NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DISTRIBUTED SUBMODULAR OPTIMIZATION FOR A UXV NETWORKED CONTROL SYSTEM

by

Bryan Lowry

March 2020

Thesis Advisor: Douglas P. Horner
Co-Advisor: Vladimir N. Dobrokhodov

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
DISTRIBUTED SUBMODULAR OPTIMIZATION FOR A UXV NETWORKED CONTROL SYSTEM

5. FUNDING NUMBERS

6. AUTHOR(S) Bryan Lowry

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

A networked control system (NCS) is a single system composed of multiple discrete agents, and is often modeled as a graph of nodes (agents) and connecting edges (communications links). NCS technology is increasingly important for control of autonomous vehicles (UxVs). Proper placement of the NCS elements is vital to mission effectiveness, but finding optimal NCS configurations is almost always computationally prohibitive. In this thesis, we explore using submodularity to solve this problem. This approach gives near-optimality guarantees while reducing the complexity of the problem, allowing solutions to be found in near real-time. A simple, novel distributed submodular approach to NCS control was developed, with simulation and analysis provided for an expeditionary warfare scenario.

14. SUBJECT TERMS
networked control system, distributed network control, adaptive submodularity, network robustness, multi-domain, sensor selection, sensor placement

15. NUMBER OF PAGES
89

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

DISTRIBUTED SUBMODULAR OPTIMIZATION FOR A UXV NETWORKED CONTROL SYSTEM

Bryan Lowry
Lieutenant, United States Navy
BS, University of Maryland, 2013

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2020

Approved by: Douglas P. Horner
Advisor

Vladimir N. Dobrokhodov
Co-Advisor

Garth V. Hobson
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A networked control system (NCS) is a single system composed of multiple discrete agents, and is often modeled as a graph of nodes (agents) and connecting edges (communications links). NCS technology is increasingly important for control of autonomous vehicles (UxVs). Proper placement of the NCS elements is vital to mission effectiveness, but finding optimal NCS configurations is almost always computationally prohibitive. In this thesis, we explore using submodularity to solve this problem. This approach gives near-optimality guarantees while reducing the complexity of the problem, allowing solutions to be found in near real-time. A simple, novel distributed submodular approach to NCS control was developed, with simulation and analysis provided for an expeditionary warfare scenario.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction **1**

1.1 Introduction 1

1.2 Motivation 1

1.3 Problem Statement. 1

1.4 Case Scenario 2

1.5 Related Work 3

1.6 Overview 3

2 Background **5**

2.1 Networked Control Systems 5

2.2 Graph Theory. 9

2.3 Submodularity 11

2.4 Utility Function 14

2.5 Centralized Submodular Optimization 16

3 Distributed Submodularity **21**

3.1 The Distributed Online Greedy Algorithm 21

3.2 Distributed Submodularity Method 23

3.3 DS Example 28

3.4 Summary 45

4 Results **47**

4.1 MTX Scenario 47

4.2 MTX Results 49

5 Analysis **57**

5.1 Network Size 57

5.2 Complementary and Competing Subfunctions 59

5.3 Observations 60

6 Conclusion **63**
6.1 Contributions 63
6.2 Future Work 63

List of References **65**

Initial Distribution List **69**

List of Figures

Figure 2.1 Example NCS represented as a graph. 10

Figure 2.2 Gridded map of San Clemente Island 12

Figure 2.3 Networked control system (NCS) after first node placement . . . 18

Figure 2.4 NCS after all nodes placed. 19

Figure 2.5 NCS placement, centralized method. 20

Figure 3.1 Simple NCS setup. 28

Figure 3.2 Simple NCS, centralized method. 29

Figure 3.3 Simple NCS, DS results. 30

Figure 3.4 Simple NCS, utility plots. 31

Figure 3.5 Simple NCS, DS results, initial configuration 2. 32

Figure 3.6 Simple NCS, utility plots, initial configuration 2. 32

Figure 3.7 Simple NCS, DS results, initial configuration 3. 33

Figure 3.8 Simple NCS, utility plots, initial configuration 3. 33

Figure 3.9 Simple NCS, DS results, initial configuration 4. 34

Figure 3.10 Simple NCS, utility plots, initial configuration 4. 34

Figure 3.11 Simple NCS, local search area 25%. 35

Figure 3.12 Simple NCS, utility plots, local search area 25%. 35

Figure 3.13 Simple NCS, local search area 37.5%. 36

Figure 3.14 Simple NCS, utility plots, local search area 37.5%. 37

Figure 3.15 Simple NCS, local search area 50%. 37

Figure 3.16 Simple NCS, utility plots, local search area 50%. 38

Figure 3.17 Simple **F** with decreased emphasis on sensing. 39
Figure 3.18 Simple NCS with decreased emphasis on sensing. 39
Figure 3.19 Simple NCS with decreased emphasis on sensing, utility plots. . 40
Figure 3.20 **F** with competing subfunctions. 41
Figure 3.21 NCS with competing subfunctions. 41
Figure 3.22 NCS with competing subfunctions, utility plots. 42
Figure 3.23 Communications decay models. 43
Figure 3.24 NCS with greater variance in communications robustness. 44
Figure 3.25 NCS with greater variance in communications robustness, utility plots. 44

Figure 4.1 San Clemente Island (SCI) with mission objective. 47
Figure 4.2 Multi-Thread Exercise (MTX) sensing potential fields. 49
Figure 4.3 Intelligence Preparation of the Battlefield (IPB) phase, start. . . . 50
Figure 4.4 IPB phase, end. 50
Figure 4.5 Total NCS utility throughout the IPB phase. 51
Figure 4.6 Insertion phase, start. 52
Figure 4.7 Insertion phase, middle. 52
Figure 4.8 Insertion phase, end. 53
Figure 4.9 Total NCS utility throughout the insertion phase. 53
Figure 4.10 Infiltration phase, start. 54
Figure 4.11 Infiltration phase, early. 54
Figure 4.12 Infiltration phase, late. 55
Figure 4.13 Infiltration phase, end. 55
Figure 4.14 Total NCS utility throughout the infiltration phase. 56

Figure 5.1 Size scaling, five and seven nodes. 57
Figure 5.2 Size scaling, 10 and 15 nodes. 58
Figure 5.3 Size scaling, 20 and 40 nodes. 58
Figure 5.4 Complementary subfunctions. 59
Figure 5.5 Competing subfunctions. 60

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

Table 2.1 MTX agent characteristics. 17

Table 4.1 MTX mission phases. 48

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

ASW	Anti-Submarine Warfare
C2	Command and Control
CRUSER	Consortium for Robotics and Unmanned Systems Education and Research
DCTS	Decentralized Monte-Carlo Tree Search
DDG	U.S. Navy Destroyer
DoD	Department of Defense
DOG	Distributed Online Greedy
DS	Distributed Submodularity
DSP	DS Position
DTN	Disruption-Tolerant Networking
EWA	Exponentially Weighted Average
IP	Internet Protocol
IPB	Intelligence Preparation of the Battlefield
ISR	Intelligence, Surveillance, and Reconnaissance
LTI	Linear Time-Invariant
MAB	Multi-Armed Bandit
MCTS	Monte Carlo tree search
MTX	Multi-Thread Exercise
NCS	Networked Control System

NPS	Naval Postgraduate School
NSW	Naval Special Warfare
SCI	San Clemente Island
UAV	Unmanned Aerial Vehicle
USN	U.S. Navy
USV	Unmanned Surface Vehicle
UUV	Unmanned Underwater Vehicle
UxVs	Unmanned Vehicles
VL	Virtual Leader

Acknowledgments

I would like to thank my advisor, Dr. Douglas Horner, for his invaluable guidance and assistance provided throughout my research. Dr. Horner's expertise and vision were instrumental in focusing my efforts and directing my work toward an important contribution in this field of study.

I would also like to thank my co-advisor, Dr. Vladimir Dobrokhodov, and Dr. Geoffrey Xie for their knowledge, patience, and assistance in helping me wrestle with difficult problems.

Finally, and foremost, I would like to thank my wife Krista for her constant and understanding support during my time here. Knowing our home and family was in her continually able care allowed me to focus on completing this thesis. Additionally, her knowledge of English being far superior to mine, I am greatly thankful for the much needed proofreading she provided.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

1.1 Introduction

In this thesis, a distributed submodular framework for controlling a networked control system (NCS) is developed. The NCS comprises both manned assets and multi-domain Unmanned Vehicles (UxVs), each having distinct sensing, mobility, and communications capabilities. Given a NCS consisting of aerial, surface, and underwater UxVs, this thesis presents a distributed framework that permits the NCS to near-optimally and autonomously position platforms to better support mission objectives.

1.2 Motivation

Unmanned systems will increasingly be employed in cooperative or collaborative groups connected through communication networks. These networks increase mission effectiveness, in part, through selected collection and dissemination of information. Viewed as a single control system, the NCS can be modeled as a linear time invariant graph of nodes (manned and unmanned platforms) and edges (connecting communication networks). An important question to address is how to position the nodes in the NCS to best support mission objectives. Previous work focused on a centralized approach to near-optimal NCS control [1]. In an adversarial environment, a NCS that relies on a centralized node is subject to a single point of failure. Developing a framework that utilizes distributed information sharing among all nodes in the network enables NCS decision-making to occur without reliance on a vulnerable central node.

1.3 Problem Statement

This thesis seeks to improve the flexibility and survivability of existing control systems for a multi-vehicle network. A framework that utilizes a distributed approach to submodular function optimization is developed. This framework allows each node in the NCS to make decisions and share information in a way that ensures the failure of one or more nodes will

not significantly impact the performance of the network. Additionally, the possibility of introducing Disruption-Tolerant Networking (DTN) into the NCS to loosen communication constraints and increase mission effectiveness is explored. This research is applicable not only to the case study presented in this thesis, but to other Department of Defense (DoD) and U.S. Navy (USN) mission areas. These include Anti-Submarine Warfare (ASW) using a distributed sensor network, missile defense, theater-level Intelligence, Surveillance, and Reconnaissance (ISR), and oceanographic research.

1.4 Case Scenario

In November of 2017, the Consortium for Robotics and Unmanned Systems Education and Research (CRUSER) at the Naval Postgraduate School (NPS) conducted the Multi-Thread Exercise (MTX) on San Clemente Island (SCI), off the coast of Southern California. This experiment provided a platform to test the collaboration of an autonomous network of UxVs with manned assets. The scenario was built around a Naval Special Warfare (NSW) unit tasked with landing on SCI and traversing the island to act on a known target. Before, during, and after the mission, the NSW unit was supported by additional manned and unmanned assets operating together to provide ISR support, transportation, and battlespace awareness. This multi-domain team composed of the ScanEagle Unmanned Aerial Vehicle (UAV), the SeaFox Unmanned Surface Vehicle (USV), and the REMUS 100 Unmanned Underwater Vehicle (UUV) can be modeled as the nodes of a NCS centered around the NSW unit.

Previous work developed a centralized framework to control the NCS [1]. However, the current method relies on a centralized controller to position all nodes in the NCS. A decentralized method of control would allow continued operation of the NCS in the event of a malfunction or casualty sustained in a potentially hostile environment. Additionally, because the centralized method relies on sequential, ordered placement of the nodes in the NCS, computational delays could inhibit the performance of the network as the number of nodes increases. A decentralized NCS allows the computational burden to be distributed amongst all the nodes in the network, with each node performing independent computations concurrently, and then sharing relevant information with neighboring nodes. Lastly, the centralized NCS relies on continuous communication with all nodes and a known environment, while a decentralized one could better account for DTN and an environment characterized by incomplete information.

1.5 Related Work

Much work in distributed submodularity has focused on finding optimal sets of nodes from larger data sets. Mirzasoleiman et al. have utilized distributed submodular maximization as a machine learning tool to select representative samples from large data sets [2]. In a similar vein, Mokhtari et al. have developed an algorithm to reach consensus on a global submodular objective function by utilizing local gradient climbing and sharing information from multiple nodes [3]. Machine learning approaches have also been applied to multi-agent exploration problems. Corah and Michael address the long-horizon path-planning problem by using multiple rounds of distributed sequential greedy assignment [4]. Best et al. utilize a more complex decentralized Monte Carlo tree search (MCTS) for similar problems that may not have submodular objective functions [5]. While the current model relies on Internet Protocol (IP) communications between discrete agents, Rohrer and Xie have proposed a hybrid architecture that would allow for switching between IP and DTN communications to account for episodic connectivity, degraded link-quality, or policy-driven prioritization [6]. Golovin, Faulkner, and Krause have developed a Multi-Armed Bandit (MAB)-type algorithm for distributed sensor selection, to be used in applications such as environmental monitoring [7]. This approach is explored further in Chapter 3.

1.6 Overview

In Chapter 2 the concepts relating to the existing NCS and submodularity are reviewed, the utility functions used in the MTX model are introduced, and the centralized approach to NCS control is summarized. In Chapter 3 aspects of distributed control methods and current work in distributed submodularity are explored. Our novel approach to distributed submodularity is introduced and a simple problem is developed to test it. Chapter 4 presents the results of this distributed approach applied to the full MTX scenario, and compares them to the centralized method. In Chapter 5 the effects of network size and the complementary or competitive nature of the utility subfunctions on the relative performance of both approaches are analyzed. Chapter 6 summarizes the contributions of this thesis and possible areas of future work.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Background

This chapter reviews topics pertinent to this thesis, including NCS control theory and the relationship of graph theory to NCS. The concept of submodularity is reviewed, and the submodular utility function used by the NCS is introduced. Lastly the current approach to centralized submodular optimization, which is fully described in [1], is summarized.

2.1 Networked Control Systems

An NCS can be described as a single system composed of multiple discrete entities with control loops that share feedback and control signals over a shared communications network [8]. NCS are often described as either static or dynamic. A static NCS is one in which the agents do not change state, such as a fixed sensor or communications network. An example example of a static NCS is provided in [7], where a fixed sensor network is utilized in an environmental monitoring application. This thesis is primarily concerned with a dynamic NCS which contains one or more agents whose state varies [9]; this NCS is composed of mobile UxVs whose state (position, communications status, etc.) changes throughout the mission. More specifically, this is a NCS that is characterized by independent decision-making agents, locally sensed information, and limited communication abilities, something that Shamma describes in [10] as cooperative control.

2.1.1 Distributed NCS

Significant research has been conducted surrounding distributed NCS. In [11], Cao et al. provide a detailed overview of many topics relating to distributed NCS and multi-agent coordination including:

1. **Consensus**: agents asymptotically converge to a common agreement
2. **Distributed formation control**: agents utilize local interactions to form a pre-designed configuration
3. **Distributed optimization**: algorithms to analyze and optimize distributed systems
4. **Distributed estimation**: estimating global parameters from local information

Many consensus problems for distributed NCS have focused on cases with deterministic topologies and ideal communications; these are network qualities assumed in this thesis. However, significant research has also been done on stochastic topologies and communications models, and the implications such uncertainty has on achieving NCS consensus. Zhang and Tian have shown conditions under which consensus can be reached for stochastic systems having double-integrator dynamics [12]. This thesis assumes double-integrator dynamics for the vehicles in the NCS, as discussed in Section 2.1.2.

The study of formation control is often divided into *formation producing* algorithms and *formation tracking* algorithms. In the former, there is no group reference, and the formation is developed from a global standpoint. In the latter, a interactions with one or more group neighbors (often described as a Virtual Leader (VL)), provide the reference for the formation [11]. In [13], Cao and Ren suggest methods for coordinated leader-tracking in a system composed of vehicles with second-order dynamics. They have shown that distributed tracking of a VL of varying velocity is possible with limited information sharing over a one-hop network. In [14], Xargay et al. demonstrate a methodology for effective cooperative control and collision avoidance for multiple vehicles executing a path-following mission. This work utilizes a graph-theoretic approach similar to that used in this thesis, as discussed in Section 2.2.

In an early work on distributed optimization, Nedić and Ozdaglar provide distributed optimization techniques for convex objective functions [15]. However, in many cases the objective function being optimized is not convex. Optimizing submodular objective functions is the subject of Section 2.3. Distributed estimation is used in many settings to ascertain global information from locally sensed parameters. In a stochastic setting where the global objective is not known, it is natural to combine distributed optimization and estimation. In [3], Mokhtari et al. develop methods to reach a global optimum when each node only has knowledge of a portion of the global objective function.

2.1.2 Formation Control

A detailed discussion of the theory behind the control system used in this model can be found in [1] and is summarized here. Because this thesis assumes each agent has the ability to measure its position in the global reference frame, a position-based approach is used as

opposed to displacement or distance-based approaches [16]. The primary goal of this type of control system is to drive the agents to a desired location .

To accomplish this, each node $i = 1, 2, ..., N$ is modeled as a particle having double-integrator dynamics

$$\dot{\mathbf{r}}_i(t) = \mathbf{v}_i(t)$$
$$\dot{\mathbf{v}}_i(t) = \mathbf{b}_i\mathbf{u}_i(t)$$

with position $\mathbf{r}_i(t) = [x_i(t), y_i(t)]^T$ and velocity $\mathbf{v}_i(t) = [u_i(t), v_i(t)]^T$. The control system for the NCS functions as a secondary controller that acts on the primary controllers of the heterogeneous agents in the network. This secondary regulator controls the acceleration through inputs $\mathbf{b}_i\mathbf{u}_i(t)$. The state of each node $\mathbf{x}_i(t) = [\mathbf{r}_i(t), \mathbf{v}_i(t)]^T$ is then represented by the Linear Time-Invariant (LTI) state-space system

$$\dot{\mathbf{x}}_i = \mathbf{A}\mathbf{x}_i(t) + \mathbf{B}\mathbf{u}_i(t) \tag{2.1}$$

$$\mathbf{y}_i = \mathbf{C}\mathbf{x}_i(t) \tag{2.2}$$

with

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ b_{i,31} & 0 \\ 0 & b_{i,31} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

For a time-varying desired formation $\mathbf{h}(t) = [\mathbf{h}_1(t), \mathbf{h}_2(t), ..., \mathbf{h}_N(t)]^T$, Dong et al. propose the control law

$$\mathbf{u}_i = \mathbf{K}_1[\mathbf{x}_i - \mathbf{h}_i] + \mathbf{K}_2 \sum_{j=1}^{N} [w_{ij}(\mathbf{x}_j - \mathbf{h}_j) - (\mathbf{x}_i - \mathbf{h}_i)] + \dot{\mathbf{h}_{i,\mathbf{v}}}. \tag{2.3}$$

Note that control input for i is based on its proximity to neighboring nodes $j = 1, ..., N$ and the edge weight w_{ij} between them. The selection of feedback gain matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ is

discussed in [1], [17].

2.1.3 Controllability and Observability

A system is said to be *controllable* if it can be driven to a desired final state from any initial state in finite time with no constraints imposed on control [18]. For LTI systems with state-space representation like that described in Equations 2.1 and 2.2, the classical criteria for controllability depends on the rank of the controllability matrix, $\mathbf{M}_c$. A system with n states is said to be controllable if rank($\mathbf{M}_c$)= n, where

$$\mathbf{M}_c = [\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}, ..., \mathbf{A}^{n-1}\mathbf{B}]. \tag{2.4}$$

A system is said to be *observable* if the whole state of the system can be determined from the measured inputs and outputs [18]. A similar rank test is applied to the observability matrix of a LTI system, whereby the system is deemed observable if rank($\mathbf{M}_o$)= n, where

$$\mathbf{M}_o = [\mathbf{C}, \mathbf{CA}, \mathbf{CA}^2, ..., \mathbf{CA}^{n-1}]^T. \tag{2.5}$$

While the system described in Equations 2.1 and 2.2 is both controllable and observable, these are binary classifications and do not account for any varying degree of cost to controlling or observing the system in a desired manner.

To quantify these characteristics in a cost-related manner, the *controllability* and *observability Gramians* can used. The controllability Gramian at time t is the positive semidefinite matrix

$$W_c(t) = \int_0^t e^{A\tau} B B^T e^{A^T\tau} d\tau \in \mathbf{R}^{nxn}. \tag{2.6}$$

The infinite-horizon controllability Gramian, given by

$$W_c = \int_0^\infty e^{A\tau} B B^T e^{A^T\tau} d\tau \in \mathbf{R}^{nxn}, \tag{2.7}$$

is often used for ease of computaion [19]. Equation 2.7 can be solved using the Lyapunov equation

$$AW_c + W_c A^T + BB^T = 0. \tag{2.8}$$

Similarly, the infinite-horizon observability Gramian is given by

$$W_o = \int_0^\infty e^{A^T\tau} C^T C e^{A\tau} d\tau \in \mathbf{R}^{nxn}, \tag{2.9}$$

and can be solved using the Lyapunov equation

$$A^T W_o + W_o A + C^T C = 0. \tag{2.10}$$

In [19], Summers et al. use these Gramians and the property of submodularity (discussed in Section 2.3) to optimize networks on the basis of controllability and observability cost. While not presented in this thesis, utilizing these metrics to evaluate a NCS for controllability and observability will be important for the integration of DTN into the system.

2.2 Graph Theory

A NCS is often modeled using graph theory to represent the network as a collection of nodes and communications links. A general discussion of graph theory and its relevance to NCS applications is given in [9]. Previous work thoroughly detailed the concepts of adjacency matrices, degree matrices, and Laplacians, and how they have been specifically applied to this model [1]. These concepts are summarized below.

2.2.1 NCS Graph Model

This NCS is modeled as a graph $G = (V, E, w)$, where V is the set of vertices representing the nodes of the NCS with E edges representing the communication links between them, which have some weight w representing communication strength [20]. As detailed in [1], the graph used to describe the NCS is a simple weighted undirected graph. The graph is simple, meaning a maximum of one edge exists between each node pair, and is undirected, meaning information flows both ways between nodes. The weights associated with each edge represent the strength of communication between nodes. A sample graph is shown in Figure 2.1, where V is represented by the spatial position of various nodes (UxVs) and the communications links E are weighted by w, the strength of the communication between nodes. Note that in Figure 2.1, the edge weights are inversely proportional to the distance between the nodes. Further details on edge weight assignment is addressed in Section 2.4.2.

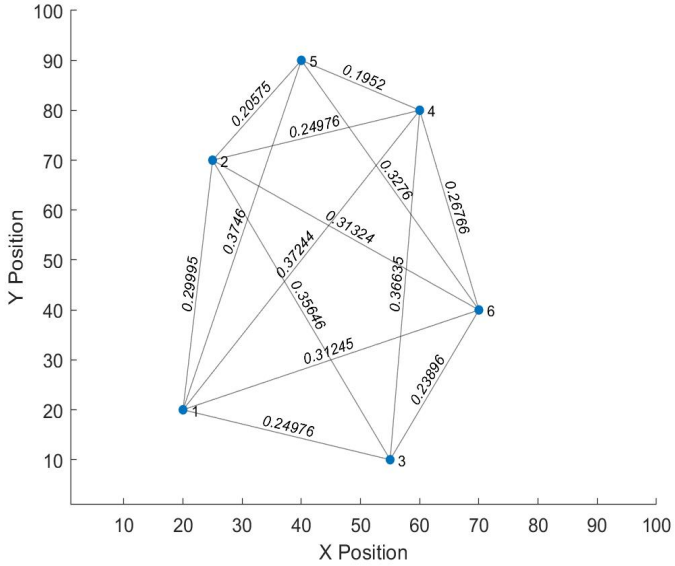


Figure 2.1. Example NCS represented as a graph.

2.2.2 Adjacency Matrix

The adjacency matrix $\mathbf{A}(G)$ is a useful way to represent the graph G as a matrix with entries w_{ij} representing the edge weights between vertices i, j. Since this NCS is modeled using an undirected graph, the adjacency matrix will always be symmetric, with zeros comprising the main diagonal. The adjacency matrix for the NCS depicted in Figure 2.1 is

$$\mathbf{A}(G) = \begin{bmatrix} 0 & 0.3000 & 0.2498 & 0.3724 & 0.3746 & 0.3124 \\ 0.3000 & 0 & 0.3565 & 0.2498 & 0.2058 & 0.3132 \\ 0.2498 & 0.3565 & 0 & 0.3663 & 0 & 0.2390 \\ 0.3724 & 0.2498 & 0.3663 & 0 & 0.1952 & 0.2677 \\ 0.3746 & 0.2058 & 0 & 0.1952 & 0 & 0.3276 \\ 0.3124 & 0.3132 & 0.2390 & 0.2677 & 0.3276 & 0 \end{bmatrix}$$

Note that in Figure 2.1 nodes 3 and 5 are disconnected, meaning the corresponding non-diagonal entries $\mathbf{A}_{3,5}(G)$ and $\mathbf{A}_{5,3}(G)$ in the adjacency matrix are 0.

2.2.3 Degree and Laplacian Matrices

One metric of connectivity used to describe a graph is the *degree* of each node, which for an undirected graph is simply the number of adjacent (connected) nodes [9]. The degrees of the vertices $\delta(v)$ then make up the entries in the degree matrix. Thus, the degree matrix for the graph depicted in in Figure 2.1 is

$$\Delta(G) = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

A third matrix sometimes used to describe a graph is the Laplacian, $L(G)$, which is simply the difference between the degree and adjacency matrices, $\Delta(G) - \mathbf{A}(G)$ [21]. The Laplacian matrix for the graph depicted in Figure 2.1 is

$$L(G) = \begin{bmatrix} 5.0 & -0.3 & -0.2498 & -0.3724 & -0.3746 & -0.3124 \\ -0.3 & 5.0 & -0.3565 & -0.2498 & -0.2058 & -0.3132 \\ -0.2498 & -0.3565 & 4.0 & -0.3663 & 0 & -0.239 \\ -0.3724 & -0.2498 & -0.3663 & 5.0 & -0.1952 & -0.2677 \\ -0.3746 & -0.2058 & 0 & -0.1952 & 4.0 & -0.3276 \\ -0.3124 & -0.3132 & -0.239 & -0.2677 & -0.3276 & 5.0 \end{bmatrix}$$

The Laplacian matrix has important implications for communications robustness metrics, and will be discussed further in Section 2.4.2.

2.3 Submodularity

Submodularity is a well-defined property of set functions that provides optimization within proven bounds. A thorough discussion of submodularity and its applications in optimization and function maximization can be found in [20], [22].

2.3.1 Submodular Set Functions

A *set function* $f : 2^V \rightarrow \mathbb{R}$ is a function that assigns a subset $S \subseteq V$ to some value $f(S)$ [20]. In this NCS model, which is composed of N nodes, the *ground set* V is the set of all possible locations where nodes may be positioned. If S is the set of N positions where the nodes are placed, then NCS with position set S is then assigned some utility $f(S)$. Figure 2.2 shows a map of SCI discretized into a gridded space. Each location j on the gridded map represents a discrete location and is a member of the ground set V from which a set S of size N is selected. For the MTX scenario, this portion of SCI covers an area that is 7.5 km square discretized into 100x100 m grid, with each location j occupying a 75 m square.

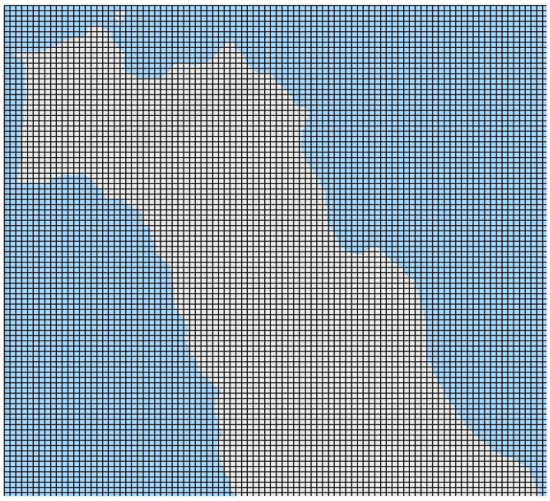


Figure 2.2. Map of SCI with ground set V (size 100 x 100) shown. V describes all possible locations for nodes to be placed.

The key property of certain set functions that leveraged in this thesis is called submodularity. The submodular property of set functions can be expressed in many ways, but is defined here as in [22]:

Definition 2.3.1. *Submodularity*: A set function $f : 2^V \rightarrow \mathbb{R}$ is submodular if for any A and B, with $A \subseteq B \subseteq V$, and any $s \notin B$,

$$f(A \cup s) - f(A) \geq f(B \cup s) - f(B) \tag{2.11}$$

In other words, adding a new node s to the smaller subset A results in a greater increase in the utility function than when adding it to the larger subset B. This is often referred to as the "diminishing returns" property of set functions. Another useful property of submodular functions is that submodularity is preserved for linear combinations of submodular functions [20]. For submodular set functions $f_1, \ldots, f_n : 2^V \rightarrow \mathbb{R}$ with nonnegative coefficients $\alpha_1, \ldots, \alpha_n$, the linear combination

$$f(S) = \sum_{i=1}^{n} \alpha_i f_i(S) \tag{2.12}$$

is submodular. This property is used to develop a more complex submodular function J to evaluate the total utility of the NCS.

2.3.2 Submodular Function Maximization

Submodular function maximization is a well-known area of research, with proven optimality bounds. The difficulty of selecting the optimal position set S from the set of all possible positions V given some constraints on S is identified by Krause and Golovin as non-deterministic polynomial-time (NP) hard [20], meaning that for this model the number of computations required grows exponentially with both the size of the NCS and the size of the gridspace it occupies [1].

The computational complexity of NP hard problems can be overcome by approximately solving them using a *greedy algorithm*, which was proven by Nemhauser et al. in 1978 [23]. Summarized in [20], the greedy algorithm starts with the empty set S_0 and maximizes the derivative of the utility function $\Delta_f(e|S_{i-1})$. In other words, the greedy algorithm finds the position that maximizes the contribution of new element e to the existing set S_{i-1} to create the new set:

$$S_i = S_{i-1} \cup \{\underset{e}{\operatorname{argmax}}\, \Delta_f(e|S_{i-1})\} \tag{2.13}$$

If S^* is the optimal set that maximizes the submodular utility function f, then iteratively selecting elements to fill the set S utilizing the greedy algorithm described in Equation 2.13 will result in an approximation not worse than $(1 - 1/e)$ of the optimal value [22]. Moreover,

if S^* is of optimal size k, then greedily selecting a set of size l yields an optimality bound of

$$f(S_l) \geq (1 - e^{-l/k}) f(S^*) \tag{2.14}$$

This important result, shown by Krause and Golovin in [20], guarantees that by selecting a monotone submodular utility function, each additional node in the NCS will cause the utility to asymptotically approach the optimal value as l increases with respect to k.

In addition to achieving solutions with known optimality bounds, this approach reduces the computational complexity from a NP hard problem to one that is solvable with polynomial time algorithms [22]. For this example, a NCS composed of N nodes located in a gridspace with p discrete positions would required p^N computations to achieve the optimal solution. Using the polynomial time algorithm, the approximate solution can be found after only pN computations.

2.3.3 Adaptive Submodularity

This thesis is primarily concerned with optimizing NCS topologies with respect to a submodular utility function (described in Section 2.4), but it may also be desirable to use feedback from the NCS to change how those topologies are generated. To do this, the machine learning technique known as adaptive submodularity can be used to optimize placement policies [24]. This adaptive approach is particularly appealing in a stochastic setting, where incomplete knowledge of the environment may result in changing mission priorities. In [24], Golovin and Krause present a framework for an adaptive stochastic optimization, along with the familiar optimality guarantees that accompany submodularity.

2.4 Utility Function

For a full discussion of the methodology and analysis used to select a submodular utility function to evaluate the NCS model, see [1]. The resulting utility function J is

$$J(S) = \alpha_s f_s(S) + \alpha_r f_r(S) \tag{2.15}$$

The utility function J is composed of a weighted sum of two submodular functions: one that evaluate the NCS for some sensing utility f_s and one that evaluates it for some commu-

nications robustness utility f_r.

2.4.1 Sensing Utility

The sensing utility is defined as the reward for positioning UxVs in high value locations. Matrix **F** is defined as the sensing potential field whose entries correspond to the sensing utility gained at each discretized location in space. In the MTX scenario, the road network, NSW team, and target objective all have high sensing values, so a node positioned closer to one of these would contribute to a higher sensing utility f_s than one positioned farther away. For a NCS with N nodes at position set S, the total sensing utility is given by the sum of the utility from each node i at position $j \in S$:

$$f_s(S) = \sum_{i=1}^{N} f_{s,i}(j) \tag{2.16}$$

For the set function $f_{s,i}(j) \geq 0$ for all i and j, adding additional nodes up to N will not cause $f_s(S)$ to decrease. This results in a sensing utility function $f_s(S)$ that is monotone submodular [1], [20].

2.4.2 Robustness Utility

In the MTX scenario, sensing utility alone is not sufficient to evaluate the worthiness of the NCS topology. Some weight must be given to measures of connectivity to ensure effective communication between different nodes in the network. Different methods to quantify robustness are evaluated in [1], but for this application it is defined based on the effective resistance of the graph. Described in [21], effective graph resistance models the graph as a circuit, with each edge modeled as a resistor with some resistance. The edge connecting nodes i and j is given a weight that is a function of the range between the nodes. For example, two nodes i and j separated by range r_{ij} are able to communicate until separated by the distance $r_{\max}$. The following inverse exponential edge weight is used to quantify the strength of communication between the nodes, with τ determining the rate of decay:

$$w_{ij}(r_{ij}) = \begin{cases} 1.0 - 0.9e^{\tau r_{ij}} & \text{if } r_{ij} \leq r_{\max} \\ 0 & \text{if } r_{ij} > r_{\max} \end{cases} \tag{2.17}$$

Thus, as the distance between nodes increases, the edge weight increases and the graph becomes more "resistant" to communication, or less robust. The effective graph resistance R can be determined from these edge weights by summing the effective resistance between all node pairs, which can be easily calculated using the eigenvalues λ_i from the Laplacian of the adjacency matrix [21]. The effective resistance of graph G that is composed of N nodes is given by

$$R_G = N \sum_{i=2}^{N} \frac{1}{\lambda_i} \tag{2.18}$$

Because it is desirable that the utility function be maximized when the graph is most robust (least resistant), the robustness utility function for graph of resistance R is defined as $1 - \text{norm}(R)$, where R is normalized between 0 and 1. For a NCS with N nodes at position set S, adding node i at location $j \in S$ results in a robustness utility of

$$f_r(S) = \sum_{i=1}^{N} (1 - \text{norm}(R_{i,j})) \tag{2.19}$$

Since $0 \leq R_{i,j} \leq 1$ for all i,j, then $(1 - \text{norm}(R_{i,j})) \geq 0$ for all i,j, resulting in a robustness utility function $f_r(S)$ that is monotone submodular [1], [20].

2.5 Centralized Submodular Optimization

The following is a summary of the centralized approach used in [1] to optimally position nodes in the NCS, which is used to evaluate the results of the distributed approach to submodular optimization developed in this thesis.

The NCS is composed of a set of nodes representing unique UxVs with different dynamic constraints. Each type of node has a set sensing and communications distance, as well as a maximum speed that determines how far it can travel at each time step. At discrete time steps, a greedy algorithm selects the location to place the next node in the network by maximizing that node's contribution to the submodular utility function described by Equation 2.15. This results in a NCS topology that achieves near-optimal utility in an effective manner by reducing the computational complexity from NP hard to polynomial-time.

This utility function describes the NCS's ability to effectively sense its environment while maintaining robust communications. These potentially competing priorities are balanced

using a weighted sum, with the weights selected by the user. For the remainder of this thesis, weights are selected such that the sensing utility is valued at twice that of the communications robustness utility. The reasoning for this is fully described in [1]. Because a NCS topology that is understandable to the user is desired, these weights were selected primarily because the resulting networks appear to have a satisfactory balance between sensing and communications robustness.

In the MTX example the NSW team acts as a VL around which the various assets position in order to maximize the utility function described in Section 2.4. Automatically repositioning the UxVs with respect to one or more friendly but non-cooperative VLs is one of the desired characteristics of this NCS. Neither the NSW team or the supporting U.S. Navy Destroyer (DDG) are positioned by the algorithm; they follow preplanned paths throughout the mission scenario. The various sensing, communications, and mobility constraints of each node type are summarized in Table 2.1

Table 2.1. MTX agent characteristics.

Agent	Speed (m/s)	Sensing Range (m)	Comms Range (m)
NSW Team	2	-	5000
DDG	5	-	5000
UUV	2.6	200	5000
USV	15	200	5000
UAV	35	500	5000

Areas of high sensing utility include the VL, the road system that the VL will travel along, and the mission objective in the Northwest corner of the map.

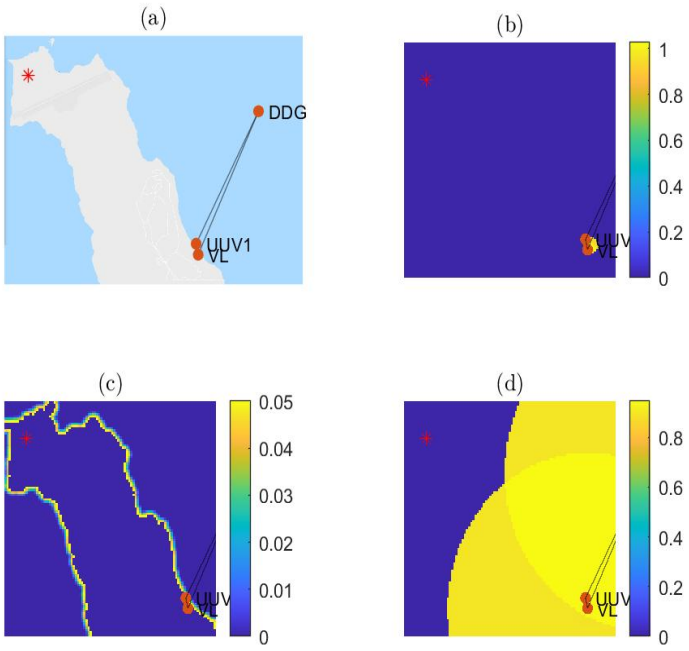


Figure 2.3. NCS, first node placed. Centralized method showing (a) the NCS topology, (b) Total Network Utility, (c) Sensing Utility, and (d) Communications Robustness Utility.

Figure 2.3 shows the results of the first step of the centralized placement algorithm. The manned assets (NSW team and DDG) are positioned according to their intended path of travel. The algorithm evaluates all possible locations to place the first node UUV1. Figure 2.3(c) shows the sensing utility based on the sensing potential field $\mathbf{F}$ for the seaborne assets, while 2.3(d) shows the communications robustness utility, which is highest between the DDG and the NSW team, acting as the VL. The weighted sum of those utilities is the total network utility, given in Equation 2.15, is shown in 2.3(b), with UUV1 positioned at the location of maximum utility. Note that there is only a very small area around the UUV with a high utility, shown in yellow. This is due to limiting the travel of each node from its previous position during a given time-step based on the constraints of the vehicle.

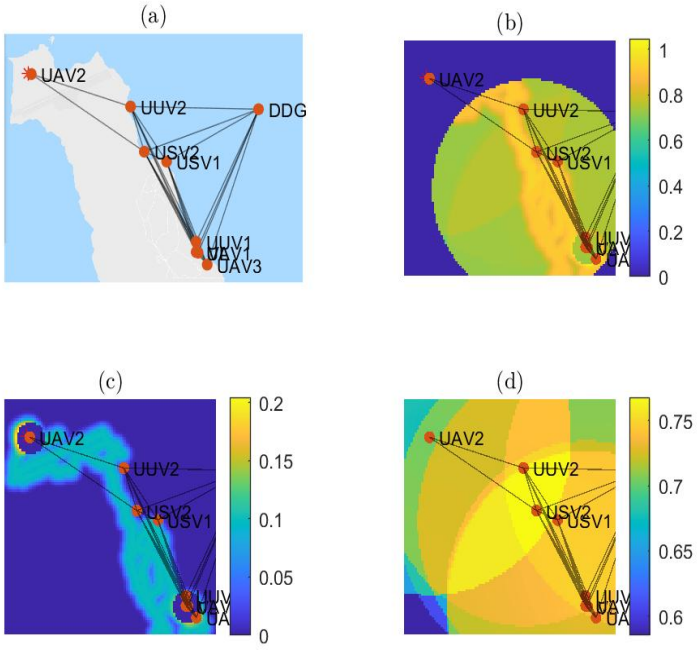


Figure 2.4. NCS, All nodes placed. Centralized method showing (a) the NCS topology, (b) Total Network Utility, (c) Sensing Utility, and (d) Communications Robustness Utility.

Subsequent nodes are placed in a similar fashion until the last node, UAV3 has been placed, as shown in the Figure 2.4. Note the much larger area of elevated utility, indicated by the much greater travelable distance of the UAV when compared with the UUV. The resultant near-optimal NCS topology is shown in Figure 2.5.

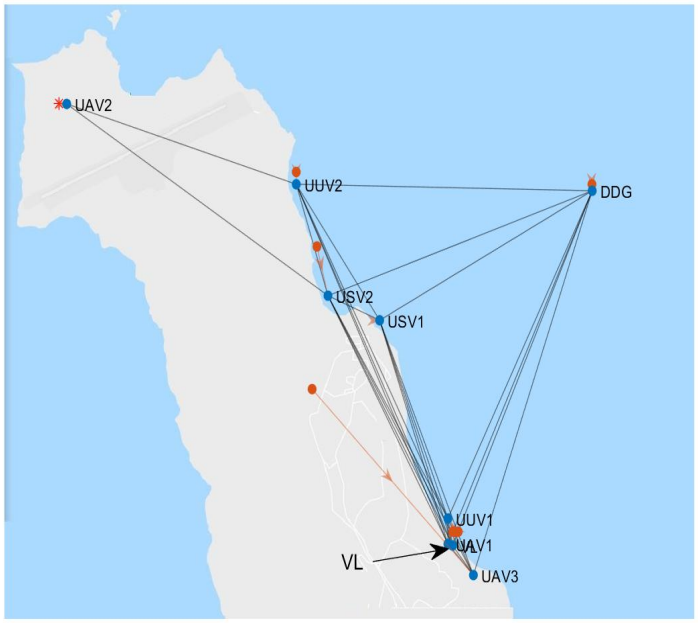


Figure 2.5. Sample results for NCS placed with the centralized method. VL is the NSW team. The red asterisk is the location of the mission objective. Orange arrows show node travel from previous position.

In the next chapter, different methods of implementing distributed control for multi-agent systems are examined, and the algorithm for NCS control using distributed submodularity is introduced.

CHAPTER 3:
Distributed Submodularity

The submodular optimization method described in Chapter 2 relies on a centralized controller to compute and plan the positions of all nodes in the NCS. In addition to the susceptibility of equipment failure, this is a vulnerability that could be exploited in an adversarial environment. Should the location or nature of the centralized controller be discovered by an adversary, it could then be targeted to disrupt or disable the NCS. In order to address this vulnerability, a framework that uses distributed submodular optimization to control the NCS is desired, where each node shares a computing burden while removing the vulnerable centralized planner. This creates a more robust NCS in which the loss of one or more nodes will not significantly degrade mission performance.

In this chapter, the applicability and limitations of the approach that most closely addresses the decentralized NCS control problem are explored. Our algorithm for distributed submodularity is introduced, and the results of several representative scenarios are analyzed by comparison with the centralized method.

3.1 The Distributed Online Greedy Algorithm

One method of distributed submodular maximization proposed in [7] relies on the use of an experts algorithm, which is summarize here.

Experts algorithms are techniques in which a user takes the advice of various "experts" and then assigns weights to the experts based on the results. This introduces prediction and learning into the optimization process [22]. A set of K experts $e_1, \ldots, e_K$ offers potentially contradictory advice to the user at each time step m. The user selects an expert and follows that experts advice, making choice $e[m]$. The user then receives some reward $\ell_m(e[m])$ and knowledge of the rewards $\ell_m(e_1), \ldots, \ell_m(e_K)$ that it would have received had it followed the advice of the other experts. Using this information, the user updates weights to adjust the probability of following each expert in a effort to maximize the reward $\sum_{m=1}^{M} \ell_m(e[m])$ [22].

One method for selecting these weights is known as an Exponentially Weighted Average

(EWA) algorithm, whereby each action $e_1, \ldots, e_K$ by experts $r = 1, \ldots, K$ at iteration m is given weight w_r^m [22]. This weight determines the probability of selecting the expert at time m, and is given by:

$$Pr(e[m] = e_r) = \frac{w_r^m}{\sum_{j=1}^{K} w_j^m} \tag{3.1}$$

At each step m, the rewards $\ell_m(e_r), r = 1, \ldots, K$ are calculated and the weights are updated exponentially, with the new weights given by

$$w_r^{m+1} = \beta^{\ell_m(e_r)} w_r^m \tag{3.2}$$

where $\beta \in (0, 1]$. β is a decay factor that determines how heavily to weight the new information against the old. The optimality bound for the EWA algorithm is proved in [22], and shown to be

$$\sum_{m=1}^{M} l_m(e[m]) + o(M) \geq max_{r=1,\ldots,K} \sum_{m=1}^{M} l_m(e_r) \tag{3.3}$$

Originally proposed by Golovin et al. in [7] and summarized in [22], a method for solving a distributed submodular maximization problem involves utilizing the experts algorithm above to select a set of nodes that satisfy the following:

1. Each element of the ground set $V = \{1, \ldots, n\}$ corresponds to one node.
2. Each node is capable of exchanging information with its one-hop neighbor, as well as more distant nodes via multi-hop routing protocols.
3. The nodes are assumed to have global time synchronization, which enables the nodes to execute sequential algorithm steps in unison.
4. The objective function $f_t : 2^V \rightarrow \mathbb{R}$ is assumed to nonnegative, monotone, submodular and time varying.
5. Each node is assumed to have oracle access to the functions f_t from previous time epochs $t = 1, 2, \ldots$, as well as the previous sets $S_1, S_2, \ldots$.

This Distributed Online Greedy (DOG) algorithm was originally developed for a distributed sensor network, like those used in environmental monitoring, where each node is a sensor that can be randomly activated [7]. Due to energy or computing constraints, a subset of the sensors are activated each time step, resulting in a reward based on the objective function

f_t. The value of the reward is then used to update the weight of each node using the EWA method described in Equation 3.2. This weight then effects the probability that the node is randomly selected for activation (Equation 3.1).

On the surface, the problem for which this algorithm was developed bears many similarities to the distributed NCS optimization problem in this thesis. However, there are key difference that makes this algorithm inapplicable in this case. The DOG method is useful for optimizing a set of *fixed* sensors, where some constraint limits the number than can be activated at one time. This thesis seeks a method that enables the utilization of all sensors, and then selects the optimal position for the nodes in the network. Additionally, the DOG algorithm optimizes a subset of nodes over the set of all nodes V, while this thesis seeks to optimize the subset of positions over the set of all possible positions V for the NCS to occupy.

3.2 Distributed Submodularity Method

As in the centralized method described in Chapter 2, the goal of the distributed approach is finding the optimal position set S from V possible discrete locations that satisfies

$$\max_{S \subseteq V} f(S) \tag{3.4}$$

where f is the submodular utility function described by Equation 2.15. Since the set S represents a vector of positions for each node in the NCS, $\mathbf{X}$ is used to represent the topology of the network. For a NCS of N nodes, $\mathbf{X}$ is a Nx2 matrix, with rows holding the position of nodes $i = 1, 2, ..., N$ in 2D space. Thus, the total utility of the NCS is actually represented by

$$J = \alpha_s f_s(\mathbf{X}) + \alpha_r f_r(\mathbf{X}) \tag{3.5}$$

From this point forward, this decentralized approach to this problem is referred to as the Distributed Submodularity (DS) method. Like the centralized method, DS relies on a connected network (no disconnected nodes). Additionally, each node must possess knowledge of the operating area, represented by the sensing potential field $\mathbf{F}$ described in Section 2.4.1. In order to compute the optimal topology for a distributed NCS, the method discussed in Section 2.5 is adapted to include a local search by each node and limited information sharing between nodes.

Described in Algorithm 1, the DS method proceeds with each node repeating a three-step process. Each node in the NCS keeps track of the vector $\mathbf{X}$ containing the positions of all nodes, as well as the current total utility J and the change in J from the last round. This process to reposition the NCS then proceeds in the following phases:

1. **Local Search:** (Lines 4-8) Each node concurrently and greedily evaluates a local area v for its optimal position. This is evaluated using the utility function described in Equation 3.5, assuming all other nodes in the NCS remain in place. The node then finds the location corresponding to the maximum increase in total utility that it can affect. This phase accounts for both the dynamic constraints on the vehicles, and any overlapping sensor coverage that would result from a potential move.
2. **Information Sharing:** (Lines 9-10) All nodes then broadcast two pieces of information: the maximum contribution to total NCS utility, and the location it would reposition to if its contribution is greater than all other nodes.
3. **Evaluation and Update** (Lines 11-19) The node which can affect the greatest change in total utility is designated and repositions itself, while all others update the position vector $\mathbf{X}$. Each node then calculates the new J and ΔJ. The process repeats until $\Delta J \leq 0$, indicating the near optimal topology has been reached.

Algorithm 1: Distributed Submodularity (DS) Method

Input: NCS size n, Current NCS topology $\mathbf{X}_o$, Sensing potential field $\mathbf{F}$, utility subfunction f_s and f_r, Local search area $v \subseteq \mathbf{V}$, Sensing and Communications Robustness utility weights α_s and α_r, NCS dynamics and communications constraints

Output: New NCS topology $\mathbf{X}_f$

$J = \alpha_s f_s(\mathbf{X}_o) + \alpha_r f_r(\mathbf{X}_o)$; // Initial NCS Utility

$\Delta J = J$; // Initialize

while $\Delta J > 0$ **do** // Each node $i = 1 : n$ performs concurrently

  for *each* $x \subseteq v$ *centered on* $X(i)$ **do** // from the perspective of node i

    $J_{NCS}(\mathbf{X}|X(i) = x)$ // NCS utility if node i positions at x

  end

  $[J_{max,i}, x_i] = max(J_{NCS})$; // Max utility, position for node i

  $\Delta J_i = J_{max,i} - J$;

  Broadcast $\Delta J_i, x_i$ to other nodes;

  Receive $\Delta J_j, x_j$ from all other nodes;

  Let k be the node that satisfies $max(\Delta J)$;

  if $i = k$ **then**

    Reposition, set $\mathbf{X}(i) = x_i$ // Update own position

  else

    Set $\mathbf{X}(k) = x_k$ // Update position of node that moved

  end

  $J_{new} = \alpha_s f_s(\mathbf{X}) + \alpha_r f_r(\mathbf{X})$; // Update NCS utility

  $\Delta J = J_{new} - J$;

  $J = J_{new}$;

end

$\mathbf{X}_f = \mathbf{X}$; // Final Optimal NCS Toplogy

There are some key differences between this distributed approach and the centralized method. The centralized method required the user to specify the order of node placement. In [1], qualitative analysis showed changing the order of node placement affected the resulting NCS topology, and therefore the overall utility of the network. The distributed

approach developed here is under no such constraint; any node may be placed in any order, as long as it will contribute the largest change in network utility. Additionally, this iterative process allows for nodes to reposition more than once per time step, a feature not possible with the centralized method. This allows for further refinement of the NCS, particularly in the communications robustness utility.

Section 2.4 describes the submodular utility function used in this model. While the same submodular utility subfunctions are used for the centralized and DS methods, it is helpful to describe why these functions can still be used in a distributed setting. Our distributed algorithm is iterative, while the centralized one is sequential in assigning nodes. The centralized method greedily adds nodes to the NCS until all are placed, and the submodular nature of the utility functions ensures that each node contributes to the overall NCS utility. The DS method starts with all nodes already in the NCS, and evaluates the network utility at their initial position (which is unlikely to be the near-optimal solution). Nodes are then iteratively positioned in a greedy manner, such that every move contributes to overall NCS utility, until no further utility-improving move can be found. While these methods differ in how nodes are placed, they both generate a NCS of N nodes by evaluated in the same submodular utility function.

The centralized method relies on a greedy algorithm that selects the location $j \in V$ to add to set S_{i-1} that maximizes the change in the submodular utility function f. This is repeated for nodes $i = 1, ..., N$ until the set is full.

$$S_i = S_{i-1} \cup \{\underset{j}{\operatorname{argmax}}\, \Delta_f(j|S_{i-1})\} \tag{3.6}$$

The optimality guarantees of this greedy approach are predicated on f being submodular [20]; if f is submodular, then

$$f(S_i) \geq f(S_{i-1}). \tag{3.7}$$

For the DS method, let S_N^k be the full position set of an NCS of size N at iteration k of the DS algorithm. Let

$$j_i^{k+1} = \{\underset{j}{\operatorname{argmax}}\, \Delta_f(j|S_N^k \setminus j_i^k)\} \tag{3.8}$$

be the new location for node i which maximizes f given that all other nodes in S_N^k remain

in place. If f is submodular, then

$$\Delta_f((S_N^k \setminus j_i^k) \cup j_i^{k+1}) \geq 0. \tag{3.9}$$

If the node whose new location contributes the most to f is

$$i^* = \{\underset{i}{\operatorname{argmax}}\, \Delta_f((S_N^k \setminus j_i^k) \cup j_i^{k+1}\}, \tag{3.10}$$

then the next iteration of S is given by

$$S_N^{k+1} = (S_N^k \setminus j_{i^*}^k) \cup j_{i^*}^{k+1}. \tag{3.11}$$

It then follows that

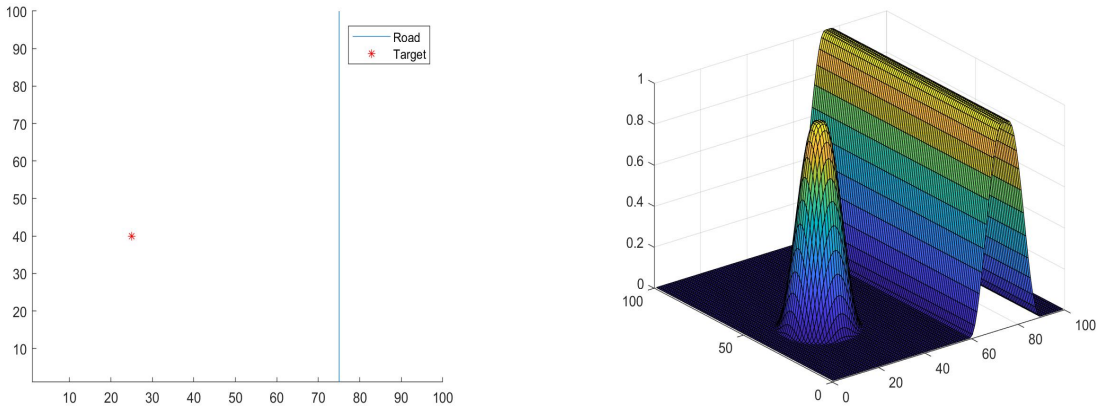
$$f(S_N^{k+1}) \geq f(S_N^k) \tag{3.12}$$

for a submodular function f.

While this algorithm is a fairly simple and straight-forward adaptation of the well-known simple greedy submodular maximization first proposed in [23] and further adapted in [20], we are not aware of this distributed form of the algorithm in the existing literature. A similar method using a Decentralized Monte-Carlo Tree Search (DCTS) is proposed by Corah in [4] that distributes planning for multiple robots over a fixed number of planning rounds. However, the fixed number of planning rounds removes advantages of iteration that the DS method adds, and the DCTS portion of the algorithm is much more computationally complex than needed for this deterministic scenario. In [3], Mokhtari et al. use a distributed greedy maximization to reach consensus on a global objective using decentralized nodes that only possess a portion of the global objective. This method combines locally sensed information with that gained by neighboring nodes to converge on a global solution. In future work where knowledge of the environment is incomplete, this method would likely prove useful for the MTX scenario. However, the focus of this thesis is developing a distributed submodular framework to optimize node-placement on a deterministic map of which all agents have complete knowledge. We propose the DS method as a simple, novel method to accomplish this task.

3.3 DS Example

In order to verify the performance of our distributed algorithm, a simple scenario is established. A five-node NCS composed of homogeneous assets is selected to provide coverage over a fixed area. Like the MTX scenario, this NCS uses the utility function described in Equation 3.5 to efficiently optimize the sensing and communications robustness of the NCS. In this simple example the sensing potential field **F** is based on the presence of a single road and target.



(a) Simple Map (b) Sensing Potentia **F**

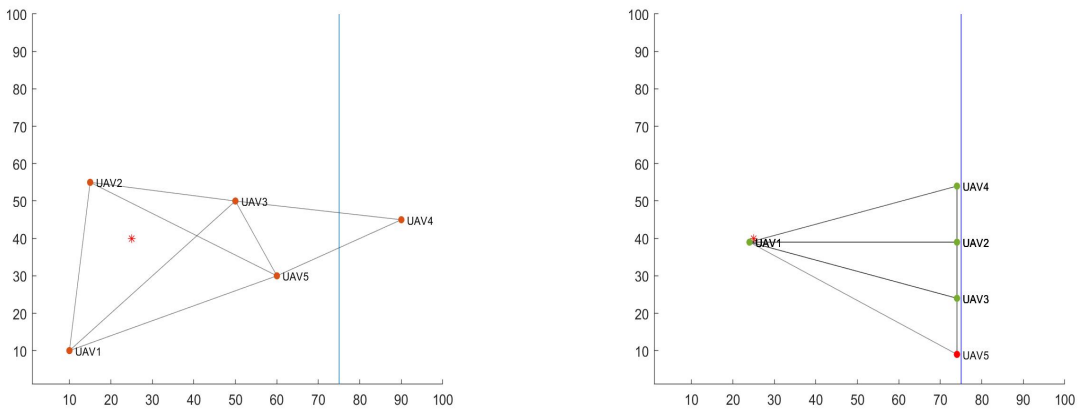
Figure 3.1. Simple NCS setup.

Here it is important to note that the overall value of the total NCS utility is somewhat arbitrary; it is a reward designed to achieve the desired network topology. Thus, the value is dependent on how the problem is developed. For the sensing potential **F** shown in Figure 3.1b, the peak sensing utility value is 1. The peak robustness utility for varies by network size, and occurs if all the nodes in the network were positioned in the same location (minimum distance between nodes). For a NCS of 5 nodes, using the communications model described in [1] and Section 2.4.2, this peak communications robustness value is 0.7493. Using the current method of weighting sensing utility twice as heavily as communications robustness, the theoretical peak utility for the scenario depicted in Figure 3.1 would then be (2)(1) + (1)(0.7493) = 2.7493.

There are several reasons why this utility isn't actually attainable and is given above only as

a reference. First, constraints are enforced on the nodes to prevent the sensor coverage from overlapping, meaning they cannot achieve peak robustness utility. Second, the distribution of areas of peak sensing potential in a given map area also means that peak robustness and peak sensing utilities cannot be achieved at the same time. The actual max utility for any individual scenario can only be found by solving Equation 3.4, which is itself NP hard. Thus for any given scenario the solution from the centralized method, with its known optimality guarantees, is the metric by which the performance of the distributed method will be evaluated.

To evaluate the DS method, nodes are positioned in some arbitrary configuration on the map. The centralized method described in Section 2.5 is used to position the nodes in the NCS. The results of the centralized method are shown in Figure 3.2. The resulting NCS topology is fairly intuitive, with nodes positioned near the areas of high sensing potential, spaced evenly along the road so that no overlap exists between the sensors.



(a) Arbitrary initial configuration

(b) Centralized method results

Figure 3.2. Simple NCS, centralized method.

Naturally, any arbitrary configuration will likely have a significantly lower utility than the near-optimal configuration generated by the centralized method. DS is then used to find the near optimal node placement for the NCS.

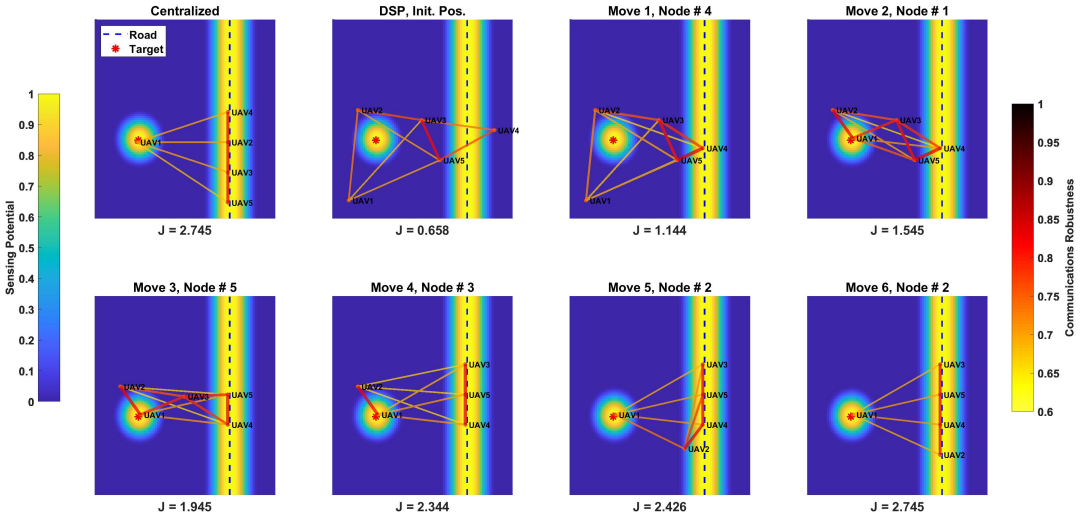


Figure 3.3. Simple NCS, DS results. Note: in NCS topology plots, DSP indicates DS Position (DSP)

The upper left plot in Figure 3.3 depicts the NCS topology generated by the centralized method, and the accompanying network utility. The subsequent plots display the progression of the DS method as it iteratively moves nodes from the initial configuration to the near-optimal NCS topology, shown in the bottom right plot.

The final configuration, reached in only six moves, is remarkably similar to the results from the centralized method. This demonstrates that our distributed algorithm can achieve results on par with the centralized method, using only a local search by each node and limited information sharing between the nodes. Note that the NCS topologies generated by the centralized and distributed methods are slightly different. Depending on the configuration of the underlying utility map, there may be multiple near-optimal solutions. Though both methods start from the same initial configuration, they ultimately arrive at a different NCS topology but with the same total utility (Figure 3.4).

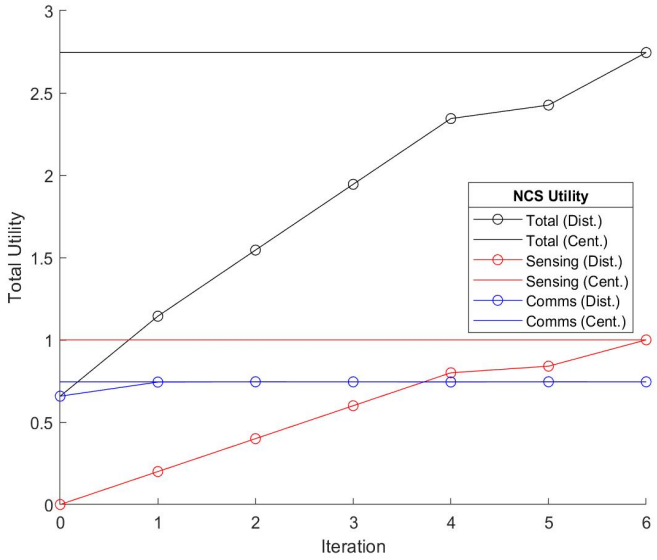


Figure 3.4. Simple NCS, utility plots.

Several factors are now analyzed to determine their effects on the performance of the distributed method.

3.3.1 Initial Position

When varying the initial configuration, the distributed algorithm is still able to achieve utility and topology similar to that of the centralized method. The number of individual moves will vary, as will the final position of each node, but the overall shape and utility of the NCS remains the same.

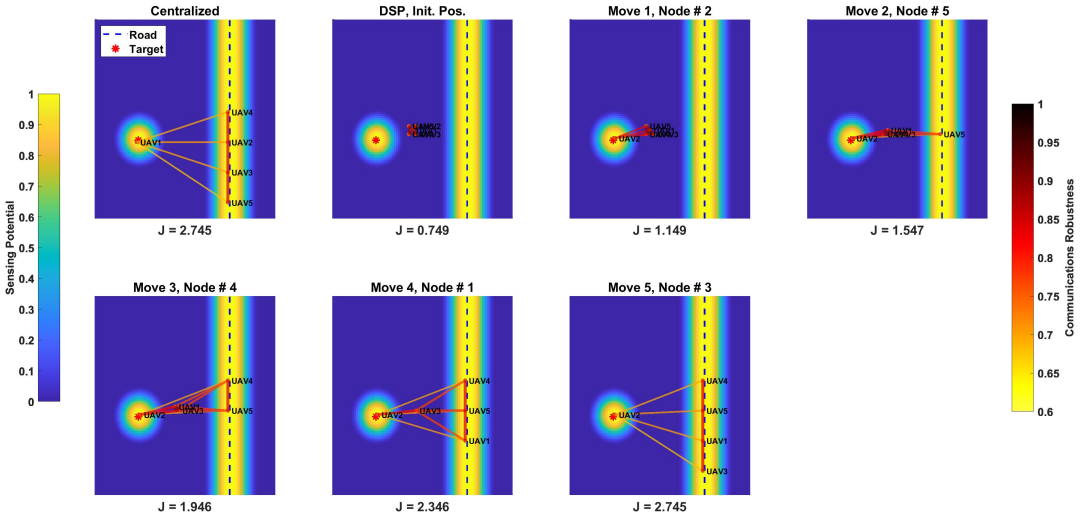


Figure 3.5. Simple NCS, DS results, initial configuration 2.

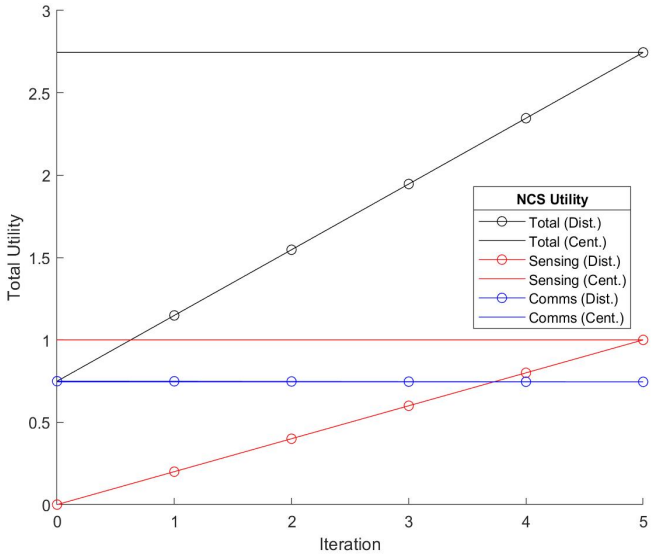


Figure 3.6. Simple NCS, utility plots, initial configuration 2.

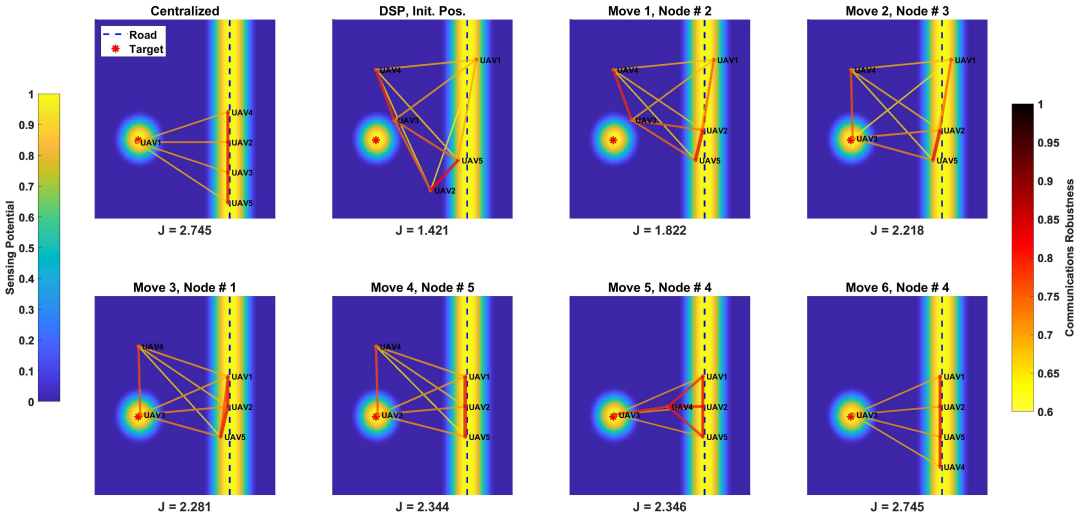


Figure 3.7. Simple NCS, DS results, initial configuration 3.

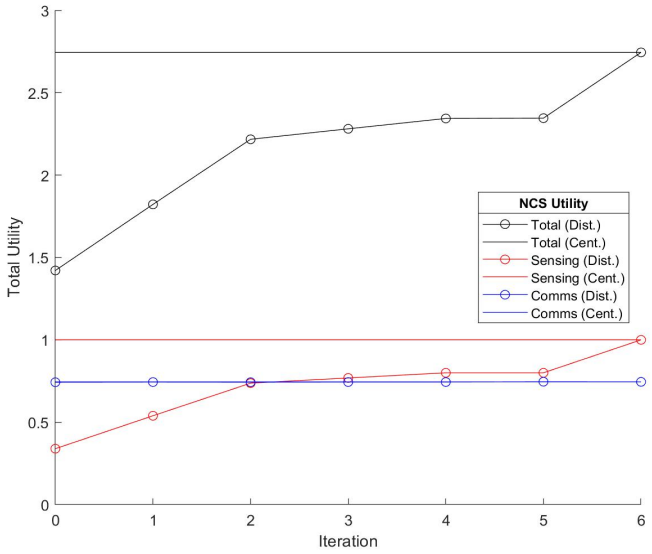


Figure 3.8. Simple NCS, utility plots, initial configuration 3.

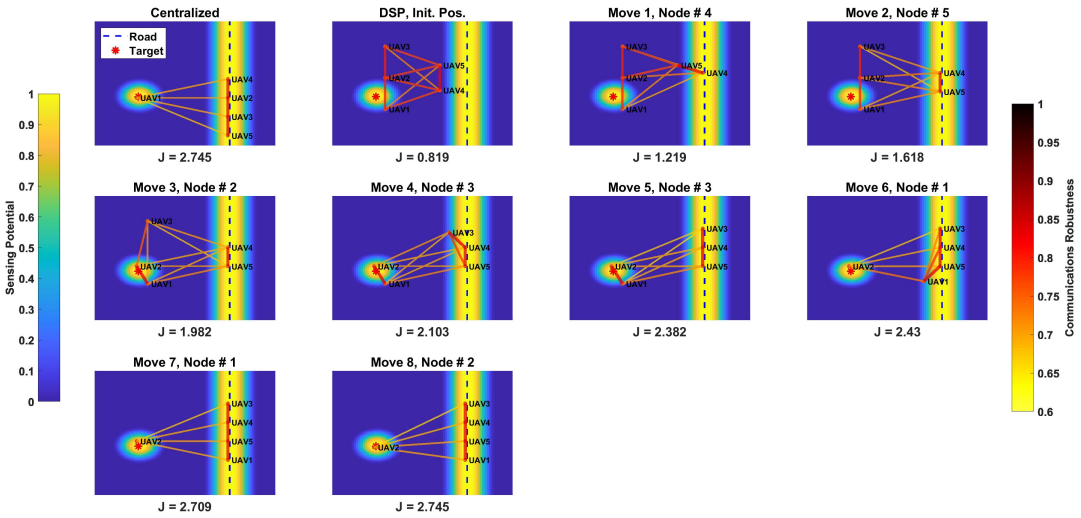


Figure 3.9. Simple NCS, DS results, initial configuration 4.

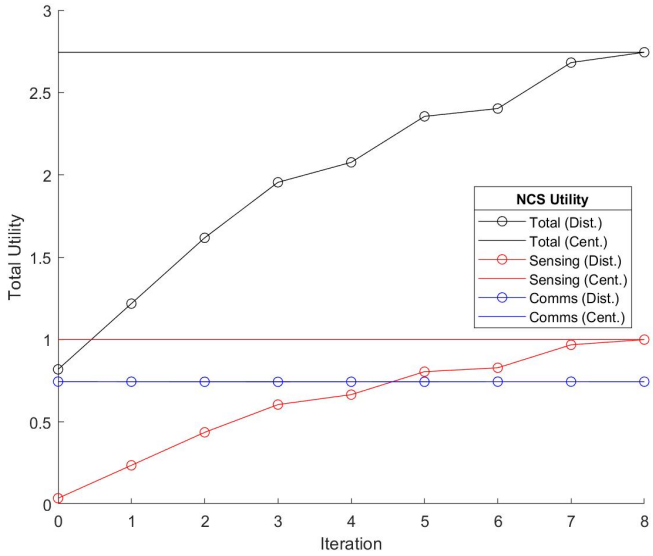


Figure 3.10. Simple NCS, utility plots, initial configuration 4.

3.3.2 Local Search Area

One advantage of this distributed method is that it has the potential to achieve the same results as the centralized method while allowing each node to only search a fraction of

the map area. This reduces the computation required by the nodes at each iteration of the algorithm. As an example, the scenario depicted in Figure 3.5 is adjusted by varying the local search area v of each node.

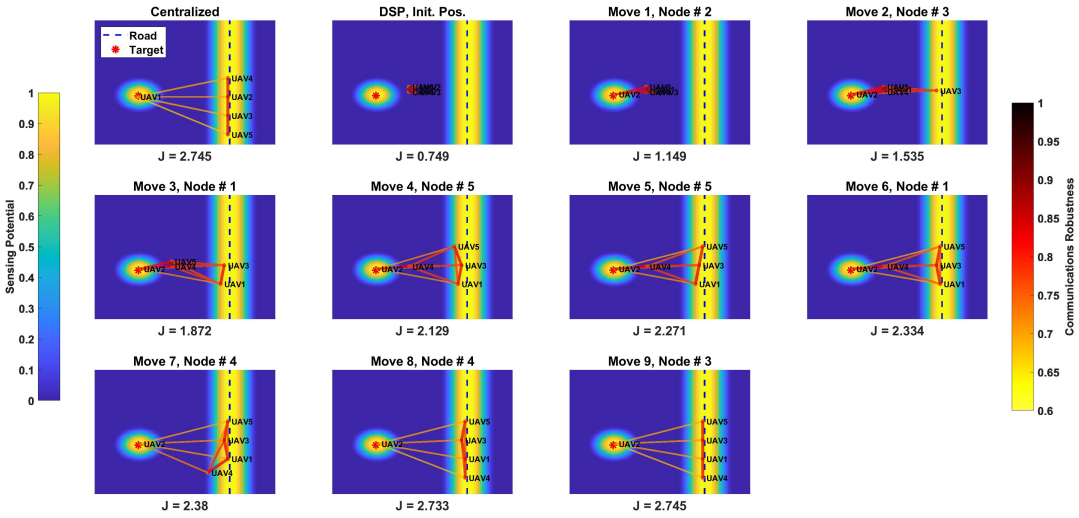


Figure 3.11. Simple NCS, local search area 25%.

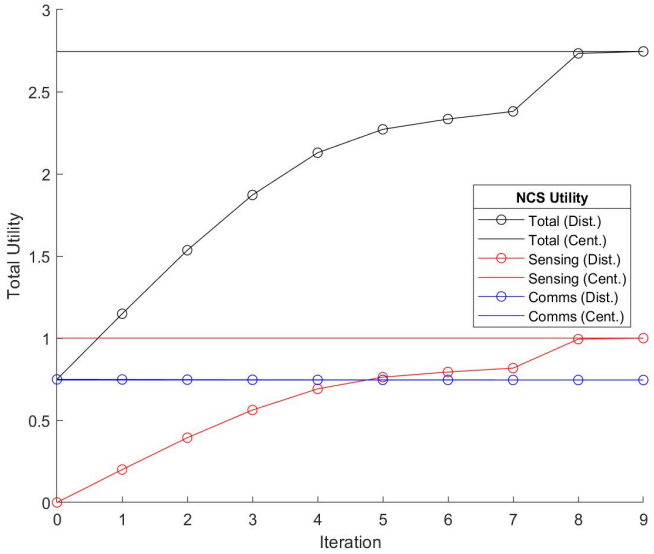


Figure 3.12. Simple NCS, utility plots, local search area 25%.

Figures 3.11 and 3.12 depict the results when the distributed method only searches a quarter of the map. Because the search is still conducted at the same discretized resolution, this requires significantly fewer computations at each iteration than when searching over the entire map area. However, this does result in more iterations, with nine moves required to achieve the optimal result.

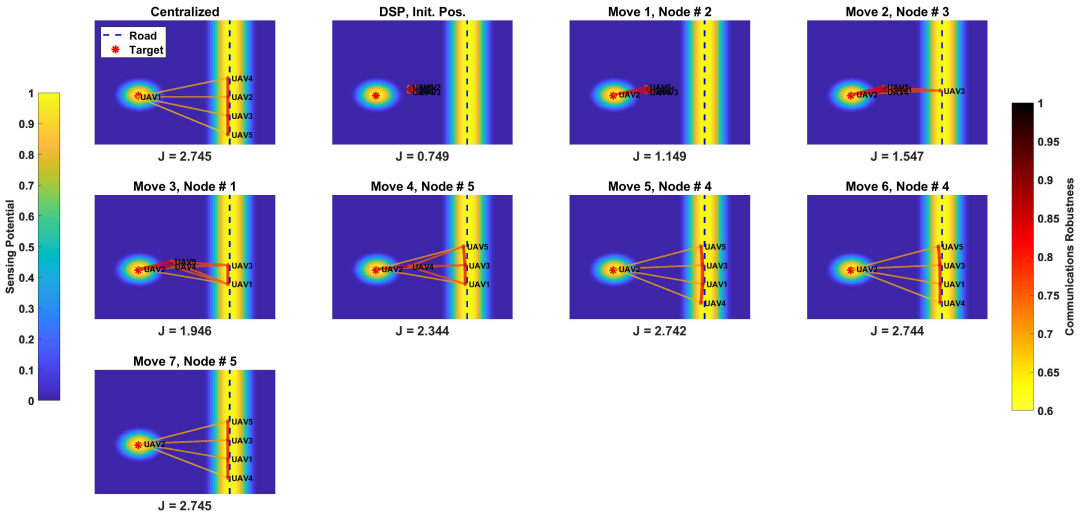


Figure 3.13. Simple NCS, local search area 37.5%.

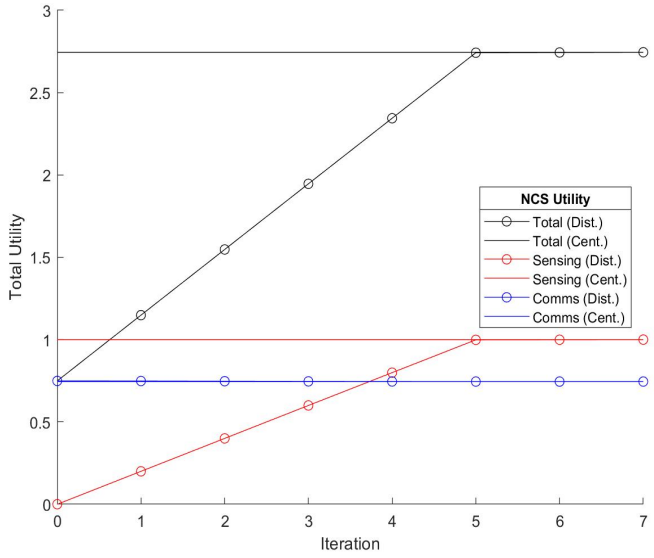


Figure 3.14. Simple NCS, utility plots, local search area 37.5%.

Figures 3.13 and 3.14 depict the results when 37.5% of the map is searched. This slight increase in computational complexity is rewarded with finding the optimal topology in seven moves, two fewer than the previous example.

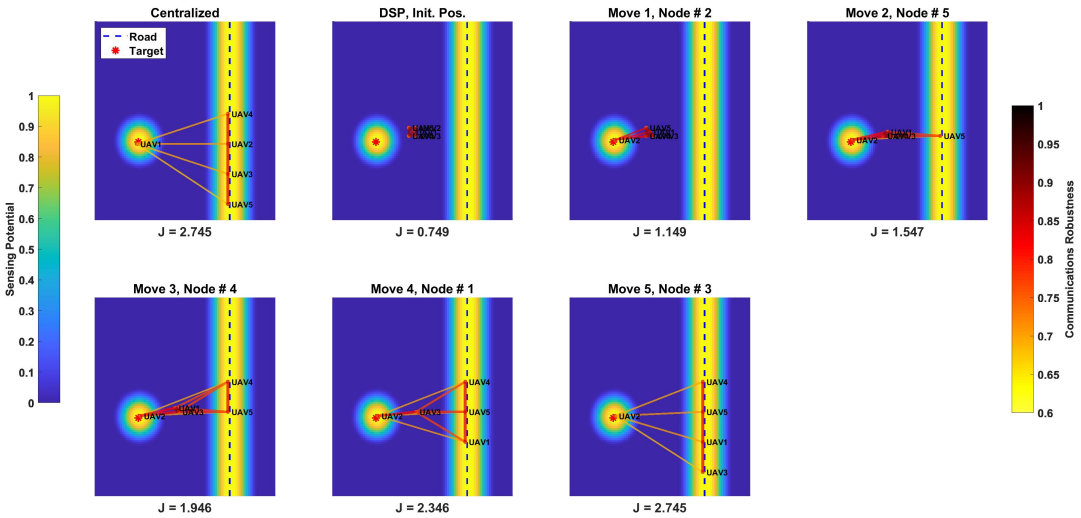


Figure 3.15. Simple NCS, local search area 50%.

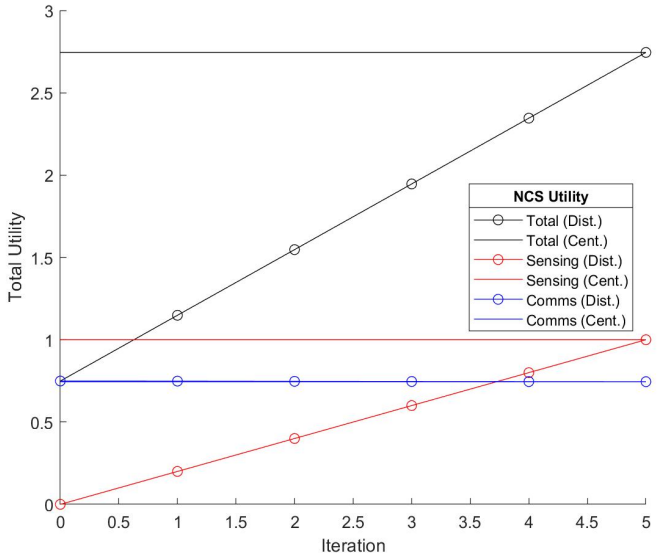


Figure 3.16. Simple NCS, utility plots, local search area 50%.

Figures 3.15 and 3.16 depict the results when 50% of the map is searched. This result represents the fewest moves possible for the distributed algorithm to solve the problem, since each node executes only one move to reach the solution. Because the distributed method spreads the computing between nodes, it actually results in a faster time to solution when compared with the centralized algorithm. In this specific case, by only searching half the map and finding the solution in five moves, the computational time required is reduced by a factor of two.

3.3.3 Communications Emphasis

It is clear from the results presented so far that the sensing utility subfunction is often the driving factor in determining the optimal topology. Next, the sensing potential field **F** from Figure 3.1b is altered to reduce the emphasis on the sensing. Shown in Figure 3.17, this is accomplished by flattening and reducing the peak magnitude of **F**.

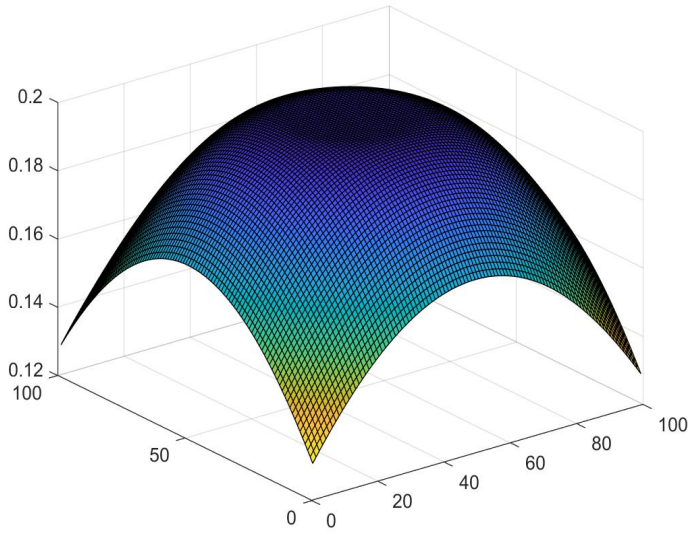


Figure 3.17. Simple $\mathbf{F}$ with decreased emphasis on sensing.

Intuitively, both the centralized and distributed methods result in a NCS topology that is tightly clustered to maximize communications robustness. Note that the distributed method rapidly converges to very near the centralized solution, then spends several rounds making fine adjustments to reach the optimal value.

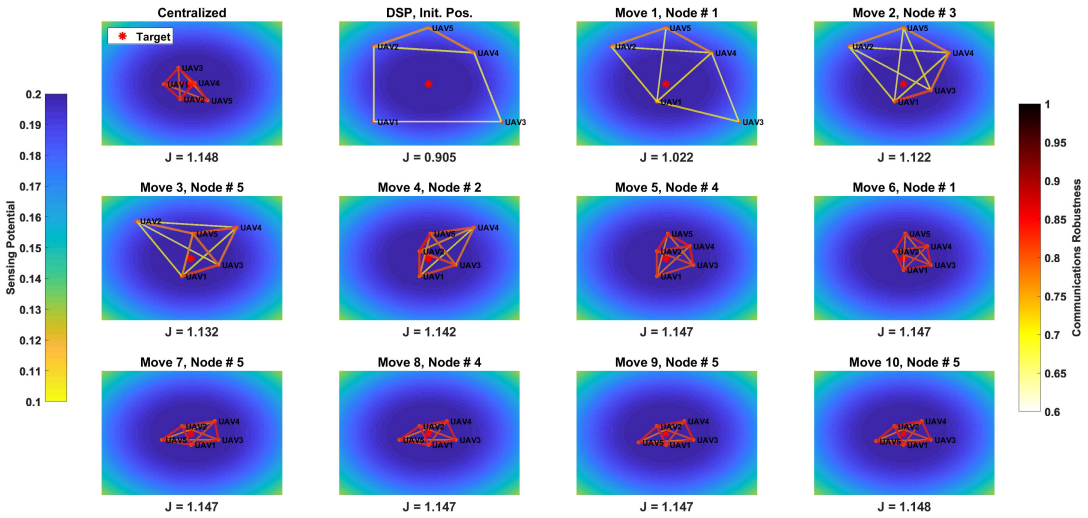


Figure 3.18. Simple NCS with decreased emphasis on sensing.

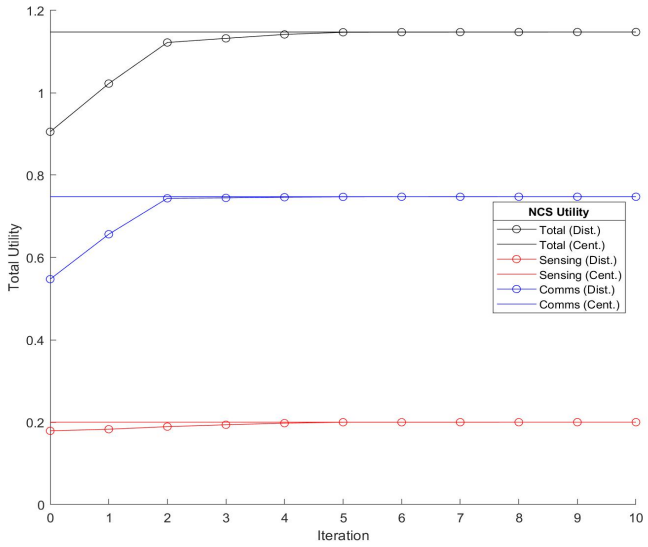


Figure 3.19. Simple NCS with decreased emphasis on sensing, utility plots.

The case presented in Figure 3.18 is one in which the two utility subfunctions are not really competing. Both the peak sensing and peak robustness values dictate a tightly clustered NCS near the center of the map. In order to create a more realistic problem, a scenario is designed in which communications robustness and sensing utility are directly opposed.

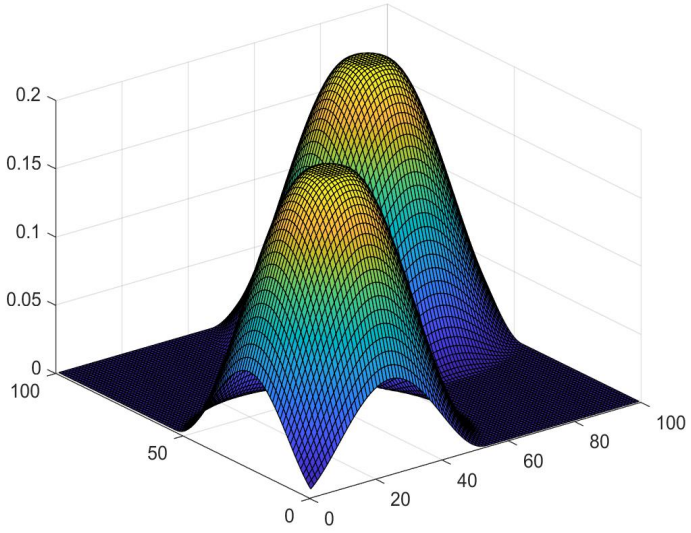


Figure 3.20. $\mathbf{F}$ with competing subfunctions.

In the scenario depicted in Figure 3.20, the areas of maximum sensing utility are located roughly the same distance apart as the maximum communication distance between the nodes.

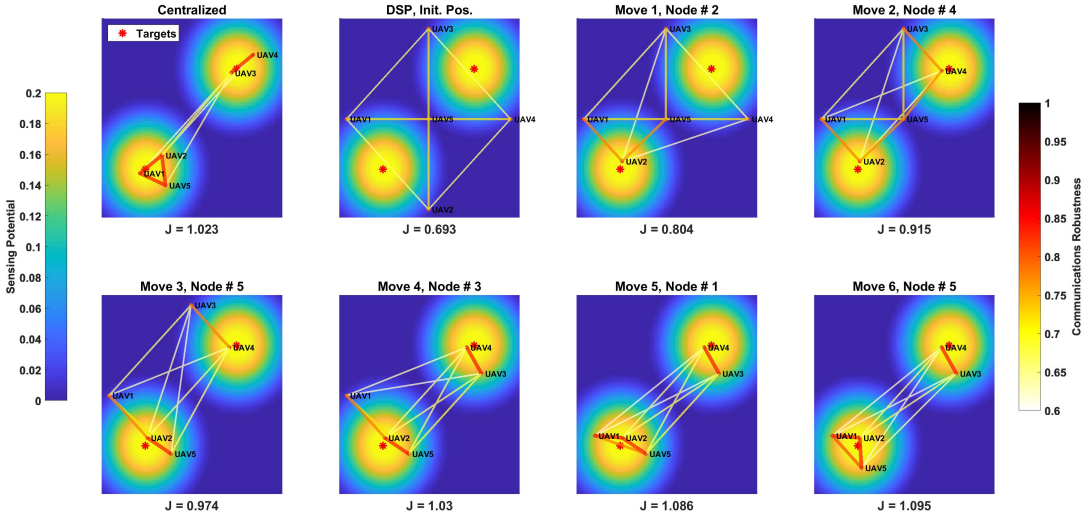


Figure 3.21. NCS with competing subfunctions.

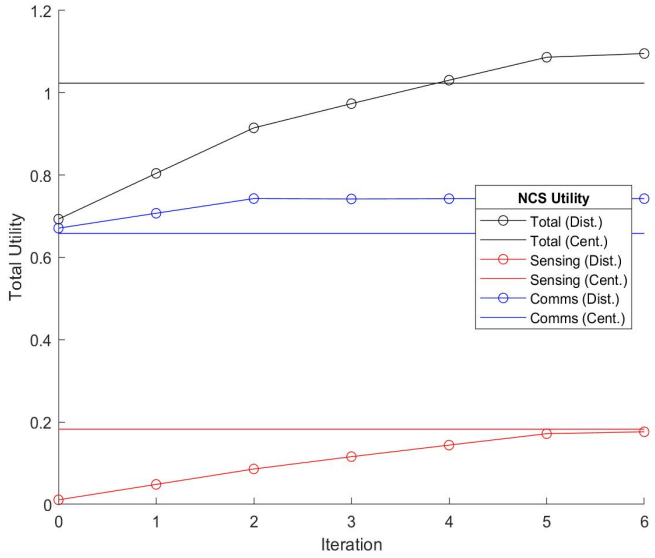
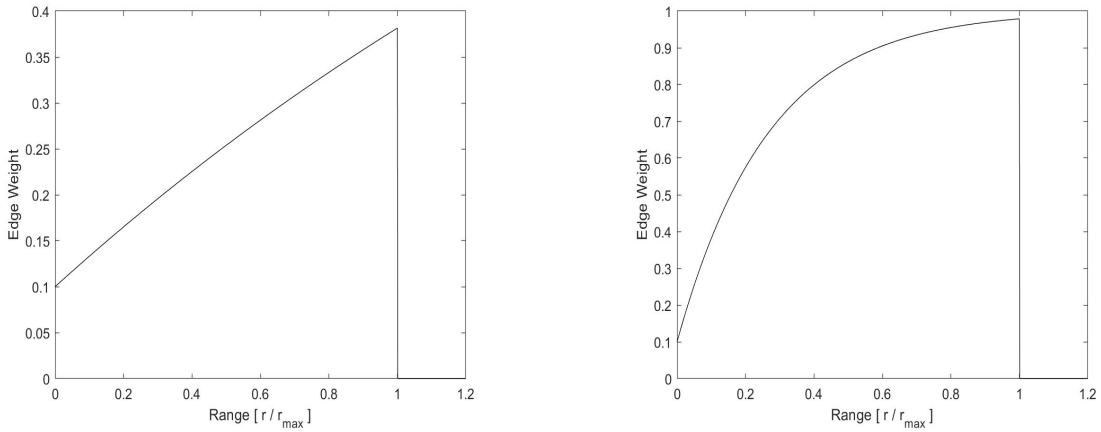


Figure 3.22. NCS with competing subfunctions, utility plots.

Surprisingly, in this case DS actually outperforms the centralized method; the resulting NCS topology has a higher total utility. Since [20] presents lower bounds on optimality, the centralized method topology is is not necessarily the optimal solution. Most of the examples presented in this chapter are relatively simple, where the subfunctions are largely complementary. In more complex scenarios with competing subfunctions, like those presented above and later in the full MTX scenario, DS may actually outperform the centralized method.

In the majority of examples examined in this chapter, there was been very little change in the robustness utility. This is largely due to the communications model used in this scenario. As described in Section 2.4.2, the communications model is based on an inverse exponential edge weight (Equation 2.17). Additionally, distributed method performance is analyzed in scenarios when the NCS experiences more drastic changes in communications robustness. This is accomplished by adjusting the rate of decay τ to increase the penalty to towards communications robustness as a function of distance.



Communications Model, $\tau = 0.005$ Communications Model, $\tau = 0.05$

Figure 3.23. Communications models with gradual and rapid decay. Communications robustness is inversely proportional to edge weight.

Figure 3.24 shows the performance of the distributed method when $\tau = 0.05$, a factor of ten greater than that used in the other cases. DS performs similarly for both cases. Note in Figure 3.25 that communications robustness contributed a large portion of the increase in total NCS utility. This is in contrast to the earlier cases where it was relatively constant. Again, in this more complex scenario with competing objectives, DS outperforms the centralized method.

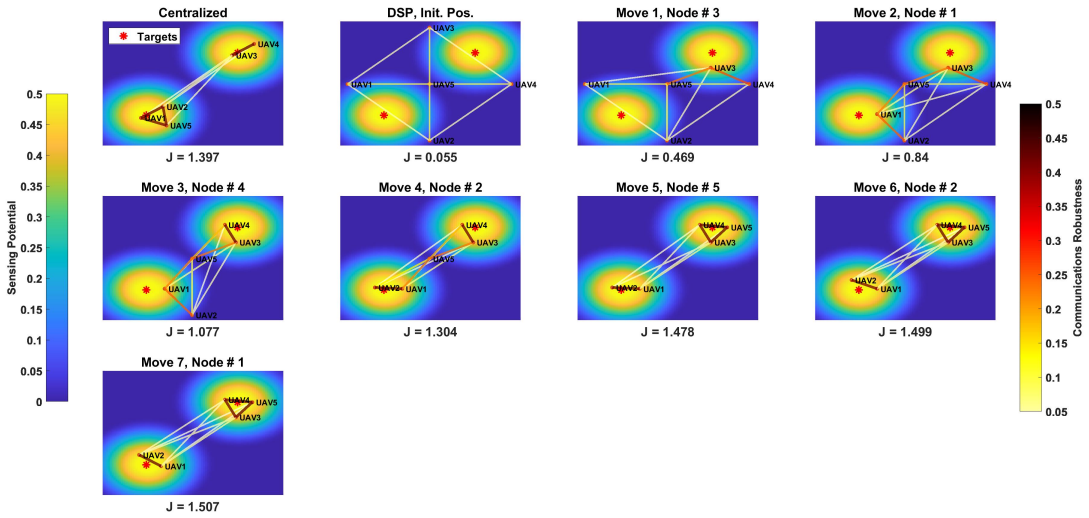


Figure 3.24. NCS with greater variance in communications robustness.

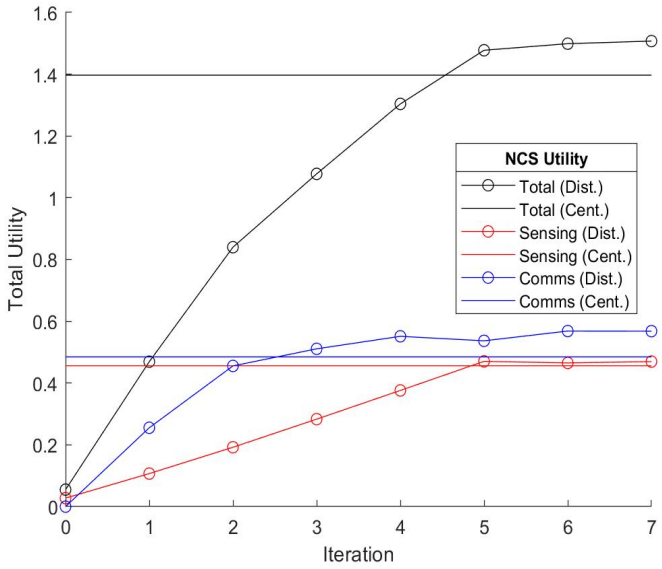


Figure 3.25. NCS with greater variance in communications robustness, utility plots.

3.4 Summary

In this section, our method for distributed submodular control, the DS algorithm, is introduced. We have shown that it is capable of achieving the same results as the centralized method under a range of different initial configurations and underlying utility maps. Additionally, by reducing the area searched by each node, in some scenarios DS has the potential to be more efficient than the centralized method. Lastly, we have posited that in more complex scenarios, DS can outperform the centralized method. This is explored further in the next chapter, where DS is introduced to the full MTX scenario.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results

In this chapter the full MTX scenario is presented, including the mission phases and the sensing potential fields **F** used for node placement. The DS method is applied to three different phases of the mission, and the results are compared with the centralized approach.

4.1 MTX Scenario

In order to test our DS algorithm on a more realistic and complex scenario, it is applied to the MTX problem discussed in Chapter 1. A NSW team supported by a network of heterogeneous UxVs will land on SCI and move toward an objective near the northwest corner of the island. The mission proceeds in six phases: Intelligence Preparation of the Battlefield (IPB), insertion, infiltration, actions at the objective, exfiltration, and extraction.

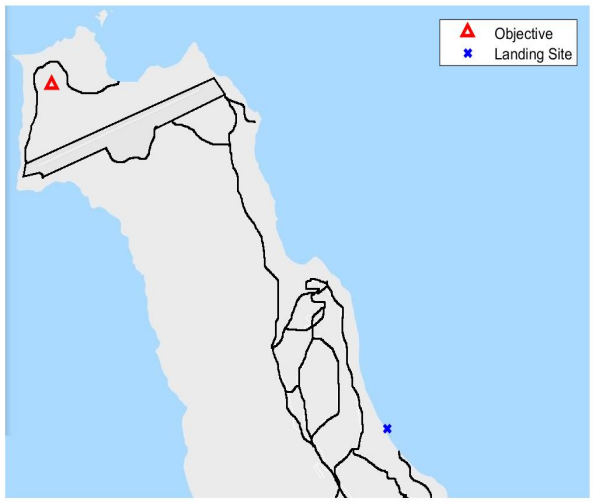
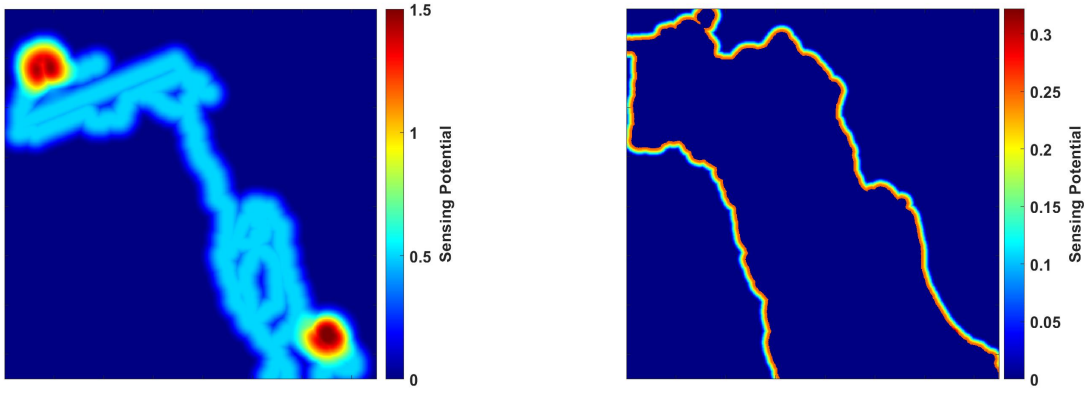


Figure 4.1. Location of objective and landing site on SCI. Roads shown in black.

Table 4.1. MTX mission phases.

Mission Phase	Description
IPB	UxVs conduct ISR over the battlespace, with the DDG further offshore providing Command and Control (C2). UUVs, acting as the VLs, conduct planned-path surveillance and reconnaissance to help determine insertion and extraction points.
Insertion	DDG moves closer to shore, USV with NSW team embarked breaks off from DDG and becomes the VL, maneuvers towards the landing area.
Infiltration	NSW team disembarks the USV and becomes the VL. NSW team travels along SCI road network toward objective, with UAVs and USVs providing sensor and communications coverage. UUVs break off from the NCS.
Actions at the Objective	NSW team acts upon the objective while NCS continues to provide sensor and communications coverage.
Exfiltration	NSW team travels toward extraction point, NCS continues to provide sensor and communications coverage. A USV breaks off and positions at the extraction point.
Extraction	NSW team embarks on a USV, which becomes the VL and returns to the DDG. All other assets leave the area for recovery.

During all phases of the mission, the asset acting as the VL moves according to a preplanned path, and is not positioned by the algorithm. To ensure coverage overhead the VL, an area of high sensing potential is applied to its position and moves with it throughout the mission. In order to ensure sensor coverage at the objective, that location has the highest sensing potential applied until after the Actions at the Objective phase, when coverage is no longer needed. Since a response from opposing forces will most likely travel along the island roads, sensor coverage of the road network is also important. Lastly, since sea control is assumed, seaborne assets only contribute to the sensing utility of the NCS along a thin band around the island's perimeter. An example of the sensing potential fields at the beginning of the infiltration phase are shown in Figure 4.2. Note the area of high sensing potential in the southeast corner surrounding the location of the NSW team.



(a) **F** for UAVs (b) **F** for USVs and UUVs

Figure 4.2. Sensing potential field for aerial and sea-based assets at the moment of insertion.

4.2 MTX Results

In this section the results from the first three phases of the MTX mission are presented: IPB, insertion, and infiltration. Actions at the objective is a stationary phase, and while the exfiltration and extraction would likely occur in different areas, the general progression of those phases is similar to infiltration and insertion. For these reasons, the focus of the analysis is on the first three phases.

4.2.1 Intelligence Preparation of the Battlefield

The IPB mission phase progresses as described in Table 4.1, with the UUVs surveying the potential insertion and extraction points while the USV provides additional ISR as well as acoustic communication with the UUVs. The UAVs provide overhead surveillance and communications. To induce the UAVs to survey the breadth of the island, time-varying objectives are introduced along the projected route. The NCS topologies generated by both the centralized and DS methods are shown in Figures 4.3 and 4.4 for the beginning and end of the IPB phase.

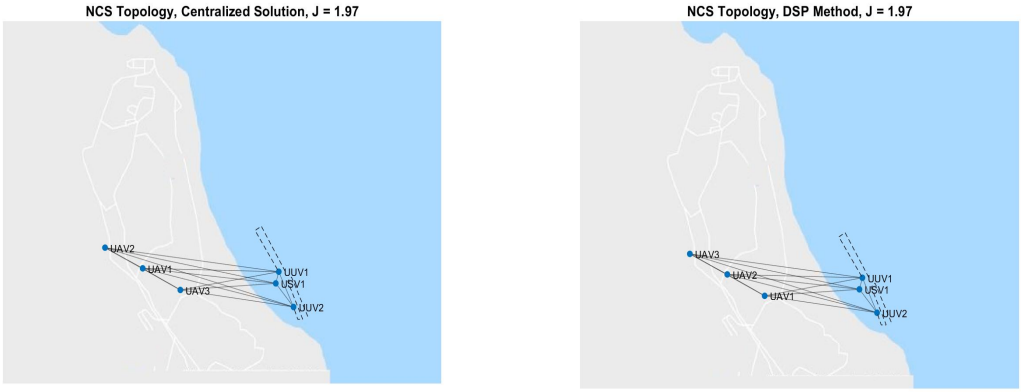


Figure 4.3. NCS topology for centralized and DS methods, beginning of the IPB phase.

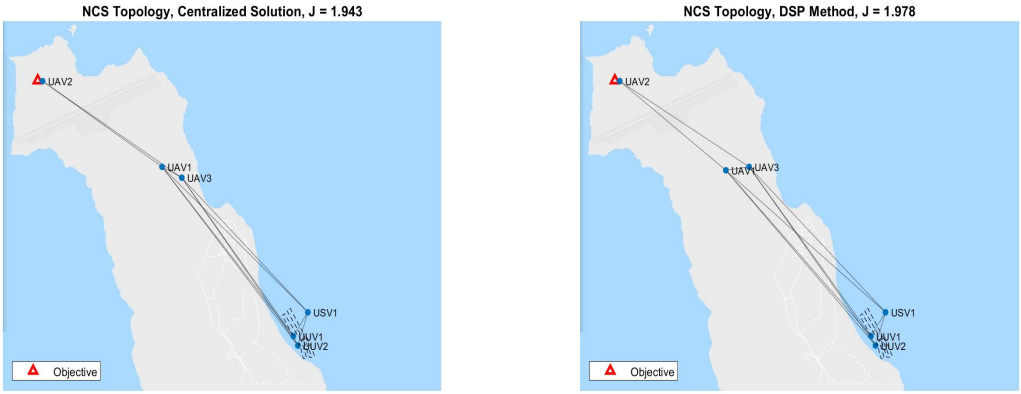


Figure 4.4. NCS topology for centralized and DS methods, end of the IPB phase.

Figure 4.5 illustrates the performance of the two methods over this phase of the mission. Throughout most of the IPB phase, the centralized and DS method provide similar NCS topologies and utilities. Note that while DS generally performs as well as or better than the centralized method, there are times during this phase when it does not. The performance of

the two methods when measured against one another is explored in more detail in the next chapter.

Note that Figure 4.5 and the other utility plots in this chapter show the network utility over the entire mission, while the previous chapter contained utility plots depicting a discrete moment in time. Thus, it is not surprising that the total NCS utility rises and falls throughout the mission phase as the underlying sensing objective map changes.

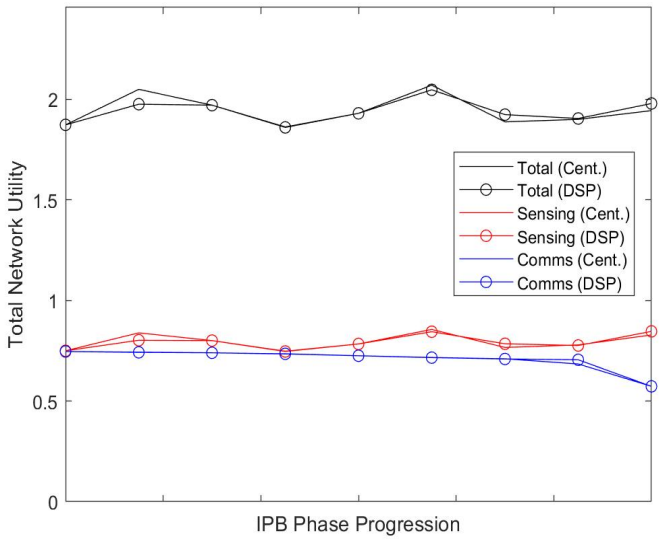


Figure 4.5. Total NCS utility throughout the IPB phase.

4.2.2 Insertion

During the insertion phase, the NSW team embarked aboard one of the USVs is transported to the landing beach while the DDG moves into a holding pattern offshore. Initially, the NCS positions assets to provide coverage over the target and portions of the road network. As USV2 gets closer to shore, some UxVs move to provide more coverage near the NSW team. The NCS topologies during the beginning, middle and end of the insertion phase are shown in Figures 4.6, 4.7, and 4.8, respectively. The dotted lines show the route of the manned assets as they enter the map area.

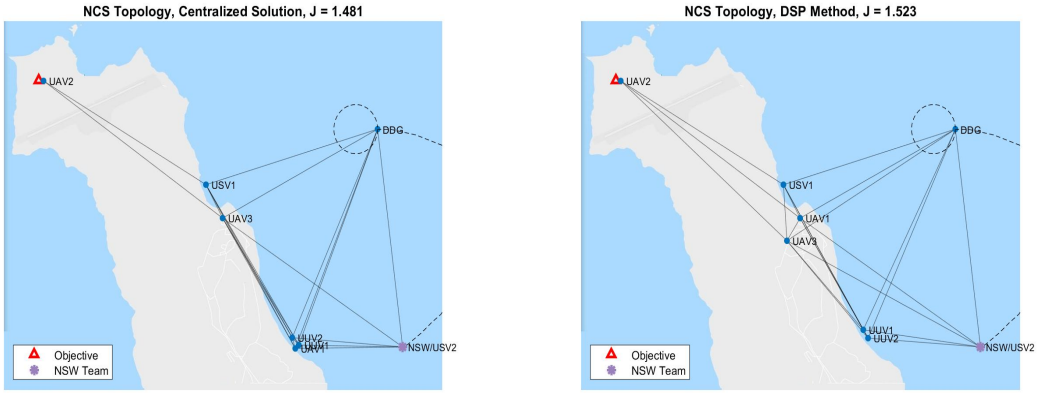


Figure 4.6. NCS topology for centralized and DS methods, beginning of the insertion phase.

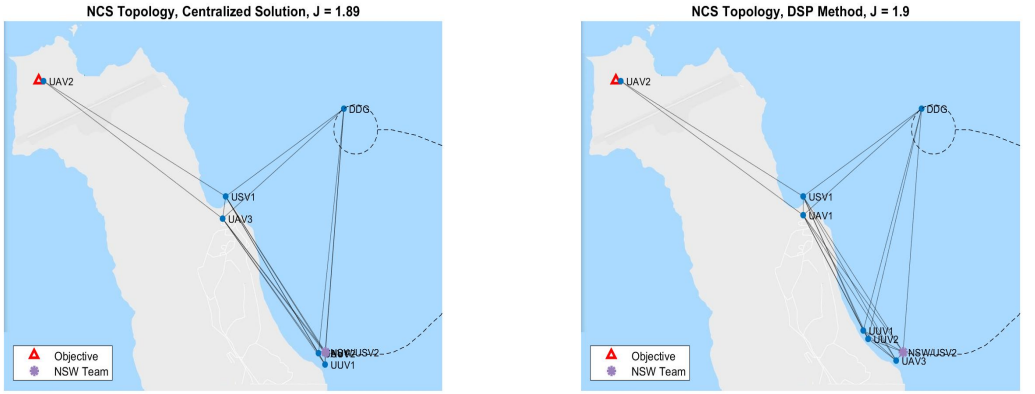


Figure 4.7. NCS topology for centralized and DS methods, middle of the insertion phase.

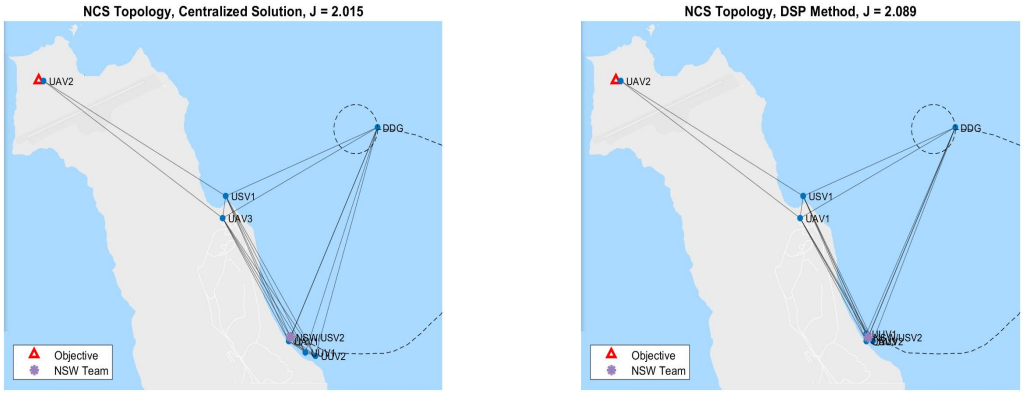


Figure 4.8. NCS topology for centralized and DS methods, late in the insertion phase.

Note that throughout the phase, the topologies generated by the two methods are largely similar, with the DS method often able to find a more robust communications pattern while maintaining the same sensor coverage. The total utility of the NCS throughout the insertion phase is shown in Figure 4.9.

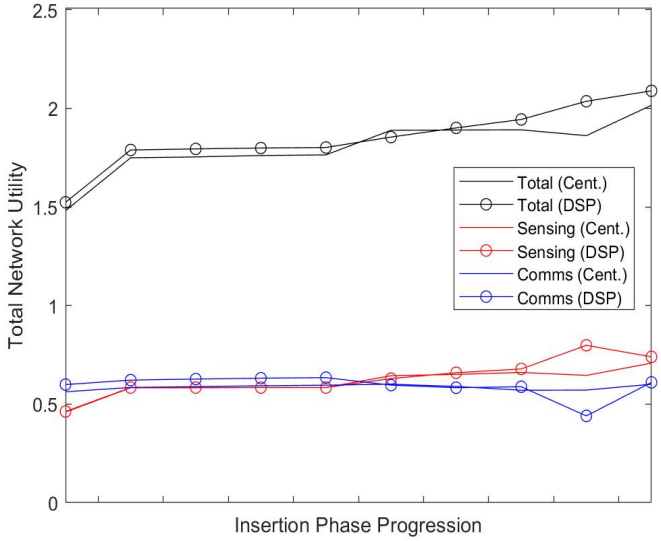


Figure 4.9. Total NCS utility throughout the insertion phase.

4.2.3 Infiltration

During the infiltration phase, the NSW team maneuvers up the island towards the objective, traveling along the road network. Note that the UUVs have dropped out of the NCS; surveillance of the coast and landing area is no longer necessary. Figures 4.10 through 4.13 depict how the NCS topology changes as the infiltration phase progresses.

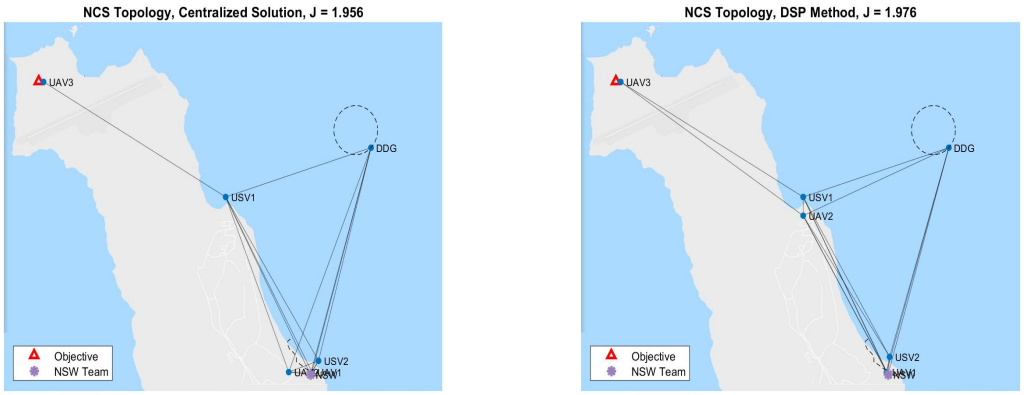


Figure 4.10. NCS topology for centralized and DS methods, beginning of the infiltration phase.

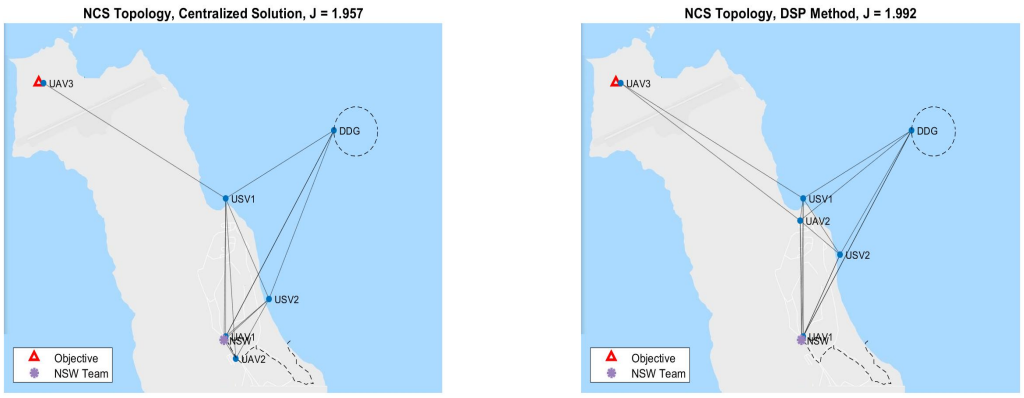


Figure 4.11. NCS topology for centralized and DS methods, early in the infiltration phase.

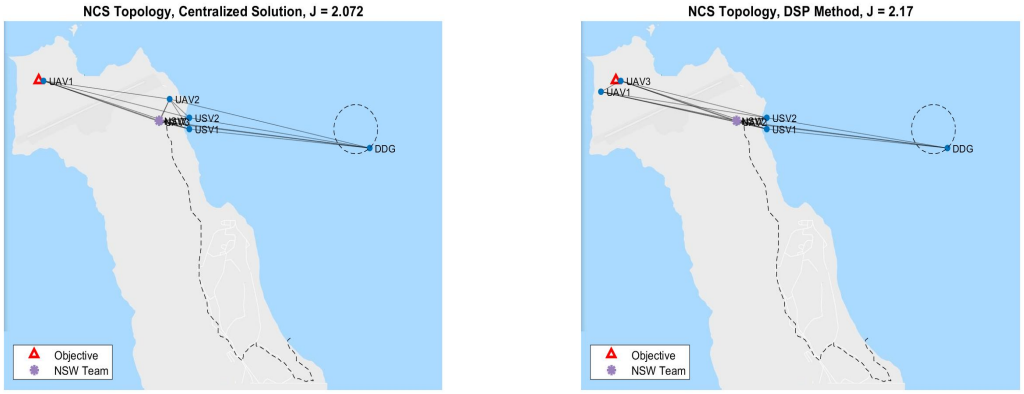


Figure 4.12. NCS topology for centralized and DS methods, late in the infiltration phase.

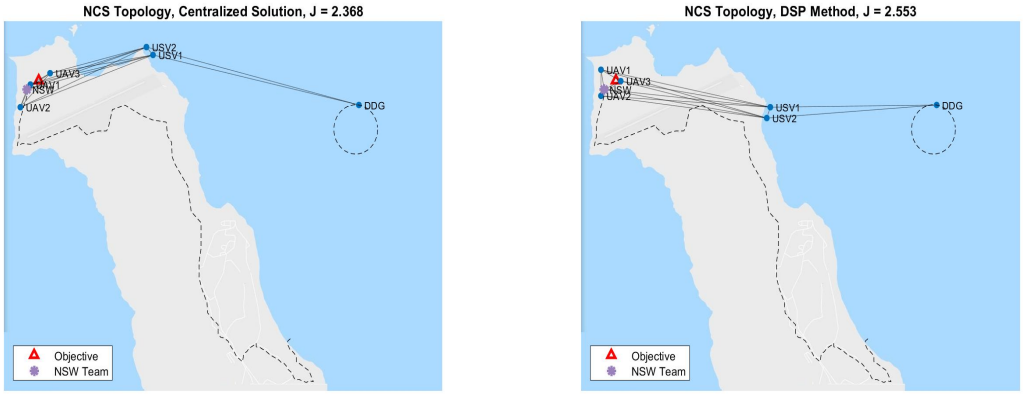


Figure 4.13. NCS topology for centralized and DS methods, end of the infiltration phase.

Again, the topologies generated by the centralized and DS methods are largely the same, and the utilities remain close throughout the mission. As during the IPB and insertion phases, there are times when DS is marginally outperformed by the centralized method, and other times when DS provides a better solution.

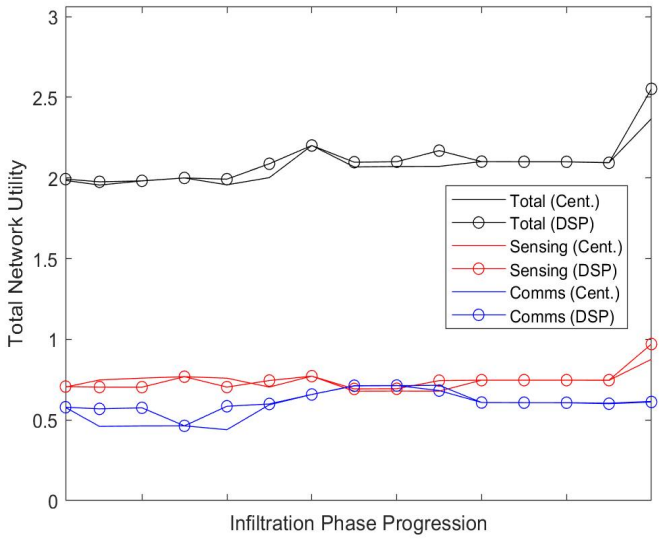


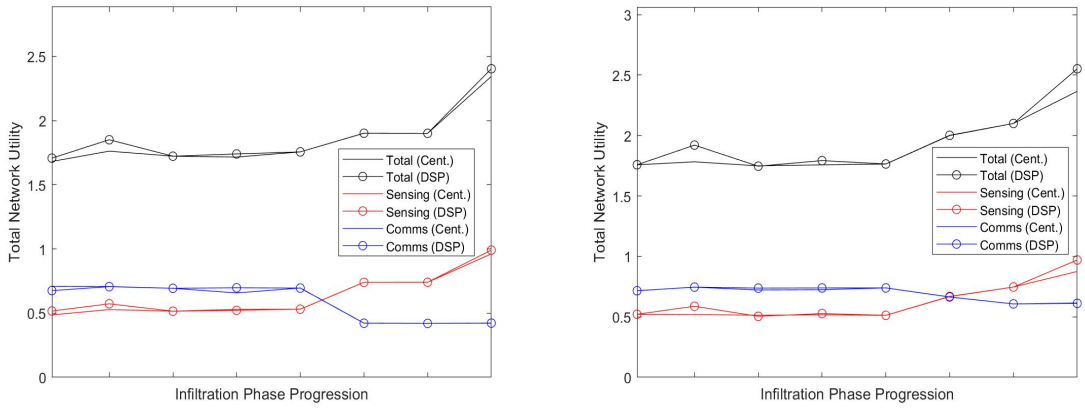
Figure 4.14. Total NCS utility throughout the infiltration phase.

In this chapter, we have demonstrated that DS is a viable distributed method for controlling a NCS of UxVs in a realistic scenario. Further, the performance is shown to be similar to or exceeds that of the centralized method, which has proven optimality guarantees [1], [20]. This framework allows for the decentralization of NCS control in a manner that increases network survivability without significantly impacting network performance.

CHAPTER 5:
Analysis

5.1 Network Size

So far this thesis has explored the performance of the DS algorithm vs. the centralized method when considering networks of relatively few nodes. In this section we explore the performance of both methods when applied to networks larger than those introduced in Chapters 3 and 4, which contained five to eight nodes. To accomplish this, the infiltration phase of the mission shown in Section 4.2.3 is used to examine the performance of the distributed and centralized methods when applied to increasingly larger networks.

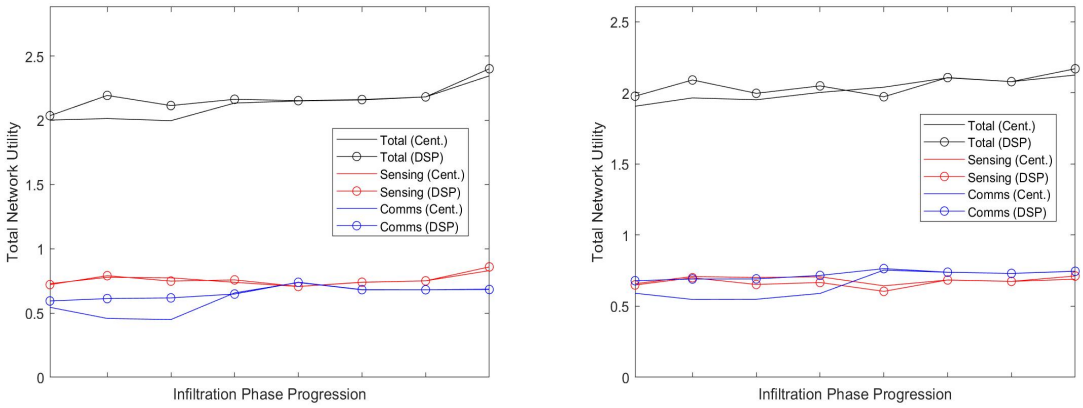


(a) Five node NCS

(b) Seven node NCS

Figure 5.1. DS and centralized performance during the infiltration phase for a five and seven node NCS.

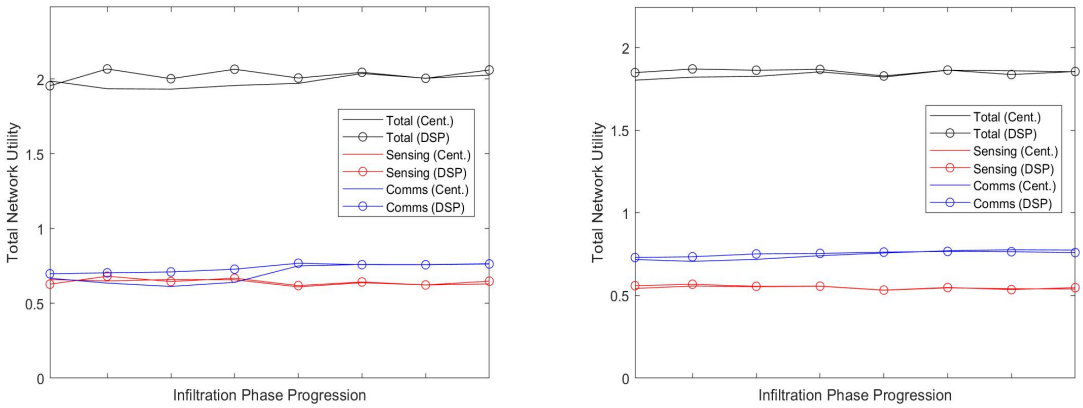
As with the results in Chapter 4, there are slight differences in the sensing or communications robustness utilities at different points during the mission phase (Figure 5.1). Overall, however, the performance between the two methods is similar for networks of a relatively small size (five and seven nodes).



(a) 10 node NCS (b) 15 node NCS

Figure 5.2. DS and centralized performance during the infiltration phase for a 10 and 15 node NCS.

Figure 5.2 shows the NCS utilities from the DS and centralized methods for a network of 10 and 15 nodes. Note that in the early phases of the mission, there is a substantial gap in performance between the two algorithms, which then closes as the mission progresses. Additionally, it is noteworthy that this difference in performance is due almost entirely to the greater communications robustness of the distributed network.



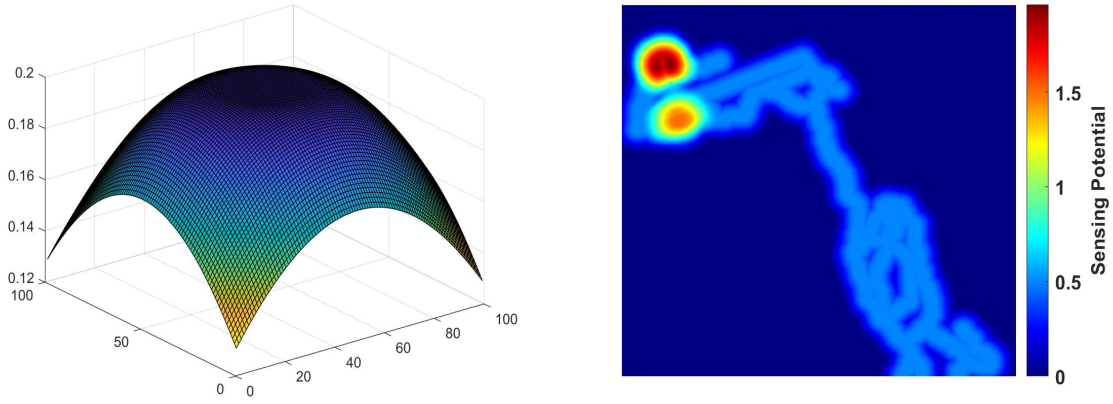
(a) 20 node NCS (b) 40 node NCS

Figure 5.3. DS and centralized performance during the infiltration phase for a 20 and 40 node NCS.

This same pattern is present, to a lesser extent, in the 20 node NCS depicted in Figure 5.3a, but it has largely disappeared in the 40 node NCS (Figure 5.3b). Note the flatness of the utilities throughout for mission for the largest network. A 40 node NCS is more than sufficient to provide sensor coverage over all the areas of interest, regardless of the NSW team's position. Additionally, the network is so interconnected that the communications robustness changes minimally as the NCS topology changes. This is consistent with the diminishing returns property of submodular set functions.

5.2 Complementary and Competing Subfunctions

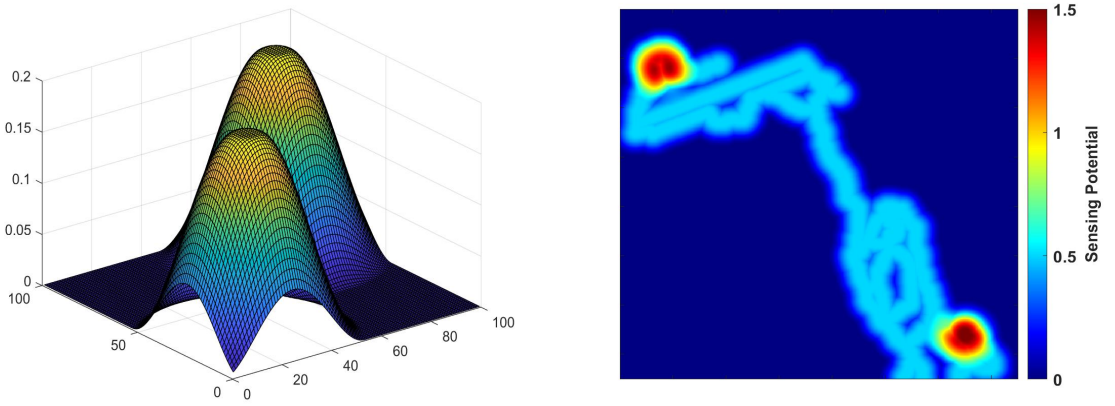
From the results of the simple problem in Section 3.3.3, it was suggested that in more complex scenarios with competing subfunctions, DS often outperforms the centralized solution. To illustrate this, Figure 5.4 shows two sensing potential fields $\mathbf{F}$ that are complementary to communications robustness.



(a) $\mathbf{F}$, simple NCS scenario (b) $\mathbf{F}$, MTX scenario, end of infiltration phase

Figure 5.4. Sensing potential fields $\mathbf{F}$ that complement communications robustness.

In both cases, the locations of high sensing potential are concentrated in one area, which results in a tightly connected NCS conducive to high communications robustness. In these scenarios the sensing and communications robustness subfunctions are complementary, and the resulting utilities and topologies from the centralized and DS methods are largely the same (Figures 3.18, 3.19, 4.13, 4.14).



(a) **F**, simple NCS scenario with two targets (b) **F**, MTX scenario, beginning of infiltration phase

Figure 5.5. Sensing potential fields **F** that compete with communications robustness.

In the scenarios presented in Figure 5.5, the locations of high sensing potential are more dispersed, and there is a resulting tension between the sensing and communications robustness utilities. It is in cases like these that have competing subfunctions that the DS method often outperforms the centralized method (Figures 3.21, 3.22, 4.10, 5.2).

Because the centralized method relies on sequential placement of nodes in the NCS, the communications robustness can only be considered for those nodes already placed. Conversely, the iterative nature of DS allows the full communications robustness picture to be considered when searching for optimal node placement. Thus the DS method can often refine the communications robustness of the NCS and improve on the centralized solution.

5.3 Observations

The MTX problem in this thesis is a complicated one. There are limits on where nodes can be placed due to a variety of factors including vehicle dynamics, prohibitions on sensor overlap, and initial NCS configurations. Additionally, the heterogeneous nature of the vehicles requires the use of separate and overlapping sensing utility fields **F**. Lastly, the dynamic nature of these sensing utility fields requires that the optimal NCS topology will change throughout the mission. All of these considerations suggest that many specific

conclusions that can be drawn from analyzing the performance of these two methods will be highly dependent on the nature of the problem they are applied to. However, analyzing the performance of the centralized and DS methods when applied to this specific scenario allows us to make some qualitative observations about this approach to distributed submodularity.

First, the performance of our DS method is comparable to that of the centralized sequential greedy algorithm; this is the most important contribution of this thesis. The benefit of using submodular function maximization for the centralized method comes from its proven optimality guarantees. The computationally complex nature of the problem limits our ability to solve it optimally in near real-time. Even knowing the optimal solution to compare our results to is well beyond reach even for networks of modest size [1], [20]. Thus, by showing that DS performance is aligned with the centralized method, we can conclude that the solutions it produces are near-optimal as well.

Second, scenarios exist in which DS outperforms the centralized method. Because the optimality guarantees in [20] are lower bounds, it is reasonable to conclude that better results may be found. From our analysis of scenarios with both competing and complementary subfunctions, it appears that DS is more likely to outperform the centralized method in complex scenarios with competing subfunctions. Qualitatively, if complementary subfunctions are thought of as making the problem simpler, then the simple greedy algorithm is more likely to closely approximate the optimal solution. Conversely, competing subfunctions are thought of as making the problem more complex, then even though the simple greedy algorithm is able to approximate the optimal solution to some (lesser) guaranteed level, there may be more room for DS to refine the solution.

Last, mission tasking and network composition influence the relative performance of these two methods. As can be seen in the examples in Section 5.1, there is a range of network sizes for which DS outperforms the centralized method early in the mission. This difference disappears both when the mission tasking (i.e. sensing utility field) changes and when the NCS size is either too small or too large. The key observation here is that while DS performs similarly to the centralized method most of the time, there are likely circumstances under which it provides a better solution. A mathematical basis for predicting the relative performance of the two methods based on these circumstances is left to future work.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 6:
Conclusion

In this chapter the findings and contributions of this thesis are summarized (Section 6.1), and the natural extensions of this field of research are discussed as a basis for future work (Section 6.2).

6.1 Contributions

This thesis builds on the previous work in [1] regarding a graph-theoretic framework for real-time control of a network of UxVs. A simple, novel algorithm is introduced that leverages distributed submodularity to arrive at near-optimal network topologies in a decentralized manner by utilizing local searches by and limited information sharing between the agents in the network. Next, a variety of simple scenarios demonstrate that this distributed approach performs similarly to the centralized method.

DS is then applied to the full MTX scenario. It is demonstrated that this distributed approach successfully approximates the performance of the centralized method, and under some circumstances outperforms the bounded optimality of the centralized solutions. The effects of network size and the complementary/competitive nature of the utility subfunctions on the relative performance of the two methods are explored.

This DS algorithm is presented as a novel, viable method for using distributed submodularity to near-optimally position network agents in real-time. The implications for this technology have wide applicability in the DoD, from the Special Warfare scenario presented here to distributed ISR and ASW networks. Civilian applications are also widespread and include wildfire response, resource exploration, or search and rescue.

6.2 Future Work

The current framework used in this thesis and previous work relies on a known deterministic map, something that is far from guaranteed in many real-life scenarios. To address this, a method that utilizes a machine learning technique called adaptive submodularity to handle

a stochastic environment could be employed. The outline for such an approach is detailed in [1]. Such an approach could also be used to optimize positioning policies that adaptively alter the utility function weights based on network inputs.

The strict communications requirements that the current centralized and distributed methods depend on are likely over-restrictive, and are also unlikely to be representative of an adversarial environment. Incorporating a communications policy that includes DTN would increase NCS robustness and allow for greater flexibility in mission accomplishment.

This thesis makes qualitative observations surrounding the circumstances under which DS can outperform the centralized method. Since the relative performance of the two methods is influenced by the sensing environment and general NCS topologies, further work is required to investigate a mathematical basis for these differences in performance. Understanding where and why the methods diverge would allow for greater insight into optimal NCS utility function design that maximizes mission performance.

List of References

[1] N. Wachlin, "Robust time-varying formation control with adaptive submodularity," Master's thesis, Naval Postgraduate School, Department of Mechanical and Aerospace Engineering, Monterey, CA, 2018.

[2] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause, "Distributed submodular maximization: Identifying representative elements in massive data," *Journal of Machine Learning Research*, vol. 17, pp. 1–44, Dec. 2016.

[3] A. Mokhtari, H. Hassani, and A. Karbasi, "Decentralized submodular maximization: Bridgin discrete and continuous settings," *arXiv*, pp. 1–10, Feb. 2018.

[4] M. Corah and N. Michael, "Efficient online multi-robot exploration via distributed sequential greedy assignment," in *Proceedings of Robotics: Science and Systems*, Cambridge, Massachussets, July 2017.

[5] G. Best, O. Cliff, T. Patten, R. Mettu, and R. Fitch, "Dec-mcts: Decentralized planning for multi-robot active perception," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 316–337, Mar. 2019.

[6] J. Rohrer and G. Xie, "Dtn hybrid networks for vehicular communications," in *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, Dec. 2013, pp. 114–120.

[7] D. Golovin, M. Faulkner, and A. Krause, "Online distributed sensor selection," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010, pp. 220–231.

[8] X.-M. Zhang, Q.-L. Han, and Z. Yu, "Survey on recent advances in networked control systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1740–1752, Oct. 2016.

[9] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ: Princeton University Press, 2010.

[10] J. S. Shamma and G. Arslan, "Dimensions of cooperative control," in *Cooperative Control of Distribued Multi-Agent Systems*, J. S. Shamma, Ed. West Sussex, England: Wiley, 2007, pp. 1–17.

[11] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 427–438, Feb. 2013.

[12] Y. Zhang and Y.-P. Tian, "Consentability and protocol design of multi-agent systems with stochastic switching topologies," *Automatica*, vol. 45, no. 5, pp. 1195–1201, May 2009.

[13] Y. Cao and W. Ren, "Distributed coordinated tracking with reduced interaction via a variable structure approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 33–48, Jan. 2012.

[14] E. Xargay, V. Dobrokhodov, I. Kaminer, A. M. Pascoal, N. Hovakimyan, and C. Cao, "Time-critical cooperateive control of multiple autonomous vehicles: robust distributed strategies for path-following control and time-coordination over dynamic communications networks," *IEEE Control Systems Magazine*, vol. 32, no. 5, pp. 49–73, Oct. 2012.

[15] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[16] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, Mar. 2015.

[17] X. Dong, B. Yu, Z. Shi, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles: Theores and applications," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 340–348, Jan. 2015.

[18] Y.-Y. Liu and A.-L. Barabási, "Control principles of complex systems," *Reviews of Modern Physics*, vol. 88, pp. 1–58, Sep. 2016.

[19] T. Summers, F. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 91–101, Mar. 2016.

[20] A. Krause and D. Golovin, "Submodular function maximizaion," in *Tractibility: Practical Approaches to Hard Problems*, L. Bordeaux, Y. Hamadi, and P. Kohli, Eds. Cambridge, England: Cambridge University Press, 2014, pp. 71–104.

[21] W. Ellens and R. Kooij, "Graph measures and network robustness," *arXiv*, pp. 1–12, Nov. 2013.

[22] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran, *Submodularity in Dynamics and Control of Networked Systems*. Switzerland: Springer International Publishing, 2016.

[23] G. L. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions," *Mathematical Programming*, vol. 14, pp. 265–294, July 1978.

[24] D. Golovin and A. Krause, "Adaptive submodularity: theory and applications in active learning and stochastic optimization," *Journal of Artificial Intelligence Research*, vol. 42, pp. 427–486, Nov. 2011.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California