*

Autonomous Agents and Multiagent Systems-Volume 1, pp. 446–453, IEEE Computer Society, 2004.

- [141] K. Hirayama and M. Yokoo, “The distributed breakout algorithms,” *Artificial Intelligence*, vol. 161, no. 1-2, pp. 89–115, 2005.
- [142] M. Yokoo and K. Hirayama, “Distributed breakout algorithm for solving distributed constraint satisfaction problems,” in *Proceedings of the Second International Conference on Multi-Agent Systems*, pp. 401–408, 1996.
- [143] L. Wittenburg and W. Zhang, “Distributed breakout algorithm for distributed constraint optimization problems—dbarelay,” in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pp. 1158–1159, ACM, 2003.
- [144] A. Meisels, E. Kaplansky, I. Razgon, and R. Zivan, “Comparing performance of distributed constraints processing algorithms,” in *Proceedings workshop on distributed constraint reasoning (AAMAS)*, pp. 86–93, 2002.
- [145] M. Benisch and N. Sadeh, “Examining dcsp coordination tradeoffs,” in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 1405–1412, ACM, 2006.
- [146] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, “Mobility improves coverage of sensor networks,” in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pp. 300–308, ACM, 2005.
- [147] B. Krishnamachari, S. B. Wicker, and R. Bejar, “Phase transition phenomena in wireless ad hoc networks,” in *Global Telecommunications Conference, 2001. GLOBECOM’01. IEEE*, vol. 5, pp. 2921–2925, IEEE, 2001.
- [148] L.-K. Soh and C. Tsatsoulis, “Reflective negotiating agents for real-time multisensor target tracking,” in *International Joint Conference On Artificial Intelligence*, vol. 17, pp. 1121–1127, LAWRENCE ERLBAUM ASSOCIATES LTD, 2001.

- [149] “BIWI Walking Pedestrians dataset.” <http://www.vision.ee.ethz.ch/datasets>, 2014. [Online; accessed March 19, 2014].
- [150] X. R. Li and V. P. Jilkov, “Survey of maneuvering target tracking. part i. dynamic models,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, no. 4, pp. 1333–1364, 2003.
- [151] N. H. Gholson and R. L. Moose, “Maneuvering target tracking using adaptive state estimation,” *Aerospace and Electronic Systems, IEEE Transactions on*, no. 3, pp. 310–317, 1977.
- [152] D. W. Gage, “Command control for many-robot systems,” tech. rep., DTIC Document, 1992.
- [153] D. W. Gage, “Randomized search strategies with imperfect sensors,” in *Optical Tools for Manufacturing and Advanced Automation*, pp. 270–279, International Society for Optics and Photonics, 1994.
- [154] B. Moshiri and F. Besharati, “Maneuvering target tracking,” *WSEAS Transactions on Circuits and Systems*, vol. 3, no. 1, pp. 176–182, 2004.
- [155] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, “Constrained k-means clustering with background knowledge,” in *ICML*, vol. 1, pp. 577–584, 2001.
- [156] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, California, USA, 1967.
- [157] P. Bradley, K. Bennett, and A. Demiriz, “Constrained k-means clustering,” *Microsoft Research, Redmond*, pp. 1–8, 2000.
- [158] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 2, pp. 224–227, 1979.
- [159] Y. Bar-Shalom and X.-R. Li, “Multitarget-multisensor tracking: principles and techniques,” *Storrs, CT: University of Connecticut, 1995.*, 1995.

- [160] P. Barooah, W. J. Russell, and J. P. Hespanha, “Approximate distributed kalman filtering for cooperative multi-agent localization,” in *Distributed Computing in Sensor Systems*, pp. 102–115, Springer, 2010.
- [161] R. Olfati-Saber, “Distributed kalman filtering for sensor networks,” in *Decision and Control, 2007 46th IEEE Conference on*, pp. 5492–5498, IEEE, 2007.
- [162] P. Zhang and M. Martonosi, “Locale: Collaborative localization estimation for sparse mobile sensor networks,” in *Information Processing in Sensor Networks, 2008. IPSN’08. International Conference on*, pp. 195–206, IEEE, 2008.
- [163] R. Olfati-Saber, “Kalman-consensus filter: Optimality, stability, and performance,” in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 7036–7042, IEEE, 2009.
- [164] C. Soto, B. Song, and A. K. Roy-Chowdhury, “Distributed multi-target tracking in a self-configuring camera network,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1486–1493, IEEE, 2009.
- [165] C. Fernández, R. Béjar, B. Krishnamachari, and C. Gomes, “Communication and computation in distributed csp algorithms,” in *Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming*, pp. 664–679, Springer-Verlag, 2002.
- [166] S. H. Semnani and O. A. Basir, “Target to sensor allocation: A hierarchical dynamic distributed constraint optimization approach,” *Computer Communications*, vol. 36, no. 9, pp. 1024–1038, 2013.
- [167] S. H. Semnani and O. A. Basir, “A hierarchical constraint optimization approach for target allocation in large-scale sensor networks,” in *Proceedings of 2011 symposium on advanced intelligent systems*, pp. 74–75, CPAMI, 2011.
- [168] L. M. Kaplan, “Transmission range control during autonomous node selection for wireless sensor networks,” in *Aerospace Conference*, vol. 3, pp. 2072–2087, IEEE, 2004.

- [169] D. A. Castanon, "Approximate dynamic programming for sensor management," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 2, pp. 1202–1207, IEEE, 1997.
- [170] N. Enami, R. A. Moghadam, and A. Haghghat, "A survey on application of neural networks in energy conservation of wireless sensor networks," *Recent Trends in Wireless and Mobile Networks*, pp. 283–294, 2010.
- [171] I. B. Jovanoski and D. Davcev, "Distributed constraint satisfaction model adaptation for balanced energy fading in wireless sensor networks," in *International Conference on Industrial Informatics*, pp. 636–640, IEEE, 2006.
- [172] B. Krishnamachari, R. Béjar, and S. Wicker, "Distributed constraint satisfaction and the bounds on resource allocation in wireless networks," in *Sixth International Symposium on Communications Theory and Application*, 2001.
- [173] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg, "Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks," *Artificial Intelligence*, vol. 161, no. 1-2, pp. 55–87, 2005.
- [174] W. Zhang, Z. Xing, G. Wang, and L. Wittenburg, "An analysis and application of distributed constraint satisfaction and optimization algorithms in sensor networks," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pp. 14–18, 2003.
- [175] W. Zhang, G. Wang, and L. Wittenburg, "Distributed stochastic search for constraint satisfaction and optimization: Parallelism, phase transitions and performance," in *Proceedings of AAAI Workshop on Probabilistic Approaches in Search*, 2002.
- [176] L. Doherty and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1655–1663, 2001.

- [177] I. Amundson, M. Kushwaha, X. Koutsoukos, S. Neema, and J. Sztipanovits, “Efficient integration of web services in ambient-aware sensor network applications,” in *3rd International Conference on Broadband Communications, Networks and Systems, 2006*, pp. 1–8, IEEE, 2006.
- [178] M. Kushwaha, I. Amundson, X. Koutsoukos, S. Neema, and J. Sztipanovits, “Oasis: A programming framework for service-oriented sensor networks,” in *2nd International Conference on Communication Systems Software and Middleware*, pp. 1–8, IEEE, 2007.
- [179] F. Mourad, H. Snoussi, F. Abdallah, and C. Richard, “Anchor-based localization via interval analysis for mobile ad-hoc sensor networks,” *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3226–3239, 2009.
- [180] H. Fouchal and Z. Habbas, “Distributed backtracking algorithm based on tree decomposition over wireless sensor networks,” *Concurrency and Computation: Practice and Experience*, 2011.
- [181] S. Tornil-Sin, C. Ocampo-Martinez, V. Puig, and T. Escobet, “Robust fault detection of non-linear systems using set-membership state estimation based on constraint satisfaction,” *Engineering Applications of Artificial Intelligence*, p. 110, 2012.
- [182] T. Sandholm, “Contract types for satisficing task allocation,” in *Proceedings of the AAAI spring symposium: Satisficing models*, pp. 23–25, 1998.
- [183] A. Ernst, H. Jiang, and M. Krishnamoorthy, “Exact solutions to task allocation problems,” *Management science*, vol. 52, no. 10, pp. 1634–1646, 2006.
- [184] M. Yokoo and K. Hirayama, “Algorithms for distributed constraint satisfaction: A review,” *Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 2, pp. 185–207, 2000.
- [185] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

- [186] W. S. Conner, L. Krishnamurthy, and R. Want, “Making everyday life easier using dense sensor networks,” in *UbiComp 2001: Ubiquitous Computing*, pp. 49–55, Springer, 2001.
- [187] H. University, “Codeblue: Wireless sensors for medical care,” jan 2012.
- [188] L. Bertuccelli, M. Alighanbari, and J. How, “Robust planning for coupled cooperative uav missions,” in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 3, pp. 2917–2922, IEEE, 2004.
- [189] “21 ideas for the 21st century,” in *Business Week*.
- [190] A. O. Hero and D. Cochran, “Sensor management: Past, present, and future,” *Sensors Journal, IEEE*, vol. 11, no. 12, pp. 3064–3075, 2011.
- [191] C. Huaiping, L. Jingxu, C. Yingwu, and W. Hao, “Survey of the research on dynamic weapon-target assignment problem,” *Systems Engineering and Electronics, Journal of*, vol. 17, no. 3, pp. 559–565, 2006.

List of Publications Resulting from This Research Work

Refereed Journal papers:

Published

- **Hosseini Semnani S.**, Basir O. A. (2014), Semi-Flocking Algorithm for Motion Control of Mobile Sensors in Large-scale Surveillance Systems, IEEE Transactions on Cybernetics, Volume pp. Issue: 99, DOI: 10.1109/TCYB.2014.2328659.
- **Hosseini Semnani S.**, Basir O. A. (2013), Target to Sensor Allocation: A Hierarchical Dynamic Distributed Constraint Optimization Approach, Computer Communications, Volume 36, Issue 9, 15 May 2013, Pages 10241038.

Submitted:

- **Hosseini Semnani S.**, Basir O. A. (2014), Robust Collaborative Multi-target Tracking Using Semi-Flocking Sensor Networks. Submitted for consideration as a Full Paper on IEEE Transactions on Automatic Control.
- **Hosseini Semnani S.**, Basir O. A. (2014), Multi-Target Engagement in complex Mobile Surveillance Sensor Networks. Submitted for consideration as a Full Paper on IEEE Sensors Journal.

Under review:

- **Hosseini Semnani S.**, Basir O. A., Van Beek P. (2014), Constrained Clustering for Flocking-based Tracking in Maneuverable Target Environment.

Conference papers, posters, talks:

- **Hosseini Semnani S.**, Basir O. A., Target-to-sensor Allocation in Large-scale Sensor Networks, CASCON 2012, Markham, Ontario.
- **Hosseini Semnani S.**, Basir O. A., Multi-Target Tracking in Large-Scale Surveillance Systems. MUSES workshop 2013.
- **Hosseini Semnani S.**, Basir O. A., A Hierarchical Constraint Optimization Approach for Target Allocation in large-Scale Sensor Networks, CPAMI Symposium 2011.
- **Hosseini Semnani S.**, Basir O. A., Semi-Flocking: a Multi-Target Tracking Approach for Large-Scale Surveillance Systems, CPAMI Symposium 2012.