Using pre-deployment planning for maintaining network connectivity in MANET models

by

Jarrett Chapman

A proposal submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 3 2019

Keywords: Mobile Ad-hoc Networks, MANET, Minimum Spanning Tree

Approved by

Dr. Alice Smith, Co-Chair, Professor of Industrial and Systems Engineering
Dr. Jeff Smith, Co-Chair, Professor of Industrial and Systems Engineering
Dr. Aleksandr Vinel, Assistant Professor of Industrial and Systems Engineering
Dr. Alvin Lim, Professor of Computer Sciences and Software Engineering
Dr. Abdullah Konak, Professor of Information Sciences and Technology, Penn State Berks
Dr. George Flowers, Dean of the Graduate School

Abstract

Mobile Ad hoc Networks (MANET) allow communication with little or no network infrastructure. Assuming no infrastructure is available, user nodes are defined as service demanding and agent nodes as service providing, communicating via device-to-device connections when within range. This dissertation uses pre-planning, assignment and possible re-planning, in a stochastic environment, to improve network connectivity over the mission.

A mission's progression is discretized, where each time step can be modeled as a Steiner Tree Problem with Minimum Number of Steiner Points and Bounded Edge Length (STP-MSPBEL). To solve the static problem, a Reduction method was developed, derived from a Minimum Spanning Tree (MST) solution method yielding a set of connecting points P. The Reduction method was verified and preformed as well as a mathematical model when comparing $|P|$. The Reduction method also performed as well as or (when possible) better than a solution method found in literature.

The dynamic deterministic environment problem considers successive static networks. A two stage solution approach is presented that first determines connecting points, P_t, at each time step using the Reduction per Time Interval (RTI) method and then assigns points in P_t to agent node tours with a genetic algorithm. In a majority of the runs this method performed as well as a mathematical model developed for validation and was better than a reactive approach.

The dynamic stochastic environment problem incorporated random deviations in user node movement. A MANET management method is presented that uses the pre-plan and assignment solution with the possibility of re-planning when the network has deviated beyond a given threshold. In comparison, using the MANET management method outperformed the reactive method in a majority of the runs.

Acknowledgments

All credit to YHWH who has given me time, intellect, resources, and people to support me.

Thank you my beautiful wife for your support and encouragement.

Thank you parents for making me do my third grade homework, setting me up for greatness.

Table of Contents

List of Figures

List of Tables

List of Heuristics

List of Notation

A, B, C, D	Set of locations used when calculating centroid node locations
α_1, α_2	Double exponential smoothing parameters
$ANCU$	Average number of connected users
A_t	Set of agent nodes
β^{max}	Maximum user node deviation vector
β_{tn}	Gaussian random number transformed deviation vector for node n at time t
B_t	Robustness metric at time t
C	Mathematical model commodity variable
c^{max}	Maximum user node cool-down time
C_t	Set of control nodes
C_t	Dynamic problem mathematical model commodity at time t variable
d^{dev}	MANET node deviation distance threshold
δ	Angular variation between checkpoints in tour
Δ_{replan}	MANET network deviation threshold
d_{ij}	Distance between nodes/locations i and j
DMMST	Dynamic modified minimum spanning tree method

E_t	Set of edges at time t
η^{max}	Maximum user node deviation duration
F_c	Mathematical model commodity supply/demand variable
f_{ce}	Mathematical model commodity flow variable
f_{DMMST}	Frequency of obtaining new MMST solution within the DMMST method
f_{Mutate}	Genetic algorithm mutation frequency
f_{RTI}	Frequency of using Reduction method
F_{tc}	Dynamic problem mathematical model commodity supply/demand at time t variable
f_{tce}	Dynamic problem mathematical model commodity flow at time t variable
$\gamma_1, \gamma_2, \gamma_3, \gamma_4$	Genetic algorithm objective function weights
G_t^*	Connected network at time t
G_t	Graph of nodes, connection points, and edges
g_t	Control node's virtual copy of the physical network G
H	Field of operation's height
H_{tij}	Acyclic path between nodes i and j at time t
I_{Assign}	Number of iterations for assignment method
I_{DMMST}	MANET number of DMMST iterations in RTI method during re-planning
I_{GA}	MANET number of genetic algorithm iterations during re-planning
k	Node type

L	Node tour length
M	Dynamic problem mathematical model big-M method
m	Dynamic problem mathematical model maximum agent node movement rate
M	Mathematical model big-M method variable
$metric_{method}$	The degree of the set of connecting points determined by *method*
MMST	Modified minimum spanning tree method
MST	Lin and Xue's minimum spanning tree based solution method for the MST-MSPBEL problem
M_t	Set of user node tours with locations defined at each time step t
$\mu_{metric_{method}}$	Mean of *metric* values derived by *method*
$\mathcal{N}, \hat{\mathcal{N}}$	Gaussian Distribution and a random number from Gaussian Distribution
$noia$	Number of infeasible inter-time step agent node movements in an assignment solution
N_t	Set of all nodes (control, user, and agent)
n_{ti}	Node i at time t
Ω	Genetic algorithm population of solutions
p^m	Required minimum number of deployed agent nodes
$popsize$	Genetic algorithm population size
Ψ	Genetic algorithm mutation swap vector

P_t	Set of locations at each time t that connect the network
Q_t	Set of agent node tours
r	Maximum node-to-node connectivity distance
R()	Random number generator
RTI	Reduction per time interval method
S	Time steps between checkpoints in tour
s	Forecast time step horizon for the network/nodes
$\sigma_{metric_{method}}$	Standard deviation of *metric* values derived by *method*
S,M,X,L	Problem size classes
STP-MSPBEL	Steiner tree problem with minimum number of steiner points and bounded edge length
T	Mission time
t	Dynamic problem time index
τ	Actual time between each time step t
θ	Initial angle of movement
θ^{max}	Maximum user node deviation angle
u_i	Mathematical model agent node usage decision variable
U_t	Set of user nodes
u_{ti}	Dynamic problem mathematical model agent node usage at time t variable

$U_{ti}[x], U_{ti}[y]$	Dynamic problem mathematical model user node x, y location at time t variable
v	Maximum node movement rate
v_{ij}	Mathematical model arc usage variable
v^{max}	Maximum user node change in velocity
v_{tij}	Dynamic problem mathematical model arc usage at time t variable
W	Field of operation's width
w	Network connectivity minimum service level
X_i, Y_i	Mathematical model agent node x, y location variables
X_{ti}, Y_{ti}	Dynamic problem mathematical model agent node x, y location at time t variable
x, y	Node or checkpoint location in area of operation
$Z(G)$	Genetic algorithm weighted objective function

Chapter 1

Introduction

1.1 Background and Motivation

United States wartime combat has changed significantly as improvements in computing and wireless communications have been incorporated. Specifically, unmanned aerial vehicles (UAV) such as the Shadow-200 FQ-7B [71] and Predator [64] have been developed with the primary goal to reduce the exposure of the human operator to potentially dangerous situations. Smaller in scale, custom built multi-rotor copters are also being developed. In addition to saving lives, they may support intelligence gathering, reconnaissance, munitions support, and network connectivity between forward deployed units and the command and control units[66].

In the wake of increasingly frequent natural disasters, installed communication infrastructures can be rendered inoperable. Simultaneously, there arises a need for a communication infrastructure when surveying damage or searching for individuals or items. Thus, search and rescue operations could benefit from mobile ad-hoc networks, an as needed network topology defined by device to device communication.

In such contexts, there are entities that specifically issue commands, carry out a mission, or maintain connectivity. Each is later defined in Chapter 3 as control, user and agent nodes, respectively. Because the control node issues commands, it is necessary for all other nodes to be network connected to receive commands. Basically, the user nodes are service demanding nodes and agent nodes, service providing nodes. This communication network will be modeled as a Mobile Ad-Hoc Network (MANET).

In a military context, user nodes represent deployed soldiers. The size of a group of soldiers is specified in U.S. Army Pamphlet 10-1[68]. The smallest grouping of soldiers is known as a fireteam/team and consists of two to four soldiers [1]. Each unit level in the U.S. Army consist of three to five smaller units. The successively larger groupings with approximate sizes are shown in Table 1.1 [68].

Table 1.1: Military Unit Type and Sizes [68]

Element	**Sub-element**	**Soldiers**
Fireteam	Soldiers (4-5)	4-5
Squad	Fireteams (x2)	9 - 10
Platoon	Squad (3-5)	16 - 44
Company	Platoon (3-6)	62 - 190
Battalion	Company (2-7)	300 - 1000
Brigade	Battalion (3-6)	3,000 - 5,000
Division	Brigade (>3)	10,000 - 15,000
Corps	Division (>2)	20,000 - 45,000
Army	Corps (>2)	50,000+

In search and rescue (SAR) missions, user nodes represent first responders or even civilian members participating in the effort. Unit sizes for search and rescue reconnaissance are explained in the Australia's National Land Search Operation Manual. "Composition of teams for general search ... should be kept small (4 to 6 persons)...[or] larger teams can be utilized depending on the terrain." For coordinated searches, after reconnaissance, e.g. during a contact search, the manual specifies eight to twelve persons, dependent on conditions [80]. Alternatively, the United States Coast Guard guide specifies a two person team with approximately twelve to thirty teams for ground search and rescue exercises [53]. The number of user nodes required for a given model or static location problem will be specified in Section 3.3 in reference to values listed here.

Nodes will move based on a sequence of locations defined as a tour. User node tours are pre-defined. Each use node follows its tour independently. To maintain connectivity the control node determines the agent nodes' locations. The literature, covered in Chapter 2, shows that there have been few published works that address continuous space stochastic environment path planning and those that do, use purely reactive methods. This is based on the assumption that the user nodes move without instruction or coordination from the control node.

In this dissertation, it is assumed that the control node has knowledge of the user nodes' intended movements. With this, a deterministic model of the system can be used to plan agent node movement. In such a case, the location of any node at any time is known or can be determined. In a stochastic environment, when faced with uncertainty where user nodes may deviate from their given tour, agent node movement can be modified or redefined as needed to maintain network connectivity.

1.2 Problem Definition

Given a network composed of control and user nodes that are given specific movement plans, this dissertation will provide agent node tours that best connect the network during the duration of the mission and minimize the number of needed agent nodes. Define this as the minimum number of agent nodes path planning problem.

A problem instance will be defined as a collection of nodes moving about a defined area of operation. User nodes are, in the military context, representative of soldiers or, in the search and rescue context, representative of first responders. The field of operation has boundaries, generalized to be rectangular in shape, thus having minimum and maximum x and y coordinates. This dissertation focuses on two dimensions but this could be relaxed to incorporate three dimensions by adding altitude for flying nodes. Users can move within the field of operation, generally starting at the control node, to accomplish certain tasks. The control node is the point of command for the operation. Each type of node will be elaborated on in Chapter 3.

1.2.1 Agent Node Path Planning

For all missions, a deterministic model is used during the planning stage. The planning solution finds the set of agent node locations defining its path. The planned path will be used to manage agent nodes movement during a mission in either a deterministic or stochastic environment. During deterministic environment simulations, the mission operates identically to the mission in the planning stage. Incorporating uncertainty, regardless of the cause, may result in need to re-plan.

1.2.2 Assumptions

This problem is based on a few assumptions. First, no permanent communication infrastructure exists, thus the need for an ad-hoc network. Second, prior to deployment, the control node assigns tours for user nodes and as a result, in a deterministic planning case, knows where each user will be at any given time. This can not be assumed in the stochastic environment. Finally, considering that agent nodes are expensive, there is also a goal to minimize the number of agent nodes needed to provide full connectivity over the mission.

For purposes of this research, user node movement will be driven by a checkpoint system. Checkpoints are defined to model discrete time steps in a continuous time mission. A checkpoint specifies a location that a node need to be at a given time. Each mission will have a certain number of checkpoints each user node must visit. After completion of a user node's checkpoint list, it will remain at the last checkpoint. In the deterministic case, user nodes follow a straight line between checkpoints. Stochastic cases will also be considered where user node movement includes a combination of random and checkpoint directed movement.

The application of this dissertation is based on the assumptions that a mission planner would plan in a deterministic environment, defining user node tours, determining connecting points, and assign connecting points to agent node tours. Then, deploying the planned system in a stochastic environment the mission planner would constantly monitor and forecast user node locations. If user nodes deviate significantly from their plans, the agent node tours are re-planned.

1.3 Research Decisions

Given pre-deployment user node movement plans, this dissertation attempts to determine the set of agent node tours that will maintain the connectivity of the network. This will be achieved by a two step solution process. The first is the process of solving successive static location problems at discrete time steps yielding sets of points needed to connect the network. A point identifies a time and location where an agent node would be needed. As a result the number of needed agent nodes for a problem will be at least the maximum over all time steps of the minimized number of points for a fully connected network at all time steps. The second step is the assignment of the aforementioned points to agent node tours. In a deterministic environment this assignment will ensure a connected network at all time steps. Using the same tours in a stochastic environment does not ensure an always connected network. For this reason, a robustness metric will be included as part of the agent node tour assignment process.

1.4 Research Objectives

The MANET literature offers many methods of dealing with path planning, clustering, routing, positioning, and connectivity problems. However, continuous space tour planning to support mobile users to ensure network connectivity is a niche area. Recent works in the field have used genetic algorithms as well as particle swarm optimization to approach the problem of path planning. However, these methods have been entirely reactive in their approach to managing network connectivity.

The goal of this dissertation is to determine agent node tours to illustrate the efficacy of pre-deployment planning compared to reactive only locating. This will be done by comparing the average number of connected user nodes and a measure of robustness over the time of the mission. However, when modeling the problem, the number of connected users is considered as a constraint. For the deterministic environment, the average number of connected users is always equal to the number of user nodes. This assumption is not always true for the stochastic environment, in which the objective is to maximize the average number of connected users and a measure of the network's ability to remain connected, termed robustness.

To identify the benefits of this research its performance in a stochastic environment will be compared to a reactive positioning method in a stochastic environment. By including a planning step, this dissertation attempts to show improvement in average number of connected user nodes over the course of a mission using equal or fewer agent nodes.

In a stochastic environment the user node may deviate from planned tours. At these points, the efficacy of the agent node plan will be evaluated. A method is developed to identify significant deviations from planned tours where additional actions, such as re-planning, will be warranted.

The military and SAR contexts have similar objectives and constraints that will illustrate this dissertation's applicability to a variety of problem contexts. The two contexts will be evaluated at differing sizes to illustrate scalability and limitations. Planning will be conducted in a deterministic environment. The primary objective of this dissertation is to plan agent node tours that minimize the size of the agent node set while maximizing average number of connected users in a stochastic environment.

1.5 Research Contributions

This dissertation will extend MANET path planning by using pre-deployment planning for agent node movement and also offer a method to re-plan based on stochastic user node movement. The combination of planning and the option to re-plan in a stochastic environment should improve network performance over the completely reactive agent node movement systems currently found in literature.

1.5.1 Planning

Solving successive static problems will yield a set of points at each time step determining the minimum connectivity requirements at each time step. To ensure that a static solution is optimal, a mathematical model will be used to validate it. To validate the dynamic problem, the static problem mathematical model will be modified. For the deterministic case, assigning the points to agent nodes and enforcing movement constraints can be modeled as a maximum flow problem.

With the incorporation of the robustness metric the problem becomes complex and non-linear. To help improve network connectivity, especially in a stochastic environment, the robustness metric favors well connected networks, specifically networks with a higher occurrence of connections between two nodes of high degree. Combinatorial meta-heuristics are well suited to handle complex solution spaces. Movement constraint violations and robustness metrics can be measured easily with the encoding of these meta-heuristics. A genetic algorithm will be used for the deterministic environment assignment process. The resultant assignment solution will be agent node tours.

1.5.2 Stochastic Missions

In the stochastic environment agent node tours will be initialized with the deterministic assignment solution and user nodes may deviate from their given tours. The network is continually monitored to determine if the forecast deviations are significant. If so, then agent node tours will be modified to attempt to maintain network connectivity. All future agent node tour checkpoints are removed and, similar to the assignment method, are assigned tours from a new set of connecting points based on user node location forecasts. To contrast, a reactive approach operates myopically prescribing agent node locations for only a few time steps in the future. This dissertation will modify agent node tours considering user node tours for all future time steps.

1.6 Research Outline

The following chapter presents the associated literature and identify where this dissertation adds to the field of research. Chapter 3 details the Mobile Ad Hoc Network model's components, notation, formulation, problem classification, and the research plan. Solution methods for static and dynamic problems are described in Chapter 4 and 5. Lastly, results and analysis of this experimentation is given in Chapter 6.

Chapter 2

Literature Review

2.1 Communications

2.1.1 Radio

The focus of this dissertation is usage of a fixed-range communication method to establish node to node connections. In the military context, assume that nodes will use the SINCGARS (Single Channel Ground to Air Radio System) which is stated to have a transmission radius of up to ten kilometers [70]. Also assume that such radios would not be available for the search and rescue context, resulting in the use of commercially available radios that generally have an effective transmission range of about one kilometer. The chosen connection method is only used as a parameter in the system. Regardless of the particular connection method, this dissertation uses network connections and location predictions to improve system operation.

2.1.2 Ad Hoc Networks

In traditional networks, relay nodes are usually at a fixed location and communicate with some other device, such as a server, via fixed connection points like hubs, routers or radio towers. Such networks require infrastructure to properly communicate between nodes.

An ad hoc network is one that is created as needed. It is a collection of several similar nodes or smaller networks that connect without the need for fixed infrastructure. Any communicating node should be able to communicate directly or act as a transceiver node to relay communication to other network nodes [62]. One unique ability that such networks have is the ability to drop and add nodes. A dropped node could affect the network's overall connectivity, but robust routing algorithms can, up to a point, be used to compensate for this [86, 85]. Nodes may be dropped or added due to their connection strength, where connection strength is directly proportional to the transmission power, node to node distance, and natural or unnatural interference.

The effect of adding nodes is the key premise behind this dissertation. For any given pair of connected nodes, if these nodes are relocated beyond either of their connection radii, then the nodes become disconnected. These nodes can be reconnected with the addition of a node acting as a transceiver. These added nodes are defined as agent nodes.

2.1.3 Sensor Networks

A derivative of ad-hoc networks, sensor networks have the typical properties of a MANET, (nodes can be added, removed, and can act as a transceiver), but have low-powered nodes that are used to gain some data about the network or surrounding environment and relay that information to some central intelligence [8, 44]. Generally, sensor networks consist of stationary nodes, with some works using a few mobile sensor nodes to re-connect the network. Monitoring could be for meteorological or surveillance purposes, conducting tasks such as reporting temperature and humidity or intrusion into an area, respectively.

The sensor network's goal is to maximize coverage and sensing ability while cooperating to relay data, but to do so with finite power. The transmission of packets is of more importance than the idea of dynamic node placement. This problem type usually assumes homogeneous node types with a given radius of sensing and transmission. Similar to ad-hoc networks, the transmission of data from the sensor to the base station is via other sensor nodes.

If the only objective is to ensure that every point within a given area is within connection range of a sensing node, then a hexagonal arrangement satisfies this. It minimizes the number of nodes needed to cover while maximizing coverage area, see Figure 2.1. Frenkiel illustrates this principle in the context of wireless tower placement [22].

Losses of nodes could occur due to insufficient power, alterations, or damage to the sensor caused by the environment or an antagonizing force. The loss of a node can be mitigated by replacing the lost node, or relocating working nodes to patch the gap in coverage [41, 44, 78].

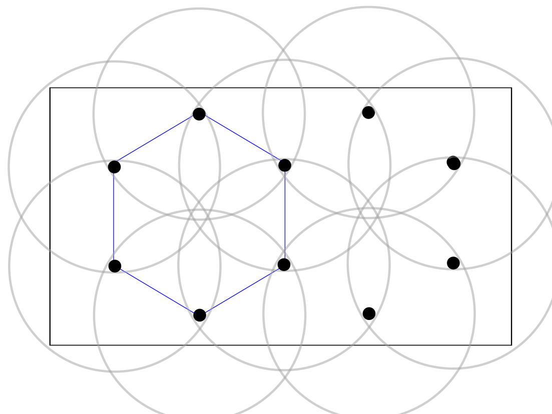


Figure 2.1: Optimal Maximum Coverage Sensor Solution

The mentioned works in the sensor network literature address the problem of locating nodes to create a connected network, but many of the requirements and parameters of the sensor networks do not apply to this dissertation. In particular, the sensor network's requirement to cover an area is not considered in this dissertation and more focus is placed on the connected network requirement. This means that each node must be connected to any other node, either directly or indirectly through the network.

2.2 Mobile Ad Hoc Networks

Mobile Ad hoc Networks (MANET) have all of the properties of the aforementioned ad hoc networks, with the additional consideration that the nodes may be mobile. The location of any given node at time t may not be the same at time $t+1$. The addition of mobile units adds a significant degree of difficulty to the ad hoc networking problem.

A majority of the MANET literature focuses on clustering, routing, positioning, and connectivity models. Clustering [58, 57, 3, 76, 77, 36] determines stable groupings to better manage network communication protocols. Routing [11, 33, 50, 49, 60, 56, 2] is focused on determining efficient paths to send packets. Positioning [38, 82, 32, 46] determines the locations of nodes (for use in routing and connectivity). Connectivity [83, 6, 7, 42, 75] deals with improving networks by managing node and network degrees. These methods use the existing topology, without modifying locations, to make decisions to improve network operations. Here, the tasks of routing and positioning are considered to be inherent features of an operational network. Clustering is considered, but only to improve network connectivity.

Two areas of research, Topology Management and Path Planning, improve network operations by manipulating the physical entities of the network. This is in contrast to the previous ones that seek to improve functioning of an existing network.

2.2.1 Topology Manipulation

Topology was defined by Gilbert and Pollak [24] as the set of edges that define the network. Practically, it is applied in managing node's transmission power where having too many high powered nodes can create intra-network interference; limiting transmission/receiving ranges would reduce this. Similarly, adding nodes would require less power for distant nodes to bridge a gap to be connected. This dissertation is closely related to the topology manipulation area in that there may be a need to add or relocate nodes.

Kusyk *et al.* [39], similar to the later discussed works of Cho *et al.* [12, 13], investigate agent node locating in a MANET with enemies. With the goal to maintain each node's network connectivity and coverage of the area of operation, a decentralized game theory approach using a genetic algorithm (GA) with a 128-bit binary encoding evaluated node configurations. There is no mention of using prior knowledge to make better decisions. It is assumed that none of the nodes have any knowledge of enemy locations. Kusyk *et al.* also present a new node spreading bio-inspired game (NSBG) method that address the maximal sensor coverage network. Kusyk *et al.* allow all nodes to be mobile.

Kulkarni *et al.* [37] implement particle swarm optimization (PSO) for a swarm of robots, where each particle is a deployed robot that is assumed to always knows its location. The objective of the swarm in the two experiments were to locate and cover a target. Unlike this dissertation, nodes had no prior knowledge of the path to take, nor was there specification that the network be completely connected.

Han *et al.* [29] implement a MANET simulation initialized with a connected network and one deployed drone used to improve (and maintain) network connectivity. Connectivity was evaluated using global message connectivity, network bisection connectivity, and the k-connectivity metric. Results show that these metrics do improve the network's connectivity, but using only one agent node and no prior knowledge sets this dissertation apart from the work of Han *et al.*.

Dogan *et al.* [52, 51] use a physical network (or testbed) of laptops, desktops and some mobile nodes in a lab environment. Locating is handled with a Simple Genetic Algorithm (SGA) with chromosomes consisting of "location, speed, and direction as a binary string that is divided into genes." The performance relative to area coverage was improved with increased node movement speed and increased communication range.

Hunjet *et al.* [31] enhance mobile ad hoc networks through Networked Autonomous Vehicle (NAV) placement and topology manipulation. The placement of the mobile nodes was done with PSO via an objective function maximizing the percentage of connected nodes and minimizing the used power for transmission. An exhaustive search using topology manipulation and a maximum of two NAVs validated the PSO. Results show that adding NAVs with use of topology manipulation reduces network power consumption, increasing longevity of the network, and increases the percentage of connected nodes. Here, centralized, not autonomous, control of agent nodes is assumed. Also, the number of agent nodes considered in this dissertation is much larger than two NAVs.

These works, [24, 39, 37, 29, 52, 51, 31], have similarities to the ideas of this dissertation, but most of the works in this field are related to sensor or (maximum) coverage networks. None use pre-deployment planning, and are all strictly reactive in nature. Additionally, controlling power usage is not a concern herein.

2.2.2 Path Planning and Node Locating

This research area focuses on determining locations for nodes at a given time to maintain network connectivity or accomplish a set of tasks or goals. For node placement and path planning for network connectivity, the difficulty of continuous space searches as well as combinations of large numbers of nodes must be considered. The following works use meta-heuristics which are capable of adequately handling combinatorial and continuous space searches.

Sahin *et al.* [15, 14] use distributed genetic algorithms to locate nodes to cover an area with the possibility of losing assets. Within this dissertation's framework, network connectivity (not area coverage) is the primary goal. Unlike Sahin *et al.*, this dissertation uses a centralized agent node locating method.

Hsieh *et al.* [30] create an actual application of robot communication and path/task planning on a nodal network. Because there are no added nodes such as agents, communication is maintained only by Line of Sight (LOS) connections. In this dissertation, connectivity is the goal with agent node locating as a form of route planning.

Guan *et al.* [28] define and solve the Electronic Warfare problem in an two stage LP. The first stage decision is placement of an Electronic Attack (EA). The second stage decision is placement of Electronic Support (ES) nodes used to connect EA nodes. A genetic algorithm was also used to handle this complex problem. The idea here is to provide jamming to enemy networks (EA nodes) and maintain ally networks (via ES nodes) while also minimizing the number of nodes needed for deployment. Similarly a two staged approach is also defined for this dissertation but in a different context. Planning based on user node's pre-defined movement takes place in the first stage, and reactive locating occurs when they must deviate. Additionally, the problem presented by Guan *et al.* is static, considers a discrete solution space of candidate solutions (not continuous space), and uses LP (and GA) solution methods.

Reina *et al.* [55] locate static (agent) nodes at discrete locations using a genetic algorithm to increase connectivity of a MANET in an emergency response center. Similarly, this dissertation considers nodes that will cover the mobile users of the system but with mobile agent nodes.

2.2.3 Applicable Works in Robotics Literature

In a real world application, finding locations to maintain network connectivity has been applied to robotic teams. Antonelli *et al.* [4, 5] develop a function for a set of robots consisting of a task directed unit and a set of units acting as a mobile antenna. The proposed work hierarchically uses a set of functions (stay connected to base, stay connected to neighbor node, stay connected to task robot), dynamically reordering based on the robots' states, to determine the robots' movement velocities. The resultant robotic network is a task directed robot connected by a chain of mobile (robotic) antennas to a base station. This dissertation's MANET model is similar to the model of Antonelli *et al.*, however it considers centralized planning for path determination, not a distributed reactive method for maintaining a network.

Ulam and Arkin [63] focus on a suite of four movement behaviors for robot teams to reconnect a disjointed network. These are all reactive including robotic behaviors to move towards inclines, retrace visited way points and others.

Vazquez and Malcolm [74] develop methods to expedite and coordinate a robot team's exploration and mapping of a facility. The robot team maintains connectivity without additional units (*i.e.* agent nodes) and avoids collisions. The system considers only robots with distributed logic; no human nodes are considered. It is similar to a mobile sensing network with the objective of surveying an unknown area.

2.2.4 Mobile Ad Hoc Networks with Particle Swarm Optimization

Recent works in locating nodes within the MANET framework have mostly used particle swarm optimization (PSO). Canonically, the PSO uses particles to search a continuous solution space with its movements based on a weighted measure of its current position compared to its best found solution and a global (or neighborhood) best found solution. Defining multiple sets of particles as the set of agent nodes, the swarm searches for locations that connect the disconnected network of user nodes to the control node. The PSO does well in continuous space searches by leveraging shared global knowledge with the local search of each particle to find good (if not optimal) solutions. This section details the previous work in the area of maintaining network connectivity using PSO for node locating.

Konak *et al.* [35, 34], Dengiz *et al.* [17, 18], and Cho *et al.* [12, 13] have used particle swarm optimization in determination of node locating to improve the overall connectivity of the network. Konak *et al.* [35] define user nodes (service demanding) and agent nodes (service supplying) that operate in the network. Assuming all nodes have GPS and their locations can be obtained via radio or satellite link, current and past locations are known and can be used to predict future locations of a node. The primary objective is to maximize network connectivity, or the total number of network connected node pairs. A secondary objective is to maximize bandwidth between user nodes and control nodes. There also exists a tertiary objective that creates an attraction point for locations that would reconnect an unconnected network. This enables the agent nodes to have some level of intelligence to rejoin the network if isolated. Using double exponential smoothing for future user node location predictions, the allocation algorithm (ALOC) looks $t + H$ time steps into the future to determine agent node placement where t is the current time and forward prediction horizon is H.

In a more recent work, Konak *et al.* [34] develop a flocking meta-heuristic that balances "separation, cohesion, and exploration behaviors." The developed rule set was designed to offer simple, decentralized augmentation of the network via the agent nodes. Here, distributed methods are not considered. A centralized locating method uses network information to make decisions beneficial to the entire network.

Dengiz's dissertation [17] is also one of the pioneering works in the field of MANET node locating with meta-heuristics. The first approach used a non-deterministic binary decoding

GA (NDBGA). Agent movement/placement was defined to use either the NDBGA or PSO in both static and dynamic simulation scenarios. Also, to show the practicality of the meta-heuristic methods, a mixed-integer program (MIP) was developed. Specific performance metrics for the set of experiments show that both are efficient at connecting the networks and in terms of solution time, the NDBGA performed better and both outperformed the MIP.

Dengiz *et al.* [18] also assume that node locations are known via mounted GPS units and communicate their locations via low frequency radio or a satellite modem. Similar to Konak *et al.*, the primary objective is to maximize all node to node connections or total network connectivity. Secondary and tertiary objectives are to maximize bandwidth between user node clusters and minimize distance to attraction points in both connected and unconnected networks. Dengiz *et al.*developed locating reactive methods, whereas this dissertation uses planning and reactive methods for locating.

Cho [12] uses PSO exclusively for agent node locating and movement in the context of military operations (Network-Centric Warfare) where enemies are present that can hinder communications or disable nodes. User node movement is based on the Random Waypoint Model (RWM), Convoy and Defense (CD), or Search and Rescue (SR), with the system using current and past locations to predict potential user node future locations. User node intelligence is extended from Konak *et al.* (and Dengiz *et al.*) to be able to react to encroachment into enemy range/territory. A Java simulation was developed to test the system similar metrics to Dengiz *et al.* and Konak *et al.*.

Cho *et al.* [13] improve the MANET PSO by incorporating another sub-objective, the pre-deployed agent level (P) that helped relocate nodes so that each will be more accessible to user nodes as the mission progressed. The addition of the P metric increased the number of connected units over the course of simulated missions.

Unlike the works of Konak *et al.*, Dengiz *et al.*, and Cho *et al.*, pre-deployment information to plan routes for the agent nodes will be used. Here, discrete time steps are used to solve a series of static problems that together represent the MANET over its operational time. For any given unconnected network, an algorithm was developed to minimize the number of, and determine the location of, the agent nodes required to connect the network. Each static problem is, in essence, a graph problem and is expounded upon in the next section.

2.3 Locating Methods Using Graph Theory

Though it has been around for some time, graph theory has recently gained more attention in the area of mathematics. West [79] explains the basics of graph theory. Graphs are generally notated by G and consist of a vertex set and an edge set, $V(G)$ and $E(G)$, also represented as $G = (V, E)$. The graph is the relationship between the vertex and edge sets and can be used to model systems with interrelated components. If every vertex has a relationship to every other vertex, directly or indirectly, it is called a connected graph. If there is an associated cost of flow along edges, then the graph is called a weighted graph. For purposes here, graphs are undirected and weighted, where the weight is the Euclidean distance (used as a connection surrogate) between nodes.

In the MANET framework, the vertex set is the set of nodes that needs to be connected. Edges are the connections between nodes, but due to radio transmission range limitations, edges are only considered if the weight/distance is less than the connection radius of the radio. This idea is developed further in Section 2.3.3, after some discussion of graph theory and its application to the MANET framework.

2.3.1 Minimum Spanning Tree

The Minimum Spanning Tree (MST) is a special type of connected graph that uses the minimal distance set of edges. Kruskal's Algorithm for MST shows that even simple methods can yield decent results [79]. Kruskal's solution is, in fact, an optimal solution for connecting any network. In the framework of MANETs, the MST is applicable for not only creating a connected network but also the max bandwidth network (assuming maximum bandwidth is directly related to distance and not hop-count or effects of interference).

Graham and Hell [26] review the origin and history of the MST as well as three algorithms for solving it. Each algorithm uses the basic idea of the Kuskal's MST heuristic to solve, with some key differences.

- Algorithm 1 - sort edges by length, start with shortest and add next shortest edge (Kruskal's Algorithm)

- Algorithm 2 - sort edges by length, start with a random edge, add next shortest edge (variation of Kruskal's Algorithm)

- Algorithm 3 - sort by node index, for each node, add the closest neighboring node (Boruvka Alogorithm)

These algorithms provide a way to connect a network but do not consider the addition of nodes in real space which could reduce a graph's total edge length in the way a Steiner Tree solution does. Figures 2.2b and 2.2c show how the total edge length could potentially be reduced [24]. Figures 2.2b and 2.2c are the same network, with the total edge length of the MST solution equal to 2 and the STP solution equal to $\sqrt{3}$. Furthermore, the MST problem does not have a limitation on edge length, a feature vital to a communications framework where signals decay with distance.

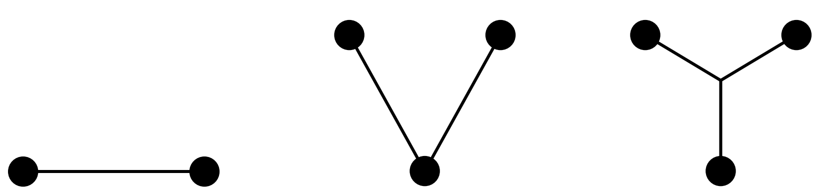
The Steiner Tree Problem addresses the need for additional points not in the original set of vertices that are placed anywhere in real space. The rectilinear version is more applicable to transportation frameworks. The Euclidean Steiner Tree Problem most aptly fits the MANET framework. Such formulations will be discussed in the next section. Later, the inclusion of an edge length constraint and minimization of number of Steiner points are also presented.

2.3.2 Euclidean Steiner Tree Problem

Given a graph $G = (V, E)$, the Euclidean Steiner Tree Problem (ESTP) is to determine the placement of Steiner points that will minimize the sum of all edge lengths while creating a connected network. The solution to the STP is a Steiner Minimal Tree (SMT). Classically, Steiner points in a Steiner tree are defined as points that have a degree of three, $|G| = 3$ [24]. Optimal SMTs have Steiner points with a degree of three, with the angle between edges equal to 120 degrees, detailed by Gilbert and Pollak [24]. The minimum degree assumption is logical in that a single vertex would not need a Steiner point; two vertices would only require a point midway between the two; three vertices would need a Steiner point to minimize the sum of all edge lengths.

Steiner points are defined as a separate set of nodes $S \notin V$, in Euclidean solution space. The completed graph using Steiner points is $G = (V \cup S, E)$. There is also a form, where both the existing vertices V and the Steiner points S are given and the STP can be formulated as a NP-Complete decision problem. For the agent node locating problem, the Euclidean space problem is considered.

Figure 2.2 shows a sample graph with V represented by solid circles. In cases with two vertices, the shortest path between them is a straight line; they can be connected directly. For three (or more) vertices, to connect them all, MST solution methods provide a simple connected network solution. The Steiner minimal tree uses an additional point in S, that minimizes the total network edge length.



(a) (b) (c)

Figure 2.2: Graphs showing a.) Linear points b.) Minimum Spanning Tree c.) Steiner Minimum Tree

There is an upper limit to the number of vertices that a Steiner point can connect. Lin and Xue [40] state that this limit is five in a bounded edge length problem. This will be expounded upon in the following section.

2.3.3 STP with Minimum Number of Steiner Points and Bounded Edge Length

The addition of the bounded edge length assumes that each node has a physical, limited radius of connectivity that may not always be connected as the MST and nodal graphs would imply. If in an MST solution two vertices are connected, but if the edge length between the two nodes is larger than the connection radius, then the nodes can not be physically connected. To resolve this, Lin and Xue [40] prove that the Steiner tree problem with minimum number of Steiner points and bounded edge-length, or STP-MSPBEL, is NP-Complete and develop a polynomial time approximation algorithm. This Minimum Spanning Tree Heuristic derives a worst-case performance ratio of at most five times the total length that of the optimal solution. This heuristic is an approximation that uses Kruskal's algorithm to obtain the MST solution. It then considers each edge of the MST solution, if greater than the edge bound, and replaces the edge with multiple nodes of degree two. Figure 2.2 is modified to illustrate the additional number of nodes required in BEL cases shown in Figure 2.3.

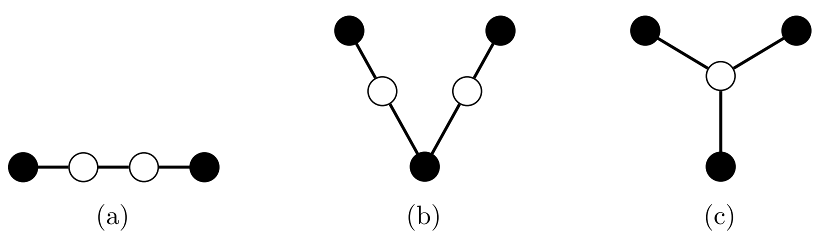


Figure 2.3: Bounded edge length graphs showing a.) Linear points b.) Minimum Spanning Tree c.) Steiner Minimum Tree

In general for the MST-BEL, the node degree upper limit of five can be shown by examining a set of vertices in a regular hexagon inscribed in a circle of radius r, with the bound on edge length equal to r, having all edge lengths equal to the circle's radius. See

Figure 2.4. The bounded edge length connected network, without Steiner points is defined as G. The length and cardinality of G are defined as $L(G)$ and $|G|$. For the given network $L(G) = 6 \times r$ and $|G| = 2$.

The SMT solution, G' places a Steiner point at the center of the circle, Figure 2.4b. Assume that the edges of G' are only those connecting vertices to the Steiner point, thus, $|G'| = 6$ and $L|G'| = 6 \times r$. In such a case, Lin and Xue state that the maximum $|G'|$, is five. Any edge connecting a vertex and Steiner point can be moved to connect two adjacent nodes, creating a new graph, G''. The resultant network is degenerate, in that the lengths are equal but cardinality is reduced, Figure 2.4c.

This upper bound was later disputed by Măndoiu and Zelikovsky [47] and then Chen *et al.* [10]. Măndoiu shows that the upper bound of the cardinality of Steiner points can be reduced to four, Figure 2.4d.

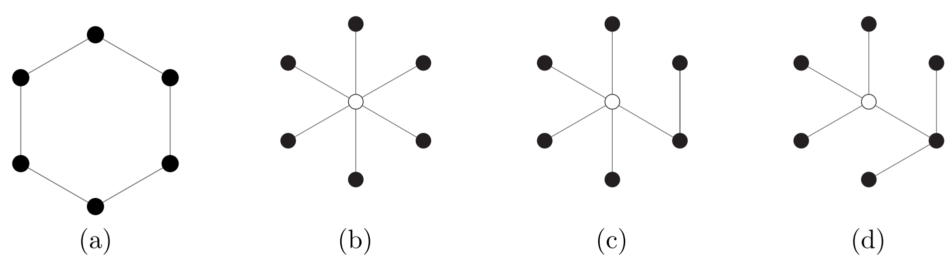


Figure 2.4: Degrees of Steiner points a.) Hexagonal graph G b.) SMT solution $|G'| = 6$ c.) SMT solution $|G''| = 5$ d.) SMT solution $|G^{*}| = 4$

The Euclidean Steiner Minimal Tree (ESMT) is NP-Complete in complexity [23]. Thus, solving larger problems is significantly difficult. Comparatively, the MST can offer decent solutions for very small computation time, generally $\mathcal{O}(m \log n)$ [81]. Gilbert and Pollak [24] take advantage of their definition of Steiner points to enumerate all combinations of vertices and resultant Steiner points using branch and bound methods to prune sub-optimal solutions.

The STP alone is not enough to represent the MANET agent node placement problem. It has an accurate representation of using additional nodes to connect a network while minimizing total network length, but does not consider radio connection radius limitations and limits connectivity of nodes in V to S to three. For this research, there is the important requirement of minimizing the number of Steiner points (agent nodes).

Kruskal's algorithm requires that as a solution is being constructed, no added edge creates a cycle. Cycles are counter-productive in an MST. For this research, this constraint must be relaxed. An addition of a node may create a local cycle due to the closeness of nodes, but cycles are not detrimental.

Detailed later (Section 4.1), groups of user nodes can be connected using fewer nodes by placing an agent nodes at the centroid. Group determination and centroid placement must be added to Lin and Xue's MST Heuristic.

Chen *et al.* [10] consider the occurrence of groups. First, given the maximum connection radius r, if there are nodes that can be connected without additional nodes, they are connected. Then, for groups of three or four nodes that can all be connected by adding a node interior to the group, a Steiner point is added at that location. Lastly, Steiner points are added to connect previously connected sub-graphs to the rest of the network.

Connectible nodes, then groups, then the remaining subnetworks are connected in that order. With the exclusion of the size of the group of nodes considered for interior Steiner point insertion, the work of Chen *et al.* is very similar to the MMST methodology developed in this dissertation.

A variation of the formulations of the STP is given by Goemans and Myung [25], specifically, a rooted STP, creating a Steiner arborescence. A Steiner arborescence has one node that all nodes proceed from, like a tournament tree. In a military (or search and rescue) framework, a control node is defined, and could be interpreted to be the root node of a Steiner arborescence. It is desired for all other nodes, agent and user, to have a direct or network connection to the control node. Goemans and Myung reviewed these problems in both digraphs and graph contexts, noting that the specification of the root node is unimportant. This is beneficial and logical in that if all nodes are connected, then root nodes do not need to be specified.

Srinivas *et al.* [59] use a two-stage method to solve the Mobile Backbone Network problem. Phase 1 finds a minimum backbone node solution to connect all regular nodes, ensuring that each is connected to at least one backbone node. The second phase is the STP-MSP problem to connect the remainder of the network. Though both static and mobile regular node networks were evaluated, the requirement that each regular node be connected to a backbone node differentiates this dissertation from that of Srinivas et al.

The STP-MSPBEL, as described by Lin and Xue [40] is the same problem as locating agent nodes in the MANET scenario. Both instances want to minimize the number of agents or Steiner points needed. Lin and Xue's added edge-length constraint simulates connection distance limitations. Both Gilbert and Pollak [24], and Lin and Xue [40] use Steiner points with maximum degree of three. Here, this constraint can be relaxed, allowing any number of nodes to be connected to a Steiner node. And, as in Srinivas's work, there is not a requirement for each user node to be connected to an agent node. The MANET system requires that all nodes be connected to a control node, similar to a Steiner arborescence problem. Goemans and Myung [25] show that specification of a root node is not required, so specification of the MANET network as a Steiner arborescence problem is unnecessary. Lastly, placement of added Steiner points is not limited to only the edges of the MST as in Lin and Xue's work.

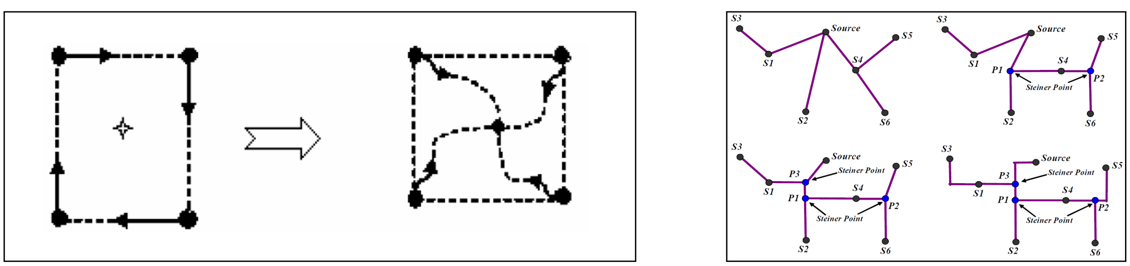
Extending the works of Gilbert and Pollak, and Lin and Xue, this dissertation will deviate from Kruskal's Algorithm for MST via relaxation of the cycle and edge constraints, elimination of node degree constraints, addition of group and centroid placement, and allowance of an unconnected graph as input. As a solution method for the STP-MSPBEL, similar to Chakraverty *et al.* [9] and Yan *et al.* [84], the Dynamic Modified Minimum Spanning Tree (DMMST) heuristic is presented in section 4.2.

2.3.4 Dynamic Node Location Methods

One approach to solve the STP-MSPBEL is to use moving (Steiner) points. Chakraverty *et al.* [9] implement a process where Players (moving points) are initially co-located at each of the unconnected nodes, moving towards adjacent Players, converging to a common Steiner tree backbone (see Figure 2.5a). Unlike this dissertation, the methodology of Chakraverty *et al.* is implemented for graph networks where candidate Steiner point locations are included in the problem formulation.

Yan *et al.* [84] also creates a process for rectilinear space that initializes a network with hidden Steiner points at existing network nodes. As the algorithm progresses, potential Steiner locations are computed based on interconnection timing delays per arc until feasible Steiner point locations are found. The problem and solution method of Yan *et al.* works well for problems like chip design, often defined in rectilinear space. The work herein is in Euclidean space, and does not consider timing delays from a source to sink nodes.

In this research, a moving node method is also developed, expounded upon in Section 4.2. Here, however, the point's movements are based on the MMST solution.



(a) (b)

Figure 2.5: Moving point methods for the static problem a.) Chakerverty's players chasing their adjacent players leading to convergence at centroid b.) Yan's calculation of Steiner points

2.4 Chapter Summary

A large part of the MANET literature is based on connectivity, clustering, positioning, and routing for improved network connectivity. The existing path planning research is done on nodal networks with tasks at nodes (such as attack or survey). Most of these methods use distributed, not centralized, logic and are reactive, assuming no knowledge of the movement of user nodes. They also assume an unplanned stochastic environment. This is assumed because any practical applications would be in a stochastic environment. However, the assumption that neither the control nor user nodes have any prior information about their tours is not. There arises a need to address foreknowledge of user node movement when path planning for agent nodes. It is more reasonable to assume that user node movement is known a priori but that there is a possibility of encountering path deviating uncertainties.

Here, using pre-deployment planning is leveraged to improve the performance of the MANET. This will be done by solving a series of continuous space static Minimum Steiner Tree Bounded Edge Length Minimum Steiner Point (MST-BELMSP) problems. This will be done assuming that the measure of node to node connection is binary (based on a connection radius as opposed to bandwidth based). With the inclusion of a bandwidth calculation, a constraint on the bandwidth of links would be needed. Since bandwidth is based on distance, a link bandwidth constraint can be generalized to be just a distance constraint.

Additionally, for most cases, all node types are assumed to be homogeneous in technical capabilities. This is in regard to the movement rate and connection radius. With heterogeneous connectivity radii, a connection between two nodes to enable two-way communication is dependent on the minimum radii. This alternate assumption could be incorporated but would require proper definition of one-way communication assumptions for such an adaptation. In regard to movement rate, assume uniform movement rates for simplicity, though this can easily be modified to accommodate multiple movement rates.

Lastly, also for simplicity, only consider one static control node. This dissertation would still be applicable if movement of the control node is allowed. Using multiple control nodes would add complications about which control node the other nodes need to be connected to. As is the case in the Steiner arborescence, the specification of root nodes is unimportant.

Chapter 3

Mobile Ad-hoc Network Model

Detailed in this chapter is the general framework for modeling the MANET including node types and their functions. The graph form for which a problem instance represented is presented. In addition, the problem sizes are defined.

3.1 Node Types

A node represents the individual or group in the system. There are three basic types of nodes used in the MANET model: control, user, and agent. A node can be only one of these defined types. Assume node locations are defined only in two dimensions. Each node at time t is represented as $n_{ti} =< x, y, k >$, a vector representing the location and type of the i^{th} node where

x, y	location coordinates $\{(x, y) : x, y \in \mathbb{R}^2\}$
k	type of node $\{$'c', 'u', 'a'$\}$

and node type sets are defined as

C_t	$\{n_i \| n_i[k] = \text{'}c\text{'}\}$, control nodes
U_t	$\{n_i \| n_i[k] = \text{'}u\text{'}\}$, user nodes
A_t	$\{n_i \| n_i[k] = \text{'}a\text{'}\}$, agent nodes

The set of nodes containing all node types is defined as $N_t = C_t \cup U_t \cup A_t$.

Time increments, represented by t, describe the discrete intervals of the MANET model's progression. The MANET model starts at $t = 0$ with user nodes moving from their initial location, usually the control node, and ends once all user nodes complete their tours at $t = T$, In a real world environment, time is continuous and nodes are continually moving. Define the actual time between increments $\tau \in \mathbb{R}$ so that the increments of t are representative of $t \times \tau$ actual elapsed time. As part of the model design, this value would be the maximum amount of time the system could operate without updating node locations. The continuous movement of nodes between each time step is not modeled; the result of the movement is modeled by a node's location at each t. Practically, this means that nodes would report their location, send/receive commands at time $t \times \tau$. There are several other complexities associated with managing a complex network such as message propagation time, bandwidth limits, energy requirements that will not be considered in this model. As such, assume connectivity at a discrete time t is sufficient to send/receive network information. It is assumed that the movement between time steps is linear.

There are also checkpoints, generally defined as tuples (t, x, y). Each tuple represents a (x, y) location at time t. Assume that checkpoints are only defined at the discrete time intervals of t. All nodes can be specified to move along a path defined by a set of checkpoints, here however, only user and agent nodes are mobile. The control nodes are fixed. This is to differentiate the capabilities and responsibilities of node types. The checkpoint subsets, termed also as the user and agent node tours, are represented as M and Q will be detailed later. Figure 3.1 shows how the nodes are graphically represented in the MANET models. Each of these is detailed in the following sections.

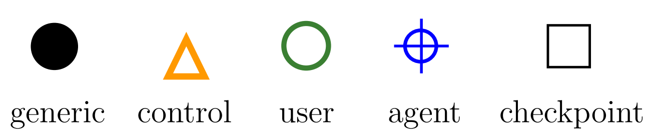


Figure 3.1: Node types

All nodes have a similar base functionality to move, similar to user nodes detailed in Section 3.1.2. In addition each node type has specific rules, dependent on the node's state, network status, and passed commands that will determine how it will move. If the desired movement location is too far, then the node will move in the direction of the desired location, bound by a predetermined maximum move rate, $v \in \mathbb{R}$.

As nodes move through the field of operation, the connections between nodes, E_t, are updated. For a given network configuration, the control node can determine which nodes it can communicate with by a depth-first tree search.

3.1.1 Control Node

The control node is the head of the system's operation. It is assumed that the commanding officers, leaders, or decision makers of the operation are stationed here. Mission information is created, distributed, and held at this node. The plans for node movement, number of units deployed, and real-time mission updates come from the control node. For this reason, it is assumed that all nodes must be either directly or network connected to the control node to receive instructions from and to send status updates back to the control node. For this dissertation, assume that there is only one control node, $|C_t| = 1$.

In both deterministic and stochastic cases, user nodes will attempt to follow the planned tour from the control node but will operate independently. Thus, in either case, the control node will manage agent node movement. Movement plans for agent nodes are derived from the methods of this dissertation contained in Chapter 5.

Update Network State

During the deployment of a MANET, for each time step, the control node updates its internal model of the system. If there is a path between the control node and any other node j, then the status and exact location of node j are transmitted back to the control node. In a deterministic case, regardless of the network connectivity state, the control node's internal model always represents where the user node will be at any time t. This is because each user node will follow the tour given by the control node. This differs from Cho's [12] work in that user nodes' movements were not deterministic and the future predictions had to be derived from historical data.

For operation of the MANET in a stochastic environment, the control node will use the internal model as an approximation of the actual network state as all information may not be available to the control node in the cases of an unconnected network state. If there is not a path from a user node to a control node, then the control node attempts to use the planned and most recent movement behavior of the user node to estimate its current location. Similarly, for an agent node, if there is not a path to the control node, then the control node uses the agent node's recent movement history, movement logic, and the commands that were previously given to estimate its location. The following set of control node functions and abilities are presented to be used primarily in the stochastic environment.

Forecast User Node Locations

The internal model is used to forecast user node locations s time steps in the future. The forecast method is dependent on the user node's adherence or deviation from its tour and its connectivity.

If the user node is connected and has not shown signs of deviation within the previous $t-s$ to t time steps, the internal model user node location will be updated based on continued movement according to its tour. It is possible for a user node to transition to a deviating state and remain connected to the network. Such a case is preferred as the information about its deviation can be communicated to the control node. Thus the control node will obtain the deviating user node's modified trajectory and forecast accordingly.

If a user node is disconnected but its most recent history does not indicate any deviation from its tour, the estimate of its location would be based on adherence to the tour. Similarly, if the most recent history indicates deviation, the estimate of its location will be based on continuing with the deviation trajectory. When a user node deviates or returns to its tour (from deviating) while disconnected, the control node will not have knowledge of the state change until the user node is reconnected to the network. If its deviation state changes while disconnected, the control node compares the user nodes expected location (based on the internal model) with the network state to infer the user node's state. Else, the control node will have to assume that the user node is either still deviating or continuing with its tour based on its state when last connected to the network.

The difficulty of determining these state changes is dependent on the frequency of network status updates, τ. With smaller values, deviations per time step would be small and would potentially grow with larger values.

User node locations are forecast using double exponential smoothing as was found in Konak *et al.* [35]. The control node maintains a history of all user node locations and, using the previous three time steps (and current location), estimates locations at time $t+s$ using:

$$U_{t+s,i}[\hat{x}] = U_{ti}[\bar{x}] + sv^{U_{ti}[x]} \tag{3.1}$$

$$U_{t+s,i}[\hat{y}] = U_{ti}[\bar{y}] + sv^{U_{ti}[y]} \tag{3.2}$$

where $U_{t+s,i}[\hat{x}]$ is the estimate of user node i's x coordinate location at the future time $t+s$, $U_{ti}[\bar{x}]$ is the moving average of the x coordinate and $v^{U_{ti}[x]}$ is the estimated velocity. The moving average of the location x coordinate and the velocity estimates are modified by

$$U_{ti}[\bar{x}] = \alpha_1 U_{ti}[x] + (1-\alpha_1)U_{t-1,i}[\bar{x}] \tag{3.3}$$

$$v^{U_{ti}[x]} = \alpha_2(U_{ti}[\bar{x}] - U_{t-1,i}[\bar{x}]) + (1-\alpha_2)v^{U_{t-1,i}[x]} \tag{3.4}$$

where α_1 and α_2 are the smoothing parameters that bias the impact of new (or historical) data. The values for the y coordinate are found similarly.

Measure Network Disruptions

User node deviations as a function of their movement are presented in Section 3.1.2. For the control node, with its knowledge and assumptions (about undetermined unconnected user node locations) about the system must determine how well the user nodes adhere to their planned tours.

Determine Agent Node Locations

The control node will evaluate the network at time $t + s$, using user node location forecasts, to decide if the agent node assigned tours need to be modified. Based on the magnitude of network deviation the control node's actions are either continue with tours or re-plan, defined next.

Continue Tour

Allowing a minimal level of network disruption, assuming the assignment plans are still valid, the agent nodes will proceed based on the existing tours. How this is performed in a physical solution is based on the amount of autonomy the agent node is capable of. If remotely controlled from the control node, the agent nodes will be commanded through the network. Otherwise, the agent nodes will move according to their tour until commanded by the control node.

Re-Plan

With large disruptions that significantly affect network connectivity the control node can decide to re-solve to determine new tours. This would imply that the previously assigned agent node tours are no longer effective. The threshold for determining to re-solve balances the quality of the network connectivity with computational effort to determine new agent node tours. If this value is too low, new plans will be needed often. If it is too high, then the average network connectivity will likely suffer.

Re-solving would not be beneficial if the computation time for a new solution (pre-solve and assignment) is longer than the real time network update interval, τ. If this is true, then the new solution would be determined after the time step in which it was needed. These cases are likely to occur in either a very large network (large computation time) or when τ approaches zero. Here, assume unlimited computation power where the computation time is significantly less than τ. The new agent node assignment replaces tours from time t to T using the re-plan solution.

Reactive

A reactive approach is incorporated in the control node decision process. At a defined frequency or each time step, the control node can implement any of the static locating solution methods presented in Section 5. The locations of the connecting points in the solution are then sent to connected agent nodes. The reactive positioning method is also used to compare the effectiveness of using the pre-plan and assignment method. This is in contrast to the meta-heuristics, particle swarm and genetic algorithm, used in previous works.

Send Locations

The new locations will be sent to all agent nodes that are network connected to the control node. If an agent node is not network connected, it will continue following its assigned tour. If it is disconnected and there are no new checkpoints on its tour to move toward, then the agent node will return to the control node.

3.1.2 User Node

Representing individuals such as search and rescue responders or infantry, these nodes are the end users of the system and often the leaf nodes of the connection tree. This set, U_t, is assumed to be homogeneous in type. Receiving a predetermined plan from the control node, these nodes can proceed to carry out missions without further instruction or without a connectivity requirement.

Tour and Movement

During the planning stage, the control node defines the user node tours. This is to represent a mission planner giving the users their tasks. This could be to survey a given area, move to a new location, or complete a tour of points. Movement patterns are specified in the following sections as they are dependent on the problem size and context. User node tours are the instructions dictated by the control node and are represented by $M_{tn} = \{(t, x, y) : t \in [0, T], x, y \in \mathbb{R}^2, \text{ the location } (x, y)\}$ for node n at time t.

User Tour Definition - Polar

Here, a random walk method is used to define the user node tour. For each user node, the first checkpoint was at the control node's location. Each node's tour will have a length, L, or a number of checkpoints. This would include initial and final locations. If it is required for the user node to return to the control node after tour completion, then the number of randomly determined locations is $L-2$, else the number of random locations is $L-1$. Each additional checkpoint is added with some level of angular variation, v and some distance, S:

$$M_{tn} = (t, M_{t-1,n}[x] + \cos(\theta + \delta \mathrm{R}())S,\ M_{t-1,n}[y] + \sin(\theta + \delta \mathrm{R}())S, \mathrm{R}()v) \tag{3.5}$$

M_{tn}	Location at time $t \in [0, L]$ for user node $n \in U_t$
δ	Variation between checkpoints
S	Steps between checkpoint
$\mathrm{R}()$	A random number generator $\in (0, 1)$
θ	Initial angle of movement

The combination of the magnitude of S and, even more, the magnitude of δ directly determines the dispersion of the network. For example, if $\delta = 0$ all nodes would travel in rays away from the control node at a distance of S between checkpoints. Increasing δ results in increased randomness in the direction of successive checkpoints, with a distance of S between them. Appendix B elaborates on a few of these parameters. During the planning stage or operation of a MANET, the node would go from checkpoint to checkpoint until all have been visited.

User Tour Definition - Random

As an alternative to disseminating from the control node, a random start within the field of operation and successive locations determined in a polar method define a random tour. The initial checkpoint is defined by $M_{0n} = (\mathrm{R}()W, \mathrm{R}()H)$ where R() is the random number generator, W is the field of operations width and H the height.

User Tour Definition - Context Specific

The U.S. Army Field Manuals specify movement formation and patterns for several types of units as well as different circumstances. Those of interest are the patrol and travel patterns defined by [69, 67, 73]. The patrol patterns require a user node to leave the control node and travel an elliptical tour of an area near the control node. The travel pattern moves user nodes from one side of the area of operation to the other.

For the SAR (search and rescue) context, the Australian Coast Guard Field Manual [80] details methods of defining search areas based on topographic maps and theoretical or statistical travel distances of lost persons. It specifies move patterns for search teams including the Track Sweep or Point and Flank patterns. The manual does not give insight into the macro coordination of movement of all search teams. Each user node is assigned an area to search.

Though there is randomness in determining these points, during a deterministic case, the user nodes will follow them exactly. In the stochastic case, there would be another mechanism to determine the randomness of a user nodes movement (or the deviation from the defined tour).

Stochastic Environment Deviations

In the stochastic environment it is assumed that user nodes travel in a straight line to successive checkpoints with probabilistic deviations. Practically, this could be a blockage of a previously available pathway or some adverse entity. Cho [12] considered the Random Waypoint Model (RWM), as well as two directed movement plans, the Convoy and Defense (CD) and the Search and Rescue (SR) methods for user node movement. The latter two methods have points, similar those defined as checkpoints in this dissertation, to move towards. Each of Cho's methods considered some degree of perturbation in user node movements to simulate real movement, not a straight line between waypoints. Random deviations from a user node's tour, regardless of the cause, will define the uncertainty of the stochastic case. The resultant user node tour with stochastic deviations will be referred to as a "realized" tour.

For any simulation environment, at time $t = 0$, all user nodes will proceed with their planned tour and have no deviation scheduled. At any time step, $t > 0$, there is a chance of deviating. A user node can be scheduled for only one deviation at a time.

Define β^{max} as the vector of the maximum (minimum) deviation parameter's angle, percent velocity modification, deviation duration, and cool-down time. The probability of deviation is held constant at 10%.

$$\beta^{max} =< \theta^{max}, v^{max}, \eta^{max}, c^{max} > \tag{3.6}$$

Define $\hat{\mathcal{N}}$ as the Gaussian distribution based random number sample from

$$\mathcal{N}(\mu, \sigma) = \mu + \sigma * \sqrt{-2 * \log(\mathrm{R}())} * \sin(2 * \pi * \mathrm{R}()) \tag{3.7}$$

Define β_{tn}, the Gaussian random number transformed parameters used to modify user node movement.

$$\beta_{tn} = < \hat{\mathcal{N}} + \theta^{max}, \hat{\mathcal{N}} v^{max}, t + \frac{\eta^{max}}{2} + \lceil \frac{\hat{\mathcal{N}} \eta^{max}}{4} \rceil, \beta_{tn}[\eta] + \frac{c^{max}}{2} + \lceil \frac{\hat{\mathcal{N}} c^{max}}{4} \rceil > \tag{3.8}$$

This vector β_{tn} specifies at time t, user node n would move with an angle modification within the range of $(-\beta^{max}[\theta], \beta^{max}[\theta])$ added to the current trajectory θ_{tnm} (the angle between node n and its next checkpoint m) with additional variation of $\beta_{tn}[\theta]$ at each successive time step until time $\beta_{tn}[c]$. A velocity deviation could be an increase or decrease to move rate, $v + \beta_{tn}[v]$. The the Gaussian random deviation and cool-down values are converted to completion times. To ensure only positive values, the deviation mean is defined as $\frac{\eta^{max}}{2}$. Similarly the maximum increment/decrement for transforming $\hat{\mathcal{N}}$ to deviation duration is defined as $\frac{\eta^{max}}{4}$ and likewise for cool-down duration ($\frac{c^{max}}{2}$). From time $\beta_{tn}[n]$ to $\beta_{tn}[c]$ no other deviation can be scheduled as the user node returns to a point in its tour with additional variation of $\hat{\mathcal{N}} \frac{\pi}{8}$ at each successive time step.

Heuristic 1: Deviation Mechanism

if $R() < 0.1$ *& not deviating & after cool-down* **then**
    get new deviation
if *deviating* **then**
    modify movement by θ + variability
else
    determine best checkpoint to continue to
    continue using tour

For a deviating user node u, determining the best checkpoint to continue the tour is a function of the current time and candidate checkpoints and their associated parameters (x, y location and time). At time t each checkpoint m of its tour where $m[t] > t$ is evaluated. Determining the best to continue to is based on a comparison of the distance between the user node and candidate checkpoint $d_{u,m}$ and the ratio $\frac{d_{u,m}}{u[v]*(m[t]-1)}$. This proportion provides a means for determining the feasibility of continuing the tour to checkpoint m based on the time to get to m.

3.1.3 Agent Node

The set A_t is used to bridge the network connections among the control node, connected nodes, and unconnected nodes. Any type of platform that is able to transmit and receive network signals can be used as an agent node. Here, unmanned vehicles such as Unmanned Aerial Vehicles (UAVs) or Unmanned Ground Vehicles (UGVs) are to be used to bridge these connections. It is assumed agent nodes are semi-autonomous.

The movement rate of these nodes should be comparable to user nodes. It is possible that agent nodes could have significantly higher movement rates (due to the ability to fly UAVs) or equal (UGVs) to user nodes. The tour for agent nodes is defined as a collection of checkpoint locations $Q_{tn} = \{(t, x, y) : t \in [0, T], x, y \in \mathbb{R}^2, \}$, the decision variable matrix representing agent node tours. This is the (x, y) location of agent node n at time t and is this dissertation's principal set of decision variable. The control node's planning defines Q_{tn} for all agent nodes at all times used to ensure network connectivity over the course of a MANET's operation. In the deterministic case, Q_{tn} is followed explicitly. In the stochastic case it may be necessary to re-evaluate Q_{tn} if any uncertainty invalidates the planned Q_{tn}.

This dissertation will determine Q_{tn} for deterministic cases and, in stochastic cases, provide a means of re-evaluation of successive values of Q_{tn} to maintain network connectivity over the course of a MANET's operation.

3.1.4 Graphs

Define a graph $G_t = (N_t, E_t)$ for $t \in [0, T]$ that represents all nodes and the connecting edges where $e_{tij} \in E_t$ is the edge at time t defined by $\{(n_{ti}, n_{tj}) \in N_t | d_{ij} < r\}$. The distance between locations i and j is defined as d_{ij} and the maximum connectivity distance is r.

Two nodes i and j, regardless of type, are connected if $\exists\, e_{tij} \in E_t$. This requires each node to be within the radius of connectivity of the other node. If all nodes have identical connection radii, r, then the maximum distance at which the two connected nodes can be from each other is r. This representation is similar to unit disk graph. Any distance greater than r but less than $2 \times r$ can be bridged by an additional node. Any distance greater than $2 \times r$ must be bridged by multiple nodes. Though not allowed in this dissertation, nodes with different size radii must be within a distance less than the smaller of the radii to enable two-way communication. Figure 3.2 illustrates the modeled connection types.

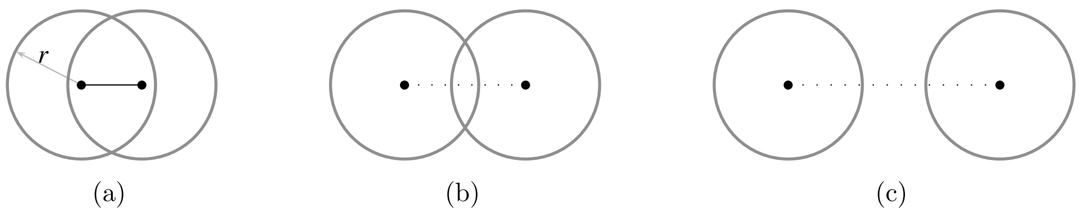


Figure 3.2: Pairs of nodes showing a.) connected nodes b.) overlapping coverage c.) distant nodes

If a network is not connected, additional nodes are needed. The added set of locations where nodes can be placed to connect G_t is defined as $P_t = \{(x, y) : x, y \in \mathbb{R}\}$. The collection of points P_t identifies locations that are beneficial for an agent node placement. P_t is not an assignment for any specific agent node.

An acyclic path between nodes and/or points i and j will be defined by $H_{ij} = \{e_{til} = (n_{ti}, n_{tl}), ..., e_{tlm} = (n_{tl}, n_{tm}), ..., e_{tmj} = (n_{tm}, n_{tj}) | e_{til}, e_{tlm}, e_{tmj} \in E_t; n_{ti}, n_{tj}, n_{tl}, n_{tm} \in N_t \cup P_t\}$. If $e_{tij} \in E_t$ then $H_{tij} = \{e_{tij}\}$. If a node is connected to the control node via the network, the existence of a path H_{ij} for $i \in C_t, j \in N_t \setminus C_t$, will indicate this.

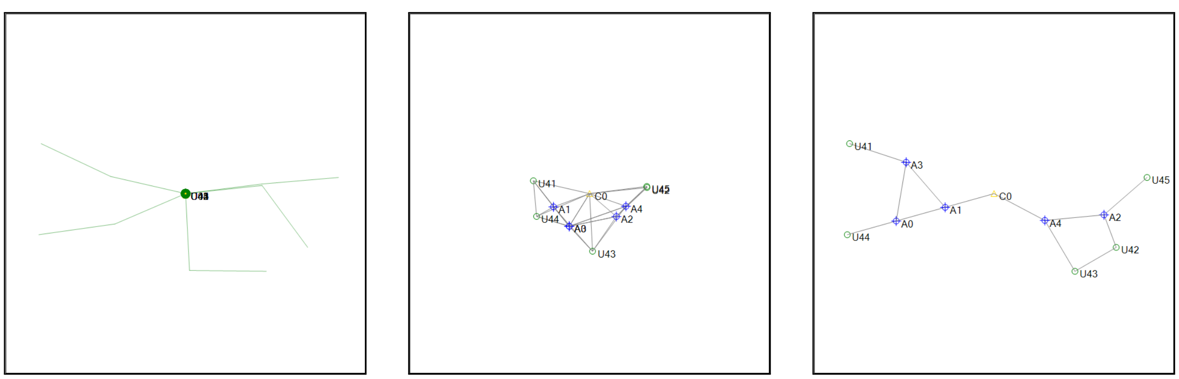
Define the connected network at time $t \in [0, T]$ where there exists a path H_{tij} for an $i \in C_t$ and all $j \in U_t$ is defined as $G_t^* = (N_t \cup P_t, E_t)$. If the path H_{ij} exists for $i \in C_t$ and $j \in N_t \setminus C_t$ then a node is connected via the network to the control node. If all nodes are network connected to every other node, then the network is also a connected graph. Any alteration in a connected network that would result in one or more nodes to not be network connected would disqualify the network from being fully connected.

3.2 Operational Context

The node types that were previously detailed will play a part in the operation of MANET scenarios. In a scenario, there is a single control node and each of the user and agent node's initial locations are defined by the tour type. After the MANET model is initiated, user and agent nodes move according to their tours.

As the user nodes carry out their missions, they may become disconnected from the network. Moving beyond the range of communication with other nodes would cause this. If a connection is broken and results in disconnecting any number of nodes from the rest of the network, then the network state changes to unconnected. Adding agent nodes can help to bridge the connection.

Figure 3.3 shows how the network evolves as user nodes move towards their checkpoints. Specifically, Figure 3.3a shows the deterministic user node tours. As they move in this direction, the agent nodes will be deployed from the control node to maintain network connectivity, Figures 3.3b and 3.3c.



(a) (b) (c)

Figure 3.3: Network states a.) Given user node tours b.) Intermediate connected network c.) Final connected network

In the MANET models, communication is important and sometimes even vital. It is necessary that the control node be able to obtain the locations and status of each of its deployed user and agent nodes. In the event that any of the nodes encounter some uncertainty, the control node should be relayed this information. Other tactical information can also be transmitted to manage the future operation and maintain connectivity.

3.3 Experimentation Design

To evaluate the effectiveness of the positioning methods developed herein, both static and dynamic problems will be used. These problems will be generated based on either a military or a search and rescue (SAR) context. The different contexts are used to determine problem size (as a factor of the number of nodes and the area of the field of operation) as well as to illustrate the applicability of the developed methods.

In any problem, there are user nodes that move to carry out a mission taking them further than the range of radio connectivity with the control node. Realistically, the user nodes will always be moving and so would the deployed agent nodes assigned to maintain network connectivity.

3.3.1 Military Context Parameters

Basic military unit sizes are listed in Table 1.1 and provide a basis to define the number of user nodes a problem would require. With the definition of each unit size and their described interoperability [68], assume that each soldier does not need to be modeled as a user node. For example, a fireteam consists of two to four soldiers that work as one unit. Also assume that all soldiers within a unit move together and are always within range of direct communication and would not require radios. With this, for platoon to company size problems, define one user node to represent a fireteam. Similarly, when evaluating larger units, battalion size or greater, redefine the user node to be a squad based on similar assumptions of proximity of sub-units and uniform movement. Table 3.1 lists the number of user nodes required for problem size definitions based on U.S. Army unit sizes [68]. In it, the abbreviations F, S, P, C, represent the fireteam, squad, platoon, and company military units.

Table 3.1: Military Problem Size

Class	Simulation User	$\lvert U_t \rvert$	Number of Soldiers	Area (km x km)
Platoon	Fireteam	5 S x 2 F = 10	45	128 x 128
Company3	Fireteam	3 P x 5 S x 2 F = 30	135	256 x 128
Company6	Fireteam	6 P x 5 S x 2 F = 60	270	256 x 128
Battalion2	Squad	2 C x 6 P x 5 S = 60	540	256 x 256

The area of operation was derived by extrapolating from several assumptions of soldier movement rates, number of soldiers and duration of missions. From U.S. Army FM 3-21.20 [65], movement rates for foot marches during different conditions are specified and listed in Table 3.2. The average travel distance per day is stated to be 20 to 32 kilometers, marching from five to eight hours at a rate of four kilometers per hour, with an absolute maximum of 56 kilometers in 24 hours during a forced march.

Table 3.2: Dismounted Rates of March (normal terrain)

	Roads	Cross-Country
Day	4.0kph	2.4kph
Night	3.2kph	1.6kph

The assumption was made that movement is during day time on a mix of roads and cross-country travel, averaging the rate of travel to be 3.2 kph resulting in a per day move rate of 16 to 25.6 kilometers for a five to eight hour march. Assume this reduction to the maximum move rate will also take into consideration pauses or breaks for any reason. It will also consider the fact that movement is the coordination of many sub-units where synchronization of movements may reduce move rate.

If it can be assumed that each problem is over the span of a maximum of five days and all user nodes originate at the control node and are allowed to move in any direction $(\pm x, y)$ at the above specified rate of 25.6 kilometers per day, define the maximum area of operation to be $(2 \times 5 \times 25.6)^2 = 65,536$ square kilometers.

Assume that this maximum area is applicable for a foot march of a large Battalion. Assume also that each sub-unit will have a smaller maximum area of operation. The areas for each size class are defined in Table 3.1.

These values were defined for the move rate of the missions used here. These are only parameters and can be modified to fit alternate missions. Should the mission call for mounted/motorized units, the maximum move rates could be adjusted to reflect a faster movement of nodes. In addition, the maximum mission distance would also be modified.

3.3.2 Search and Rescue (SAR) Context Parameters

For large magnitude natural disasters hurricanes, earthquakes, and tsunamis are considered. Generally, the impact of such events is most drastically pronounced in densely populated areas such as major cities. In 2005 Hurricane Katrina brought devastation to New Orleans, Louisiana and in 2008 Hurricane Ike made landfall in Galveston, Texas. The resultant tsunami that affected the surrounding countries after the 2004 Indian Ocean earthquake decimated the capital city, Banda Aceh, of the Aceh province in Thailand. Natural disasters like these have brought destruction to each country's infrastructure resulting in challenging SAR operations.

Though such events can be large enough to affect entire countries (depending on size), such large scales were not considered. The largest problem defined in the SAR context was that of a city sized search. This will allow generalization and categorization of the surface area needed to cover during SAR operations.

Man-made disasters usually do not have the same effect on such a large area as natural disasters, but are rather concentrated. As an example, the September 11th terrorist attacks in 2001 was a focused attack that, along with the thousands of deaths, impaired the communication system that first responders relied on for status updates. The communications range of first responser's radios was diminished when used inside and around buildings [21]. The area associated with this type of response and probable number of needed user nodes was generalized to be a small SAR problem. Table 3.3 lists the land areas of the mentioned cities and event locations [72]. Auburn, AL is listed as a point of reference.

Table 3.3: Land Areas of Affected Cities

City	**Area** (km^2)
Auburn, Alabama	140.80
Banda Aceh, Thailand	61.36
Galveston, Texas	106.75
New Orleans, Louisiana	438.80
World Trade Center (NY)	1.00

To classify SAR areas, assume a city is defined as an area larger than 300 square kilometers. Based on the definition of a city block to be a 175 x 175 meter area [69], the successively smaller areas are defined as fractions of the larger based on the size of a city block. The smallest area is defined as a neighborhood, similar in size to the World Trade Center example in Table 3.3.

To specify the number of user nodes, assume a search team consists of three to five people [80] and define a user node to be one search team. Defining the search area for a single user node to be approximately one square kilometer and assuming a move rate of 3.2 kph or slower (to allow for visual scanning) [65], would roughly equate to a 6 x 6 block area or 7.35 kilometers of travel. At the defined maximum SAR move rate, this would be 2.3 hours of travel. Adding miscellaneous time for breaks, a single SAR shift can defined as three hours. This would be a reasonable amount of work, as searchers may be volunteers and environmental conditions could be taxing. Also, because the search area could be very large, and have search teams composed of volunteers, assume that only a portion of the total required number of search teams are available. For this dissertation, arbitrarily consider only sixty percent of needed users nodes are present.

For a city size search (greater than or equal to 300 km^2), at least 300 user nodes are needed. This would be considered a very large problem dependent on the search area. Limitations on the largeness of the problem are the reasonable size of searching or the number of user nodes that can be simulated in a timely manner. That is to say, extending searches beyond a city would require coordinating mechanisms that would superceed the focus of this research. Additionally, the computational effort required to do the simple tree search for network connectivity would greatly increase the computation time required for a solution.

Table 3.4: SAR Problem Size

Class	$\lvert U_t \rvert$	**Area** $(km \times km)$	**Note**
Neighborhood	9	4×4	6×6 (175m) blocks
District	41	8×8	$7 \times$ Neighborhood
Region	141	16×16	$4 \times$ District
City	240	20×20	$1.5 \times$ Region

3.3.3 Static Locating Problem

Consider the topology of a MANET at a single time to be the static problem. For evaluation define G by randomly placing user nodes in the area of operation having the control node centralized among the user nodes. In section 2.3.3 this was defined as the static minimum node locating problem for MANET and was equivalent to the STP-MSPBEL. The following sub-sections detail the experimentation regarding the static problems (the military and the SAR contexts) in section 3.3.3 and dynamic problems in section 3.3.4.

Static Problem Size Definition

Using the network parameters (area and $|U_t|$) defined in the previous sections, the size classes for the static method are derived from the expected size of the solution network. Since the MST solution provides an upper bound, it is used to determine the expected degree of the set of connecting points $E[|P|]$ for a network configuration. These sizes were then divided into small, medium, large, and extra large class sizes (S, M, L, X). The static problem class sizes are detailed in Table 3.5.

Table 3.5: Static Problem Size Classification

Class	**Area** (km^2)	$\lvert U \rvert$	$\mathbf{E}[\lvert P \rvert]$	$\mathbf{E}[\lvert N \cup P \rvert]$
S	10	4	10	14
S	16384	10	15	25
M	16384	10	20	30
M	1024	20	10	30
L	121	41	30	71
X	32768	60	40	100

Static Locating Formulation

The static problem assumes only one time step, $t = 1$; the network variables G_t, N_t, C_t, U_t, A_t, E_t, and P_t can be represented as G, N, C, U, A, E, and P. The static problem has a simple objective to minimize the size of the set of additional points needed to connect G.

$$\text{Given } G = (N, E), \text{ determine } G^* \text{ that minimizes } |P| \tag{3.9}$$

In the static locating problem, the set of points, P, are a means for defining locations for a deployed set of agent nodes, A.

3.3.4 Dynamic Locating Problem

Dynamic Problem Size Definition

Experimentation of the dynamic problem used the same problem size classifications as the static problem with the addition of specifying user node tour type. Table 3.6 lists the classes used to test the dynamic (both deterministic and stochastic) problem.

Table 3.6: Dynamic Problem Size Classification

Class	**Context**	**Tour**	**Area** (km^2)	$\lvert U_t \rvert$	$\lvert A_t \rvert$	$\lvert N_t \rvert$
S	Search	Track/Sweep	11.56	4	10	14
S	Military	Travel	16384	10	15	25
M	Military	Travel	1024	20	10	30
M	Military	Patrol	16384	10	20	30
M	Military	Polar	16384	10	20	30
M	Military*	Random	16384	10	25	35
L	Search	Track/Sweep	121	41	30	71
X	Military	Polar	32768	60	40	100

*not specified by Army documentation.

Military Context

The patrol, travel [67, 69, 73], polar, and random tour types were used in testing. These can require that the control node is also mobile, but for this dissertation, it will remain stationary. The random checkpoint definition provides the opportunity to show that the solution methods are not dependent on well formed user node movement patterns. The polar checkpoint definition is not completely random but has some level of randomness as it radially disperses user nodes away from the control node. Figure 3.4 shows a sample of each type of movement pattern.

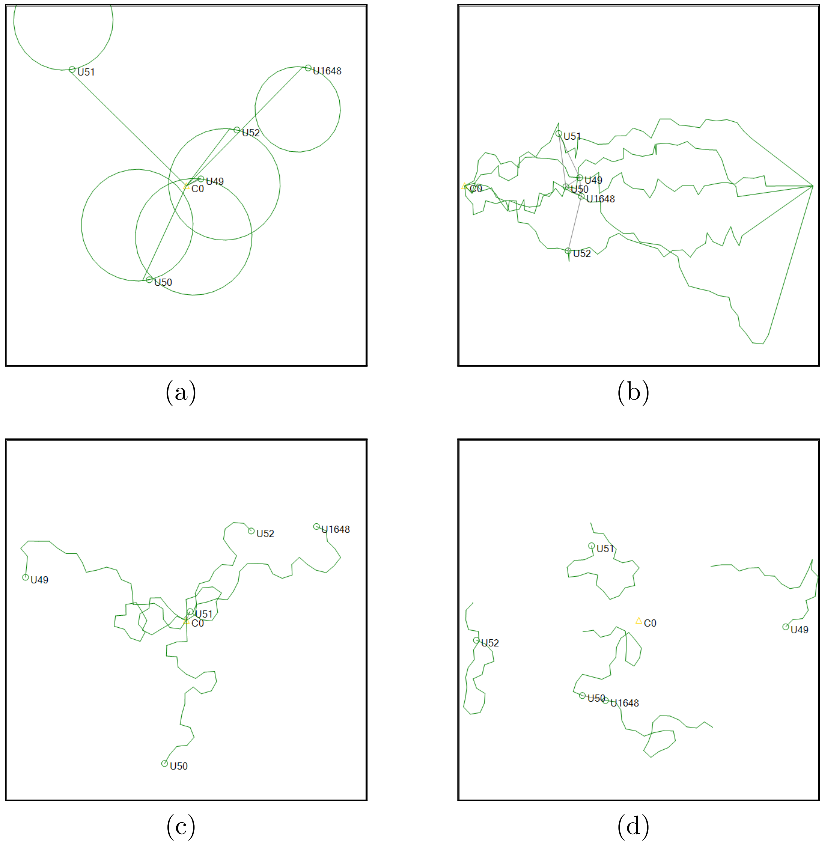


Figure 3.4: Sample military context maps showing a.) patrol b.) travel c.) polar d.) random user node tours

Search and Rescue (SAR) Context

Assume that at the start of a SAR problem, each search team will start at the first point of their tour, not at the control node. The search team's pattern within its area was modeled using the track and sweep pattern. Four different orientations of this pattern are modeled and randomly chosen (top-left to bottom right, top-right to bottom-left, bottom-right to top-left, and bottom-left to top-right). Figure 3.5 shows an S-SAR size problem. As a result of the previously stated 60% availability of staff needed, not all areas are covered.

Using the aforementioned movement patterns to define user node tours, the deterministic and stochastic environments will be discussed in the following sections.

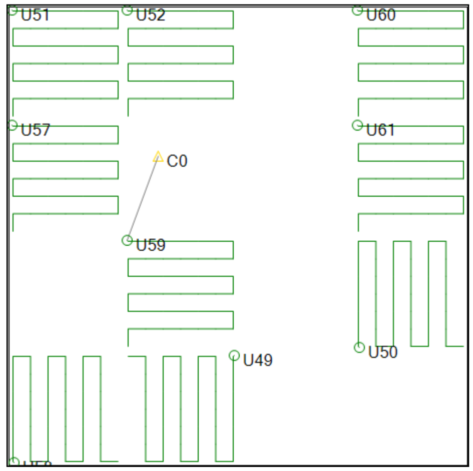


Figure 3.5: Sample SAR context map of user node tours with 60% availability

Dynamic Deterministic Environment

Heuristic 2 shows the sequence of events for the deterministic MANET model. Since the control node will have perfect knowledge of all of the node locations it has no additional responsibilities. In a real world application, the control node could actually be receiving network state updates and sending agent node locations. For simplicity, these functions are omitted leaving only the need to calculate the network metrics.

Heuristic 2: Deterministic Model Logic

Input: Number of steps T, agent node tours Q

for $t = 0$ **to** T **do**
    MOVE user nodes toward checkpoint
    MOVE agent nodes according to Q
    CALCULATE metrics

Dynamic Stochastic Environments

Heuristic 3 shows the events for the stochastic MANET model. Since there is possibility of user node deviations, the control node receives updates from the system and makes agent node location decisions. The addition of the deviation vector, the control node's FORECAST and DETERMINE steps differentiate the stochastic from the deterministic environment simulation.

Heuristic 3: Stochastic Model Logic

Input: Number of time steps T, agent node tours Q, deviation vector β_n

for $t = 0$ **to** T **do**
- UPDATE control node's knowledge of the network topology
- FORECAST network topology for s steps in the future (control node)
- DETERMINE new agent node locations (control node)
- SEND new agent node locations from control node
- MOVE user nodes toward checkpoint $f(\beta_n)$
- MOVE agent nodes
- CALCULATE metrics

User node movement has been described in Section 3.1.2. The control node determines which nodes it is connected to and obtains their actual locations and any additional pertinent information. With this information, the control node can use a few locating methods to determine the desired future agent node locations. For all connected agent nodes, this new locating information is sent to them, and they act accordingly. In previous works, this was done with meta-heuristics: genetic algorithm [18] and particle swarm optimization [12, 13].

Dynamic Locating Formulation

To evaluate the effectiveness of the locating methods throughout a dynamic problem, metrics are defined and recorded at the end of each time step. Cho [12] used the average number of connected users, average number of node hops per connected user node, average loss of bandwidth due to enemy node jamming, and mission completeness rate as metrics. Some of these are applicable for the current implementation.

The number of hops per connected node is not necessary given the modifications of this methodology. Minimizing the number of agent nodes in the simulation could result in an increase or decrease in the number of hops for a connected node. For example, it is possible for a case to exist where having more nodes would allow for a more direct line of communication to the control node via fewer node hops. However, without such additional nodes, the path to the control node could be circuitous and require more node hops. As a result, this metric will not be considered for evaluation.

The total number of agent nodes needed over the course of the mission for a fully connected network is the primary metric and objective and is defined as:

$$\min \max_{t} |A_t| \qquad t \in [0, T] \tag{3.10}$$

It is also important to ensure that all user nodes are network connected for each time step. The average network connectivity over the mission will provide this insight. Though it may be possible for the network to be disconnected between time steps, it is ensured to connect at real world time $t \times \tau$. As a result, the Average Number of Connected Users ($ANCU$) is an approximation of the real world network connectivity due to the defined discrete time intervals t = (0, T). This secondary objective is defined as:

$$ANCU = \frac{\sum_{t=0}^{T} \sum_{j=1}^{|U_t|} \{1 | \exists H_{t,i,U_t[j]}\}}{T+1} \qquad i \in C_t \tag{3.11}$$

With Cho's inclusion of jamming enemy nodes, maximizing bandwidth was an important metric to ensure communication between friendly nodes was maintained. Here, it was assumed that there is no interference that would degrade network bandwidth. And assume uncapacitated bandwidth between nodes; nodes will always be able to transmit and receive. Instead, the robustness of the network based on node degree was considered. The robustness/connectivity metric is defined by the assortativity metric of the network [48]. At a time t the assortativity metric range is [-1,1] representing a disjoint and well connected network, respectively. This value has been normalized within the range [0,1] to avoid possible confusion. Define the robustness metric at a given time B_t.

$$B_t = \frac{\sum_{e_{tij} \in E_t} (D_{ti} - \bar{D}_t)(D_{tj} - \bar{D}_t)}{\sum_{e_{tij} \in E_t} (D_{ti} - \bar{D}_t)^2} \tag{3.12}$$

where

D_{ti}	remaining degree of node i at time t (not including connecting edge)
$\bar{D}_t$	$\frac{1}{\lvert E_t \rvert} \sum_{e_{tij} \in E_t} D_{ti}$

It is beneficial to include a measure of movement rate violations. This is so that solutions that connect the network over all time steps with high robustness scores are not considered optimal while allowing agent nodes to violate movement constraints. Define *noia* to be the number of infeasible assignments, a count of the number of inter-time-step movement violations:

$$noia = \sum_{t=0, s=t+1}^{T} \sum_{n=0}^{|A_t|} \{1 | d_{ij} > A_{t,n}[v], \forall i, j \in A_t[M_t]\} \tag{3.13}$$

Among other measures, Manzano *et al.* [45] detailed the Endurance, Quantitative, and Qualitative Robustness Metrics which consider node reliability, which is not in the scope of this dissertation. Similarly, Sydney *et al.* [61] investigated network robustness in regard to node failure and malicious attacks, developing the Elasticity metric. There are several other metrics such as the distance, average degree, and clustering detailed by Mahadevan *et al.* [43] that do not provide the level of complex interrelationships that the assortativity metric does. The assortativity metric is most applicable in that it provides a better indication of the level of interconnectedness, focusing on high degree nodes, of the network.

The main focus of this dissertation is the dynamic locating problem. This is done in two steps. The first involves determining the connecting solutions P_t for each $t \in T$. Second, P_t is used to determine the size of the agent node set, $|A_t|$, and the assignment of points to their tours. To do this, pre-solve for P_t:

$$\text{Given } G_t,\ \text{determine } G_t^* = (N_t \cup P_t) \text{ that minimizes } |P_t| \text{ for } t \in [0, T] \tag{3.14}$$

Then, solve the assignment problem: Given G_t and P_t, determine Q_{tn} for agent nodes n at times t to cover all points in P_t for $t \in [0, T]$ by solving Formulation 3.15.

$$\min \quad z = |A_t| \tag{3.15}$$

$$s.t.$$

$$\sum_{n=1}^{|A_t|} \{1 | Q_{tn} = P_{ti}\} = 1 \qquad i \leq |P_t|, t < T \tag{3.15a}$$

$$(t - s)v - d_{ij} \geq 0 \qquad i = Q_{sn}, j = Q_{tn}, n \leq |A_t|, s < t < T, v \in \mathbb{R} \tag{3.15b}$$

$$\sum_{i=1}^{|C_0|} \{1 | Q_{0n} = C_{0i}[x, y]\} \qquad n \leq |A_t| \tag{3.15c}$$

$$\frac{\sum_{t=0}^{T} \sum_{j=1}^{|U_t|} \{1 | \exists H_{t,i,U_t[u]}\}}{T + 1} \geq w \qquad w = [0, 1], i \in C_t \tag{3.15d}$$

The primary objective is to determine the minimum size set A_t used to connect the network at all times. This will be done subject to some assignment restrictions based on time and distance. Constraint 3.15a ensures that Q_t covers all points in P_t. Constraint 3.15b considers the movement rate (v) of agent node n , and the time between successive assigned points (Q_{sn} and Q_{tn}) s and t. Where applicable, to ensure all nodes start at the control node, constraint 3.15c assigns the first point in each agent node n's tour, Q_{0n}, to be the location of the control node. Lastly, a service requirement is defined, constraint 3.15d, to meet a given level of connectivity, w, at each time where Q_t is defined. In a deterministic environment, where node movements are predictable and there is perfect knowledge of node tours, $w = 1$ always. In a stochastic environment, this is not the case. For military contexts, it is desired to have w approach 1. For other contexts, this value could be less than one based on the service requirements of the network.

The deterministic environment formulation, given binary connectivity, does not inherently require a robustness metric because it is assumed that it will always be connected. Adding the robustness metric to the deterministic environment assignment problem could result in better connected networks, *i.e.* increased number of paths or throughput (if it were being measured). Adding the robustness metric to the assignment method with addition of the *noia* metric could improve the performance of the assignment solutions when deployed in a stochastic environment. Combining the additional metrics, *noia* and B_t, the formulation becomes:

$$\min \quad z = |A_t| - \sum_{t}^{T} B_t + noia \tag{3.16}$$

$s.t.$

$$\sum_{n=1}^{|A_t|} \{1|Q_{tn} = P_{ti}\} = 1 \qquad i \leq |P_t|, t < T \tag{3.16a}$$

$$(t-s)v - d_{ij} \geq 0 \qquad i = Q_{sn}, j = Q_{tn}, n \leq |A_t|, s < t < T, v \in \mathbb{R} \tag{3.16b}$$

$$\sum_{i=1}^{|C_0|} \{1|Q_{0n} = C_{0i}[x,y]\} \qquad n \leq |A_t| \tag{3.16c}$$

$$\frac{\sum_{t=0}^{T} \sum_{j=1}^{|U_t|} \{1|\exists H_{t,i,U_t[u]}\}}{T+1} \geq w \qquad w = [0,1], i \in C_t \tag{3.16d}$$

3.4 Chapter Summary

In this chapter the problem contexts, the notation, and formulations have been defined. The solution methods for the static problem are presented in Chapter 4 followed by the dynamic problem solution methods for the deterministic and stochastic environments in Chapter 5. The solution methods provided include methods described by this dissertation, validation methods, and the previous works in the literature.

Chapter 4

Static Problem Solution Methods

In previous works, user nodes move using the Random Waypoint Model (RWM), or some variation, to traverse the field of operation [13]. Unlike Cho, it can be assumed that before deploying units, all operations would have some level of planning, whether it is informal or explicit. This research intends to leverage control knowledge, and pre-deployment planning to improve system performance during operation of the MANET models. With the assumption of deterministic (straight-line) movement, all user node locations at any given time can be determined. In a planning state, the system would be able to efficiently locate agent nodes to maintain connectivity of the network. This would help improve the performance of the Mobile Ad hoc Network compared with a reactive agent node locating method.

Dividing the mission time into discrete time steps makes tour planning tractable. Solving the static location problem at each time step provides a basis for developing agent node tour plans for the dynamic problem. The following methods provide a static locating heuristic solution, P, and a mathematical validation model is discussed.

4.1 Modified Minimum Spanning Tree Method

Because the Minimum Spanning Tree, MST, is an easily calculable solution for connecting networks, it was used as a starting point for developing a locating method. As was presented by Lin and Xue [24], their modified version of Kruskal's algorithm solves the MST with bounded edge lengths (MST-BEL) problem, inserting Steiner points on each MST edge with length greater than the edge bound. For the discussion in the following sections, the MST solution method is in reference to Lin and Xue's MST-BEL solution method. Additionally, the term "point" will refer to potential location for an agent node. In this dissertation, to potentially reduce the number of Steiner points compared to Lin and Xue's approach, the MST method was modified.

If there are unconnected nodes, the Modified MST (MMST) will attempt to bridge all connections by checking for groups (defined below), then, starting with the shortest node to node distance, bridge the connection with additional points, p_i. The following lists the cases in which each method would be used. Figure 4.1 shows this graphically.

Case 1: Nodes are within r, require no additional connecting point.

Case 2: Nodes are within $2 \times r$ and can be bridged by a single point.

Case 3: Nodes are further than $2 \times r$ and must be bridged by more than one point.

Case 4: Nodes are in a group and placing a point at the centroid would increase connectivity.

Case 5: Nodes are in a group and placing a point at the centroid would not increase connectivity.

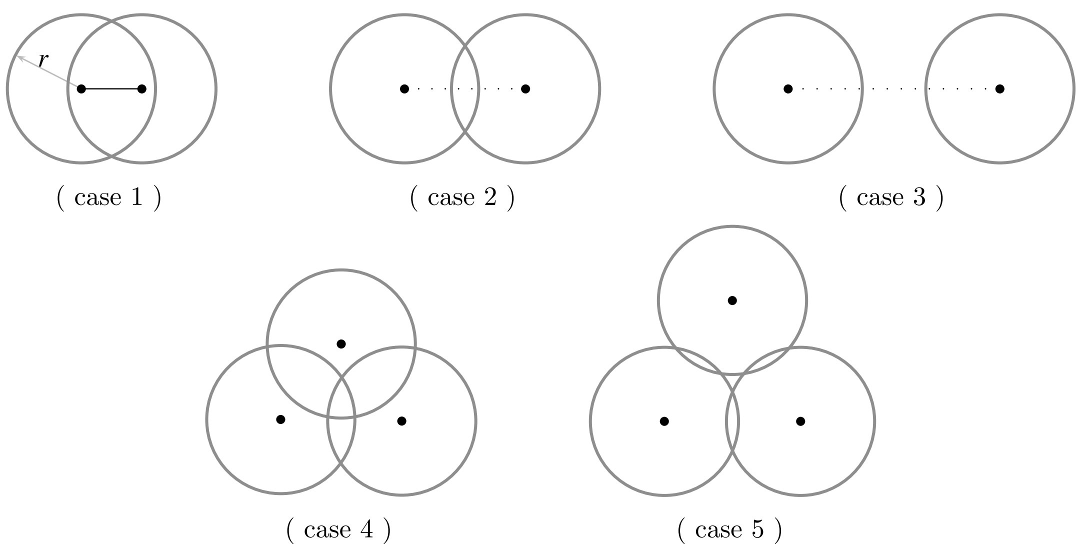


Figure 4.1: Connection types of pairs and groups of nodes

A group is defined as any number of nodes that are all within $2 \times r$ from each other. This is just an extension of the $2 \times r$ connectivity rule. This implies that there is overlapping coverage area. When multiple nodes' coverage areas overlap, a single point can be placed between the nodes in the overlapping area to connect them. Such sets of nodes are defined as a "group". In Heuristic 4, lines 3 - 8 detail how groups are identified.

From the Steiner point literature, the degree of such a centrally placed connecting node in a group should be three [24, 20]. However, Lin and Xue [40] and Chen *et al.* [10], show that the maximum Steiner point node degree is five. There is a case where there are at most five unconnected nodes that are all within $2 \times r$ from each other. If all five nodes can be connected by placing a node in the center, then this is the case that Lin and Xue describe. For this reason, groups are considered to be from three to five nodes in size.

In this dissertation, the classification for a centroid placed Steiner point does not require all additional points to have a minimum degree of three. Linearly placed nodes bridging a connection between a pair of noes can have a minimum degree of two. Figure 4.1 case 2 and 4.1 case 3 demonstrate such instances.

Given a graph with connection radius of r, five nodes can be arranged in a regular pentagon resulting in an unconnected network, Figure 4.2a. Lin and Xue's MST based solution would require four additional nodes to connect this network, Figure 4.2b. However, the graph can be connected with a centroid placed Steiner point, Figure 4.2c. This is optimal. Any number of nodes arranged in a regular n-sided polygon, where $n > 5$ would inherently be connected, Figure 4.2d. Any other polygon would have sub networks that can be reduced to one of the aforementioned cases.

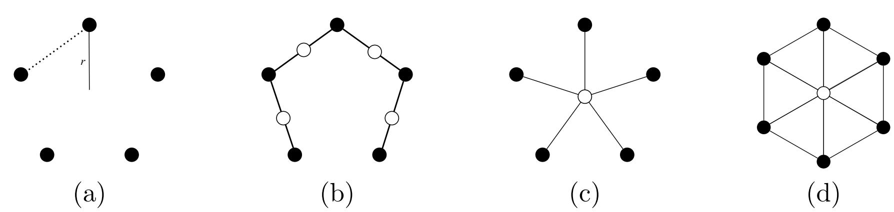


Figure 4.2: Steiner points in polygons a.) regular pentagon with radius r b.) Lin and Xue's MST solution c.) SMT solution d.) connected regular hexagon with radius r and a redundant Steiner point

Figure 4.1 case 4 shows three nodes with overlapping area where a single point can be used to connect three previously unconnected nodes. There is, however, a case where each of the nodes is within $2 \times r$ of each other, but a single point cannot connect them. In such a case, this formation is considered as multiple instances of case 2 and should be bridged with linearly placed points located equidistant between each pair of nodes.

Three possible methods are used to find this centroid point: average, mid-point, and circumcenter. The tests offer different solutions and are done sequentially; the first one that connects the group is used and the remaining tests are not evaluated.

The average point is the mean of the coordinates in a group. This is tested first and can consider the maximum number of points in a group (5). Midpoint finds the midpoint between the maximum and minimum coordinates of the group. These first two tests offering slightly different solutions work well for most cases where the circumcenter does not.

The circumcenter point is found by determining the center point of a circle with its circumference passing through each of the nodes. Because it can only determine the center of three points, it is tested last. It is easiest to see its applicability in a graph where nodes form a regular triangle. It is also beneficial since the distance from each point to the center will be equal. The circumcenter may not find a connecting point if two points are very close, and another far from them. The equations of each method are as follows for points A, B and C, with the complete centroid placement heuristic pseudo code in Heuristic 10.

Average point:

$$P_x = \frac{A_x + B_x + C_x}{3}$$
$$P_y = \frac{A_y + B_y + C_y}{3}$$

Mid-point:

$$P_x = \frac{Max(A_x, B_x, C_x) - Min(A_x, B_x, C_x)}{2}$$
$$P_y = \frac{Max(A_y, B_y, C_y) - Min(A_y, B_y, C_y)}{2}$$

Circumcenter point:

$$P_x = \frac{(A_x^2 + A_y^2)(B_y - C_y) + (B_x^2 + B_y^2)(C_y - A_y) + (C_x^2 + C_y^2)(A_y - B_y)}{D}$$
$$P_y = \frac{(A_x^2 + A_y^2)(C_x - B_x) + (B_x^2 + B_y^2)(A_x - C_x) + (C_x^2 + C_y^2)(B_x - A_x)}{D}$$

where:

$$D = 2(A_x(B_y - C_y) + B_x(C_y - A_y) + C_x(A_y - B_y))$$

Bridging is the process of adding points, in a straight line, from an unconnected node, to a control network connected node. The method determines the number of points needed to bridge the connection and how the nodes will be placed. If only one point is required, then it is placed half-way between the two unconnected nodes. If multiple points are needed, then the additional points are placed equidistant based on the value of r.

Heuristic 4 shows the pseudo code for the MMST. It is used to connect a network given a maximum number of additional points, m, that can be placed. The heuristic checks for groups of nodes that would benefit by centroid placed points. Line 4 references the grouping and centroid placement methods detailed in Heuristics 10 and 11. If none exist, then the closest unconnected node is bridged to the closest control network connected node by adding points to P_t. This is done recursively, with all existing nodes, including points in P_t, used in each calculation.

4.2 Dynamic MMST

Similar to Chakraverty *et al.* [9] and Yan *et al.* [84], the Dynamic MMST (DMMST) is a method of moving additional points to derive a good, if not optimal, Steiner Minimum Tree solution for the static problem. Unlike the reviewed works, here, the movement of points is based on iteratively created MMST solutions. Heuristic 5 outlines this process.

Given a static graph $G = (C \cup U, E)$ (Figure 4.3a) where the user nodes are dispersed throughout the field of operation, the MMST solution (Figure 4.3c) is used to define the

Heuristic 4: Modified Minimum Spanning Tree (MMST) Method

Data: connection radius r, maximum number of agent nodes m
Input: network $G = \{C \cup U, E\}$
Output: connected network $G^* = \{C \cup U \cup P, E\}$

while $|P| \leq m - 1$ **do**
  if *G is not connected* **then**
    if $FindCentroids(G, r)$ **then**
      add centroids from $FindCentroids$ to P
    else
      find closest pair of unconnected nodes i and j
      if $d_{ij} < 2 \times r$ **then**
        add the midpoint of i and j to P
      else
        add $\lceil \frac{d_{i,j}}{r} \rceil$ new points to P evenly between i and j
  else
    return P
return null

initial set of additional connecting points P. The MST solution is provided in 4.3b for comparison.

At a frequency of f_{DMMST} time steps, a new MMST solution is found for the updated network and each point $p_i \in P$ moves towards the new MMST solution points. Each successive time step uses $G = (C \cup U \cup P, E)$ to calculate the new connecting MMST solution. Iteratively using the MMST as an agent node locating method results in the additional points always attempting to move towards a connected solution. This process will always connect the network if at least $|P|_{MMST}$ number of additional points are available.

If there exists an STP solution with $|P| < |P|_{MMST}$, the continual calculation of MMST solutions, combined with motion of the points, and iterative removal of points can yield the needed Steiner tree points. Figure 4.4 illustrates an example of the evolution of the DMMST method into the Steiner Minimum Tree solution. Here, the starting solution P is a $|P|_{MMST} - 1$ node solution with a randomly chosen point removed (4.4a). The intermediate step, 4.4b, shows the movement of the remaining points based on a newly calculated MMST

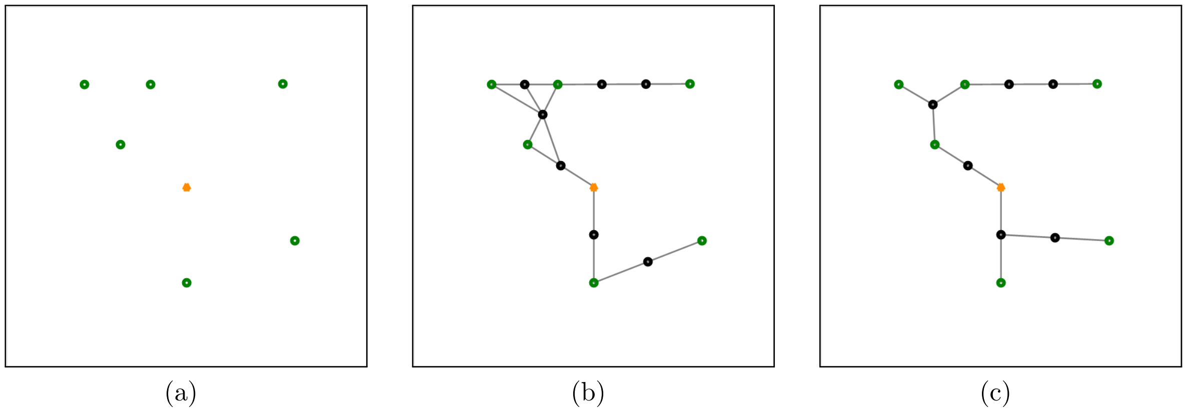


Figure 4.3: Example network a.) Given network G b.) MST-BEL solution where $|P| = 7$ c.) MMST solution $G \cup P$ where $|P| = 6$

solution. Finally, 4.4c shows the resultant connected network using $|P|_{MMST} - 1$ number of nodes. The iterative removal of a point and search are the basis of the Reduction method, discussed in the following section.

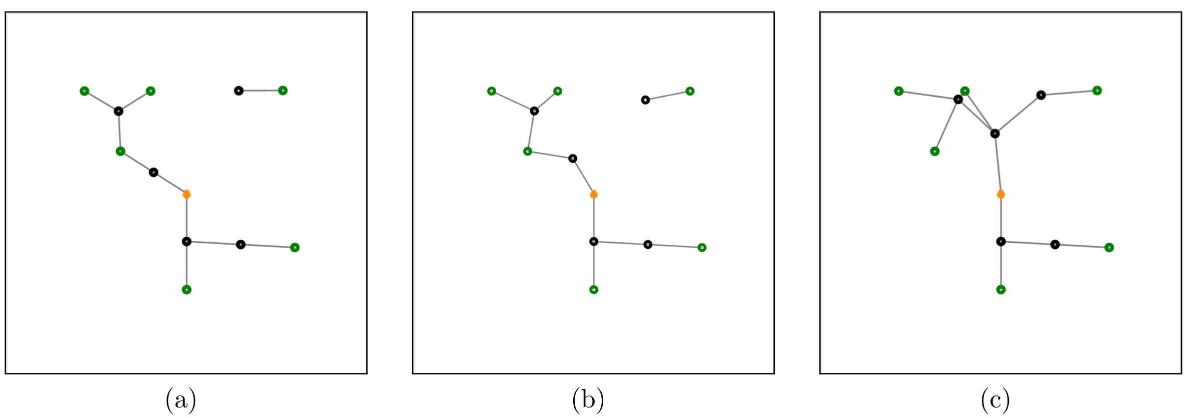


Figure 4.4: Progression of Dynamic MMST locating method a.) MMST solution with $|P| = |P|_{MMST} - 1 = 5$ points b.) Intermediate DMMST network where $|P| = 5$ points are moved c.) DMMST Steiner point solution for the $|P| = 5$ network

4.3 Reduction

With the knowledge that any static configuration can be connected with the MMST methodology, the $|P|_{MMST}$ number of points is then defined as an upper bound on the number of points needed. Previous works [35, 34, 17, 18, 12, 13] used particle swarm optimization as a method of finding a connected network but can require long run times to find a connected

Heuristic 5: Dynamic MMST Method

Data: number of DMMST iterations I_{DMMST}
Input: network $G = (C \cup U, E)$, time t, MMST calculation frequency, f_{DMMST}
Output: $G^* = (C \cup U \cup P, E)$

define P_{MMST} the set of locations for MMST solution
for $i = 1$ **to** I_{DMMST} **do**
    if $i \,\%\, f_{DMMST} = 0$ **then**
        update P_{MMST} with the MMST solution for $G = (C \cup U \cup P, E)$
    foreach *Point* p_i *in* P **do**
        move p_i towards closest point in P_{MMST}
return $G^* = (C \cup U \cup P, E)$

solution. The DMMST method is significantly faster than a replicated particle swarm, and can be used as part of a process to find a connected network. It can also be used with $|P| - n$ number of additional points where $n \in \mathbb{I}$ to attempt to find a connected network.

The Reduction method uses the DMMST($|P|_{MMST}$) as the initial test of feasibility. Starting at the control node, points in P are moved towards an MMST solution. With a frequency of $f_{DMMST} \geq 1$ a new MMST solution is computed. This prevents convergence around local optima. When f_{DMMST} is small, ex. $f_{DMMST} = 1$, an MMST solution is computed each iteration. At such a rate, oscillation between two MMST solutions is likely to occur without sufficient time for points to be moved toward either location.

A lower limit on the number of allowable points, p^m is defined as a parameter. This is to address unit deployment requirements. For example, if it is necessary for a minimum of four units to be deployed, then $p^m = 4$.

If there are enough available points, $|P| > p^m$, then one can be removed with the modified network being defined as G^-. If G^- can be connected using the DMMST method, then $|P|$ is reduced by removing a random point $p_r \in P$. This is the first iteration of the recursive removal and relocating of points that comprise the Reduction method.

If a connected solution for G^-_{DMMST} can be found, then G^* is set to that DMMST modified solution, G^-, and recurses. If for any value of $n > 0$, the DMMST($|P| - n$) is unable to connect the network, this may mean that the network cannot be connected with $|P| - n$ additional points, or that the Reduction method has simply failed to find the solution. This process continues, removing additional points, until the method cannot connect the network (see Heuristic 6).

Heuristic 6: Reduction Method

Input: connected network $G = (C \cup U \cup P, E)$,
minimum number of points allowed p^m, MMST calculation frequency f_{DMMST}
Output: Best found minimum point connected network G^*

$G^* \leftarrow G$
determine a point to remove, $p_r \in P$
define the modified network as G^-
if $|P| \geq p^m$ **then**
  remove p_r from G
  $G^- \leftarrow G - p_r$
else
  return G
if $DMMST(G^-, 0, f_{DMMST})$ *is connected* **then**
  $G^* \leftarrow G^-_{DMMST}$
  return $Reduction(G^-_{DMMST}, p^m, f_{DMMST})$
return G^*

Figure 4.5 is provided to detail this process graphically. Given an irregular pentagon, Figure 4.5a, the MST and MMST solutions (not shown) are identical and place connecting points on four of the five edges of the pentagon. This solution defines the upper bound on number of points needed and the starting set of four movable points. The Reduction method uses the DMMST to find a connecting solution. With a frequency of f_{DMMST}, new MMST solutions are determined. Figure 4.5b shows an intermediate MMST solution with a red circled point and dashed lines showing how it connects the network. Each of the four points are moved until the network is connected, Figure 4.5c.

The next iteration of the Reduction method randomly removes the node that was located at the red double-circled location, Figure 4.5d. Similarly, the movable points are directed to connecting solutions (Figure 4.5e) until the three point solution is determined (Figure 4.5f).

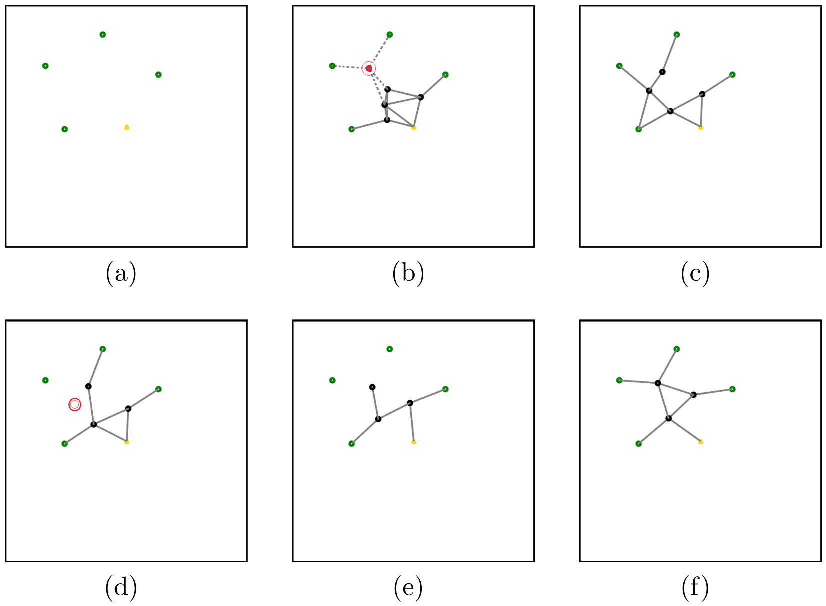


Figure 4.5: Progression of the Reduction locating method a.) given network ($|P|_{MST} = |P|_{MST} = 4$, though not shown) b.) at iteration 151 a new MMST solution is determined, indicated by the circled red point. Each point in P moves towards the closest point in the new MMST solution c.) DMMST solution with four points d.) randomly chosen point removed, indicated by double red rings e.) remaining points move toward new MMST solutions f.) the Reduction solution requiring three points

4.4 Quadratic Linear Program Validation

The mathematical model was developed to compare the Reduction solutions of smaller problems to an exact search. The following formulation is derived from Konak *et al.*'s ALOC model [35] that maximized flow of commodities, similar to a max flow problem. The static network is modeled to consider all user node to user node pairs as the set of commodities C. The flows over edges e that are within a distance of r ($d_{i,j} < r$) to another (user or agent) node are represented by f_{ce}. The supply/demand of all commodity pairs is regulated by F_c. These and a few other necessary variables are defined here, followed by the static mathematical formulation. Define:

$$\min \quad z = \sum_{i \in A} u_i \tag{4.1}$$

$s.t.$

u_i	$(0,1)\ \forall i \in A$	use agent node i variable
X_i, Y_i	$\mathbb{R}\ \forall i \in A$	x, y location variables for agent node i
C	$\{< i, j > \mid i, j \in U, i < j\}$	commodity: user-user network connection
f_{ce}	$(0,1)\ \forall c \in C$	flow variables of commodity c over edge $e \in E$
v_{ij}	$(0,1)\ \forall i, j \in N$	use arc variables from node i to node j
F_c	$(0,1)\ \forall c \in C$	supply/demand of all commodities
r	$\mathbb{R}$	connection radius
M	$\mathbb{R}$	big-M method

$$- \sum_{e_{nj} \in E} f_{ce} + \sum_{e_{in} \in E} f_{ce} = 0 \qquad \forall c \in C, e[i] \neq c[i], e[j] \neq c[j] \tag{4.1a}$$

$$- \sum_{\substack{e_{ij} \in E \\ e[i]=c[i]}} f_{ce} + \sum_{\substack{e_{ij} \in E \\ e[j]=c[i]}} f_{ce} = F_c \qquad \forall c \in C \tag{4.1b}$$

$$- \sum_{\substack{e_{ij} \in E \\ e[i]=c[j]}} f_{ce} + \sum_{\substack{e_{nj} \in E \\ e[j]=c[j]}} f_{ce} = -F_c \qquad \forall c \in C \tag{4.1c}$$

$$(U_i[x] - U_j[x])^2 + (U_i[y] - U_j[y])^2 - r^2 \leq M(1 - v_{ij}) \qquad \forall i, j \in U, i \neq j \tag{4.1d}$$

$$(U_i[x] - X_j)^2 + (U_i[y] - Y_j)^2 - r^2 \leq M(1 - v_{ij}) \qquad \forall i \in U, j \in A \tag{4.1e}$$

$$(X_i - U_j[x])^2 + (Y_i - U_j[y])^2 - r^2 \leq M(1 - v_{ij}) \qquad \forall i \in A, j \in U \tag{4.1f}$$

$$(X_i - X_j)^2 + (Y_i - Y_j)^2 - r^2 \leq M(1 - v_{ij}) \qquad \forall i, j \in A \tag{4.1g}$$

$$f_{ce} \leq u_i \qquad \forall c \in C, e_{ij} \in E, e[i] \in A \tag{4.1h}$$

$$f_{ce} \leq u_j \qquad \forall c \in C, e_{ij} \in E, e[j] \in A \tag{4.1i}$$

$$f_{ce} \leq v_{ij} \qquad \forall c \in C, e \in E \tag{4.1j}$$

$$F_c \geq 1 \qquad \forall c \in C \tag{4.1k}$$

With the primary objective of minimizing the number of agent nodes, the objective function was defined the sum of the number of nodes used. Constraints 4.1a - 4.1c manage flow of commodities through the network. Constraint 4.1a maintains zero net flow on all non-commodity source and sink nodes. Constraints 4.1b and 4.1c maintain sink and source properties of commodities at nodes.

To model connectivity, constraints 4.1d - 4.1g consider arc usage based on distance. Because the locations of agent nodes are defined by variables X_{ti} and Y_{ti}, each type of node pair (user-user, user-agent, agent-user, agent-agent) are defined respectively. Constraints 4.1h and 4.1i define the inbound and outbound flow of a commodity to an agent node based on usage (of the edge). Similarly, constraint 4.1j allows flow on an arc only if the arc is used. Constraint 4.1k ensures that all commodities are sent, *i.e.* all node pairs are connected.

4.5 Chapter Summary

Here, the Reduction method and its underlying functions, the MMST and DMMST, are described. The Reduction method iteratively randomly removes additional points, attempting to minimize $|P_t|$. The DMMST method provides a means for connecting the network with fewer than $|P|_{MMST}$ points. Searching for clusters and using heuristic placed nodes, the MMST seeks to improve upon the solution process described by Lin and Xue [40]. Evaluating a single topology of a network at a given time, these static methods can be used to solve locating problems at discrete time steps of a dynamic problem. Next, the static methods are used in the dynamic problem locating methods.

Chapter 5

Dynamic Problem Solution Methods

The main focus of this Chapter is the dynamic locating problem. The static solution methods were developed to aid in solving the connectivity problem at each discrete time step. A solution method was developed to solve the collection of static problems via the set of added points, P_t, for each $t \in [0, T]$. The locations of the method's solution are then used in an agent nodes assignment problem.

5.1 Deterministic Environment Pre-Solve Method

To solve Formulation 3.14, define the Reduction per Time Interval Method (RTI) as the per time interval method of finding P_t at each time $t \in [0, T]$. It is based on the Reduction method that yields good, if not optimal, minimum number of additional node connected solutions for a static problem. The RTI method uses the Reduction method with frequency $t < f_{RTI} < T$ throughout the course of the MANET model's operation. In a real world stochastic environment the value and relationships of parameters t and f_{RTI} should be carefully considered by the system designer. The time interval between t and $t+1$ should represent the absolute minimum required update rate to monitor the system. Similarly, with the deployment of agent nodes, the time interval $t, t + f_{RTI}$ represents the usage of the agent nodes.

The additional points needed to connect the network are added to P_t for each $t \in [0, T]$. Heuristic 7 shows the general logic during the RTI method. This may result in several sizes of P_t for different times. From the sets of P_t, a lower bound on the number of agent nodes needed, $\max_t P_t$, can be deduced. However, there may still be movement infeasibilities with

$|A_t| = \max_t P_t$ in which case additional agent nodes are needed. Thus, in formulation 3.15, the objective function attempts to minimize $|A_t|$.

Heuristic 7: Reduction per Time Interval Method

Data: Number of time steps T
Input: network $G_t = (C_t \cup U_t)$ where each user node, n, has a checkpoint list M_{tn}
Output: set of points P_t

for $t = 0$ **to** T **do**
  foreach *user node* u *in* U_t **do**
    move u towards next checkpoint in M_{tn}
  if $t \,\%\, f_{RTI} = 0$ **then**
    $P_t \leftarrow Reduction(G_t, 0, 0)$
return P_t

Figures 5.1 illustrate the progression of the RTI method. Assuming $f_{RTI} = 15$, $r = 100$ meters, and user nodes move at a rate of ten meters per time step. For $t = [0, 10]$, the network is connected, but the RTI method is not used until $t = 15$ where it was determined only one additional point was needed. At the next few RTI evaluations, $t = 30, 45, ..75$, the network requires multiple additional points. In this example, all user nodes' tours end at the control node at $t = 90$, requiring no additional points. The collection of additional points shown graphically in Figure 5.1 is also shown in Table 5.1. The notion $P_{t,n}$ refers to the n^{th} connection point required at time t.

Table 5.1: Tabular format of $P_t = \{(x, y)\} \forall t \in T$

t	P_{t1}	P_{t2}	P_{t3}	P_{t4}	P_{t5}	P_{t6}
0	(100,300)					
15	(180,300)					
30	(199,310)	(299,320)	(349,310)			
45	(197,275)	(314,193)	(295,291)	(350,375)		
60	(165,354)	(231,409)	(296,463)	(214,286)	(263,218)	(312,150)
75	(173,361)	(173,239)				
90	(100,300)					

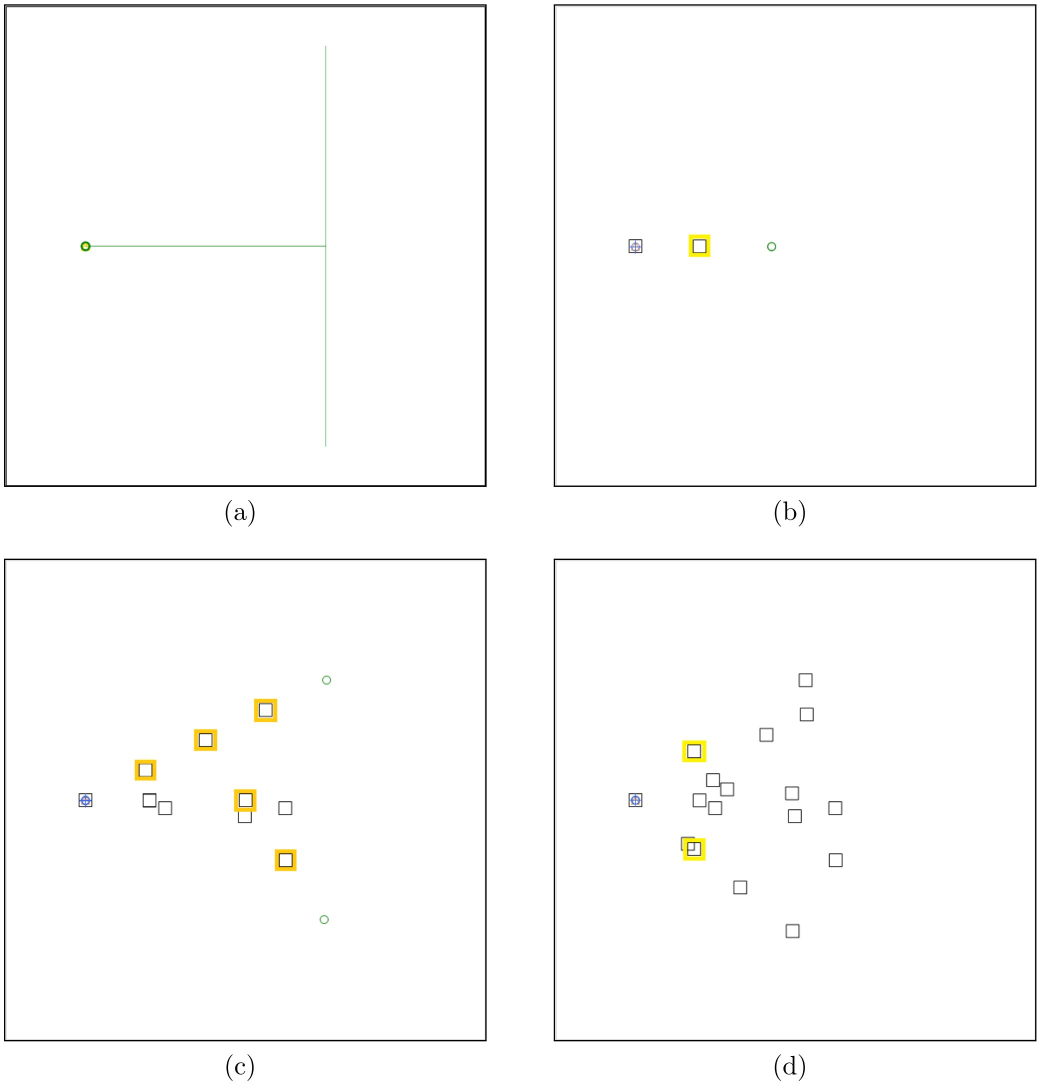


Figure 5.1: Progression of the RTI method where checkpoint boxes represent connection points. a.) G_t, showing the user node tours. b.) At $t = 15$ one additional point is needed, highlighted in yellow. c.) At $t = 45$ multiple points are needed and are highlighted in yellow, shown with previously found additional points. d.) At $t = 90$ all points found with the most recently found points (at $t = 75$) highlighted in yellow.

5.2 Deterministic Environment Point Assignment Method

After pre-solving for P_t for $t \in [0, T]$ (Formulation 3.14), the agent node tour optimization is done by solving the assignment problem, Formulation 3.15. Maximizing robustness is a conflicting objective. Increasing B_t would inherently require additional agent nodes to improve the node degree relationships. Additional agent nodes would also benefit the average network connectivity. Though these are benefits, it is desirable to minimize $|A_t|$; see Formulation 3.15.

Next, a meta-heuristic that searches the agent node tour assignments solution space to yield good solutions is presented. Following it, a mathematical model is presented to optimally solve the agent node tour assignment problem for validation, Formulation 3.15.

5.2.1 Meta-heuristics Overview

A guided search with a meta-heuristic can find good solutions in a relatively shorter time (compared to the mathematical model). Previous works use a particle swarm optimization for a reactive positioning approach. A genetic algorithm was believed to be a good fit for the assignment combinatorial problem. The ability to include infrequent alterations/ mutations furthering solution diversity should be beneficial.

The following sections detail the encoding and mechanics to apply this problem to a meta-heuristic search, specifically the genetic algorithm. It does not use the formulation of the mathematical model, but incorporates the objectives of the dynamic problem ($\min |A_t|$, $\max ANCU$, $\max B_t$, $\min noia$) into the objective function. For example, constraint 3.16d where $w = 1$ is inherently part of the mathematical problem. However, in a meta-heuristic, enforcing this constraint would result in a majority of the candidate solutions being considered infeasible. Instead of a constraint, it was incorporated into the objective function and measured as $ANCU$.

Also, as stated in section 5.1, with $|A_t| = \max_t |P_t|$, there may be movement constraint violations. That is to say that based on the identified points of P_t there may not be a consecutive assignment of locations that an agent node can travel from/to given its limited move rate, v and the time available to get to the next point. This type of violation is counted in $noia$ and is part of the genetic algorithm's objective function. Lastly, the primary objective to minimize $|A_t|$ is offset by the option to add additional agent nodes that would improve $ANCU$ or B_t.

5.2.2 Genetic Algorithm Encoding

The metaheuristic method assigns connection points from the pre-solve matrix P to the agent node assignment matrix Q. Assume initially that the size of Q is determined by the size of P_t. The matrix Q will be of size T $\times \max_t |P_t|$ but may vary in width due to the addition of agent nodes by the metaheuristic search. For any element of Q where there is no assignment, a null location, indicated by "-", is used to ensure the vector has a uniform width over all time steps. Table 5.2 provides an example pre-solve solution and encoded assignment solution for use with a meta-heuristic.

Table 5.2: Example pre-solve matrix P and padded assignment matrix Q

t	$P_{t,1}$	$P_{t,2}$	$P_{t,3}$	$P_{t,4}$
0	$P_{0,1}$			
10	$P_{10,1}$	$P_{10,2}$		
20	$P_{20,1}$	$P_{20,2}$	$P_{20,3}$	$P_{20,4}$
30	$P_{30,1}$			

t	A_1	A_2	A_3	A_4
0	-	-	$P_{0,1}$	-
10	$P_{10,2}$	-	-	$P_{10,1}$
20	$P_{20,1}$	$P_{20,3}$	$P_{20,4}$	$P_{20,2}$
30	$P_{30,1}$	-	-	-

Define Ω as the population of solutions and Ω_{ihtn} as the h^{th} heuristic determined assignment at iteration $i = (1, I_{Assign})$ of a point in P_t to an agent node n at a time t.

5.2.3 Genetic Algorithm Mechanics

The basic mechanics of the GA will be described here with the mutation method detailed in the following section.

- **Initialize Population** - The initial population Ω_0 of size *popsize* is randomly generated based on P_t.
- **Parent Selection** - For each time step, the population Ω_i will be sorted by OFV and the best $\frac{popsize}{4}$ pairs are selected for the parent sub-population $\Omega_i^{Parents}$.

- **Generate Offspring** - For each pair of parents, two offspring will be generated using uniform crossover. For each t in Q_t, the offspring will inherit from a randomly selected parent the assignments at that time.

- **Mutation** - Three types are defined and when a mutation is done one type is randomly selected. See section 5.2.4

- **Evaluate** - Define the function $Z(G)$ to be the weighted difference of the number of agent nodes, average number of connected users, robustness score, and number of infeasible assignments to determine the goodness of a solution. (A solution includes G_t, the user node tours, and agent node assignment solution Ω_{ih}.)

$$Z(G_t \cup Q_t) = \gamma_1 |\Omega_{ih0}| - \gamma_2 ANCU - \gamma_3 \sum_{t}^{T} B_t + \gamma_4 noia \tag{5.1}$$

$$\text{where } \gamma_1, \gamma_2, \gamma_3, \gamma_4 = (0,1)$$

- **Control Population** - At the end of each iteration, the population consists of the general population and offspring. For any assignment solution where there is an agent node column, Ω_{ihtn} that consists of all "-" values for all t ∈(0,T), remove that column. The population is then sorted and the lowest scoring solutions are culled to return to a size of *popsize*.

5.2.4 Genetic Algorithm Mutations

The canonical idea of mutation is to provide some perturbation of a solution. Here, a simple swap mechanism provides this. To accommodate finding feasible solutions (by loosening the constriction of the inter-time-step movement constraints) a method to add agent nodes is provided. Also, a method to remove an agent node is provided to address the objective of minimizing $|A_t|$. For a given population member, only one mutation per iteration can be performed at a maximum rate defined as f_{Mutate}. These methods are shown in Heuristic 8, lines 7-17.

Swap

Define $\Psi =< t, n_1, n_2 >$ as the swap vector of assignments for nodes n_1 and n_2 at time t where $n_1 \neq n_2$. When used in Heuristic 8, assume n_1 and n_2 are randomly chosen unless specified with another value. Table 5.3 shows an example swap $\Psi =< 10, A_1, A_4 >$.

Table 5.3: Example Swap in Ω_{ih}

t	A_1	A_2	A_3	A_4
0	$P_{0,1}$	-	-	-
10	$P_{10,1}$	$P_{10,2}$	-	-
20	$P_{20,1}$	$P_{20,2}$	$P_{20,3}$	$P_{20,4}$
30	$P_{30,1}$	-	-	-

t	A_1	A_2	A_3	A_4
0	$P_{0,1}$	-	-	-
10	$\boxed{-}$	$P_{10,2}$	-	$\boxed{P_{10,1}}$
20	$P_{20,1}$	$P_{20,2}$	$P_{20,3}$	$P_{20,4}$
30	$P_{30,1}$	-	-	-

Agent Node Column Addition

This method adds an agent node column to the assignment matrix Ω_{ih}. It performs a swap where one of the nodes is the newly added agent node, $\Psi =< R()T, \Omega_{iht[\text{last}]}, n_2 >$. Table 5.4 shows the addition of an agent node column, A_5, and a swap that included that node, $\Psi =< 20, A_5, A_2 >$.

Table 5.4: Example Agent Node Column Addition and Swap in Ω_{ih}

t	A_1	A_2	A_3	A_4
0	$P_{0,1}$	-	-	-
10	$P_{10,1}$	$P_{10,2}$	-	-
20	$P_{20,1}$	$P_{20,2}$	$P_{20,3}$	$P_{20,4}$
30	$P_{30,1}$	-	-	-

t	A_1	A_2	A_3	A_4	$\boxed{A_5}$
0	$P_{0,1}$	-	-	-	-
10	-	$P_{10,2}$	-	$P_{10,1}$	-
20	$P_{20,1}$	$\boxed{-}$	$P_{20,3}$	$P_{20,4}$	$\boxed{P_{20,2}}$
30	$P_{30,1}$	-	-	-	-

Agent Node Column Removal

If an assignment solution has added agent nodes where $|\Omega_{ih0}| > \max_t |P_t|$, then a node can be removed. If $|\Omega_{ih0}| < \max_t |P_t|$, the solution is infeasible because it would not be possible to assign all points in P_t to Q, violating constraint 3.15a. This type of constraint violation is not allowed. Randomly choose a node to remove, reassigning any checkpoints associated with the removed node to the other agent nodes, and remove it form Ω_{ih}. Table 5.5 shows the reassigning of $P_{20,3}$ to A_2 so that A_3 can be removed.

Table 5.5: Example Reassign Checkpoints and Agent Node Column Removal in Ω_{ih}

t	A_1	A_2	$\boxed{A_3}$	A_4	A_5
0	$P_{0,1}$	-	-	-	-
10	-	$P_{10,2}$	-	$P_{10,1}$	-
20	$P_{20,1}$	-	$\boxed{P_{20,3}}$	$P_{20,4}$	$P_{20,2}$
30	$P_{30,1}$	-	-	-	-

t	A_1	A_2	A_4	A_5
0	$P_{0,1}$	-	-	-
10	$P_{10,2}$	$P_{10,2}$	$P_{10,1}$	-
20	$P_{20,1}$	$\boxed{P_{20,3}}$	$P_{20,4}$	$P_{20,2}$
30	$P_{30,1}$	-	-	-

Heuristic 8: Tour Assignment - Genetic Algorithm

Input: set of connection Points P
Output: agent node tour assignments Q

define starting population Ω_0
for $h = 0$ **to** $|\Omega_i|$ **do**
    Evaluate $\Omega_{i,h}$
for $i = 0$ **to** I_{Assign} **do**
    Choose parents
    Create offspring and add to Ω_i
    for $h = 0$ **to** $|\Omega_i|$ **do**
        if $R() < \frac{f_{Mutate}}{4}$ **then**
            if $|\Omega_{i,h,0}| > \max_t P_t$ **then**
                randomly select agent to remove A^-
                reassign checkpoints from A^-
                remove A^-
        else if $R() < \frac{f_{Mutate}}{2}$ **then**
            add A^+ to $\Omega_{i,h}$
            swap $\Psi =< R()T, A^+, n_2 >$
        else if $R() < f_{Mutate}$ **then**
            swap $\Psi =< R()T, n_1, n_2 >$
        Evaluate $\Omega_{i,h}$
    Control population
    Store best solution
return best solution

5.3 Assignment Mathematical Model for Validation

The mathematical model, equation 5.2, is used to validate the performance of the deterministic environment pre-plan and assignment methods. With perfect knowledge and no variability in user node movement, the solutions should provide an always connected solution. The pre-plan and assign method should perform as well as the mathematical model.

The dynamic environment mathematical model is identical to the static formulation, derived from the ALOC formulation of Konak *et al.* [35], with the addition of a time index and the last two constraints. To manage inter-time-step constraints, 5.2l ensures that the agent nodes used are consistent over time steps and 5.2m is the agent node movement constraint. Essentially, it solves the multiple static problems (based on the time index), checking the inter-time-movement distance and ensuring consistent agent node usage over time. A description of the notation follows. Note that the Robustness objective formulation (see Equation 3.16) is complex and non-linear and is not included in this mathematical formulation.

$$\min \quad z = \sum_{\substack{i \in A_t \\ t \in T}} u_{t,i} \tag{5.2}$$

$s.t.$

$$-\sum_{e_{tnj} \in E} f_{tce} + \sum_{e_{tin} \in E} f_{tce} = 0 \qquad c \in C_t, e[i] \neq c[i], e[j] \neq c[j], t \in T \tag{5.2a}$$

$$-\sum_{\substack{e_{tij} \in E \\ e[i]=c[i]}} f_{tce} + \sum_{\substack{e_{tij} \in E \\ e[j]=c[i]}} f_{tce} = F_{tc} \qquad \forall c \in C_t, t \in T \tag{5.2b}$$

$$-\sum_{\substack{e_{tij} \in E \\ e[i]=c[j]}} f_{tce} + \sum_{\substack{e_{tnj} \in E \\ e[j]=c[j]}} f_{tce} = -F_{tc} \qquad \forall c \in C_t, t \in T \tag{5.2c}$$

$$(U_{ti}[x] - U_{tj}[x])^2 + (U_{ti}[y] - U_{tj}[y])^2 - r^2 \leq M(1 - v_{tij}) \qquad \forall i, j \in U_t, i \neq j, t \in T \tag{5.2d}$$

$$(U_{ti}[x] - X_{tj})^2 + (U_{ti}[y] - Y_{tj})^2 - r^2$$

$$\leq M(1 - v_{tij}) \qquad \forall i \in U_t, j \in A_t, t \in T \tag{5.2e}$$

$$(X_{ti} - U_{tj}[x])^2 + (Y_{ti} - U_{tj}[y])^2 - r^2$$

$$\leq M(1 - v_{tij}) \qquad \forall i \in A_t, j \in U_t, t \in T \tag{5.2f}$$

$$(X_{ti} - X_{tj})^2 + (Y_{ti} - Y_{tj})^2 - r^2$$

$$\leq M(1 - v_{tij}) \qquad \forall i, j \in A_t, t \in T \tag{5.2g}$$

$$f_{tce} \leq u_{ti} \qquad \forall c \in C_t, e_{ij} \in E, e[i] \in A_t, t \in T \tag{5.2h}$$

$$f_{tce} \leq u_{tj} \qquad \forall c \in C_t, e_{ij} \in E, e[j] \in A_t, t \in T \tag{5.2i}$$

$$f_{tce} \leq v_{tij} \qquad \forall c \in C_t, e \in E \tag{5.2j}$$

$$F_{tc} \geq 1 \qquad \forall c \in C_t \tag{5.2k}$$

$$u_{si} = u_{ti} \qquad \forall i \in A_t, t = s + 1, s, t \in T \tag{5.2l}$$

$$(X_{ti} - X_{si})^2 + (Y_{ti} - Y_{si})^2 - m^2$$

$$\leq M(1 - u_{si}) \qquad \forall i \in A_t, t = s + 1, s, t \in T \tag{5.2m}$$

C_t	$\{< i, j > \mid i, j \in U_t, i \neq j\}$	commodity representing user-user network signal
u_{ti}	$(0, 1)\ \forall i \in A_t$	use agent node i at time t
f_{tce}	$(0, 1)\ \forall c \in C_t$	flow of commodity over all $e_{tij} \in E_t$ at time t
v_{tij}	$(0, 1)\ \forall i, j \in N_t$	use arc from node i to node j
X_{ti}, Y_{ti}	$\mathbb{R}\ \forall a \in A_t$	x, y location variables for agent node i at time t
$U_{ti}[x], U_{ti}[y]$	$\mathbb{R}\ \forall u \in U_t$	x, y location variables for user node i at time t
F_{tc}	$(0, 1)\ \forall c \in C_t$	used to ensure flow of all commodities at time t
r	$\mathbb{R}$	connection radius
m	$\mathbb{R}$	maximum move distance between time steps
M	$\mathbb{R}$	for big-M method

5.4 Stochastic Environment Positioning Methods

To evaluate the stochastic environment, three levels of randomness are defined. Level 1 randomness was previously presented in the dynamic deterministic environment problem but not labeled as such. Level 1 randomness is used to evaluate the variation due to seed (of the random number generator) of the pre-plan and assignment process in the deterministic environment. Define the term "Plan" as the result of the Level 1 testing. Note that the randomness required to generate the instances used as input here is not considered. Level 2 randomness is used to generate the user node tour realizations in the stochastic environment. Level 3 randomness is used to evaluate the variation due to seed (of the random number generator) of the usage of the Plan solution with possible re-planning in the stochastic environment.

After completing the Plan experimentation, there are ninety instances (nine types, ten instances) and three replications of each (270 total replications). Preliminary testing revealed very low variance between replications allowing for a small run size. From each run of three replications, it can be logically assumed that the decision maker will select the best Plan solution. The best of each of the Level 1 runs will be used to generate five realizations (Level 2). Each realization will define a Level 3 run of three replications. Figure 5.2 shows this graphically, illustrating the three Plan replications, selection of the best solution (indicated by *), generation of realizations, and the use of the Plan method (with possibility of re-planning) in a stochastic environment.

For example, a solution labeled M_SAR_09_1003_2004_3002 is the second replication of the fourth realization of the best found Plan replication of the ninth instance of an M size problem with a SAR tour type. Figure 5.3 shows a few of the maps from this process with user node tours, drawn in green 5.3a, of an S_SAR instance, the selected best Plan solution 5.3b, and two stochastic environment user node tours drawn in black, 5.3c and 5.3d.

Practically, the decision maker plans the mission in the deterministic environment and deploys the system in a stochastic environment. Assume that during the mission, the size

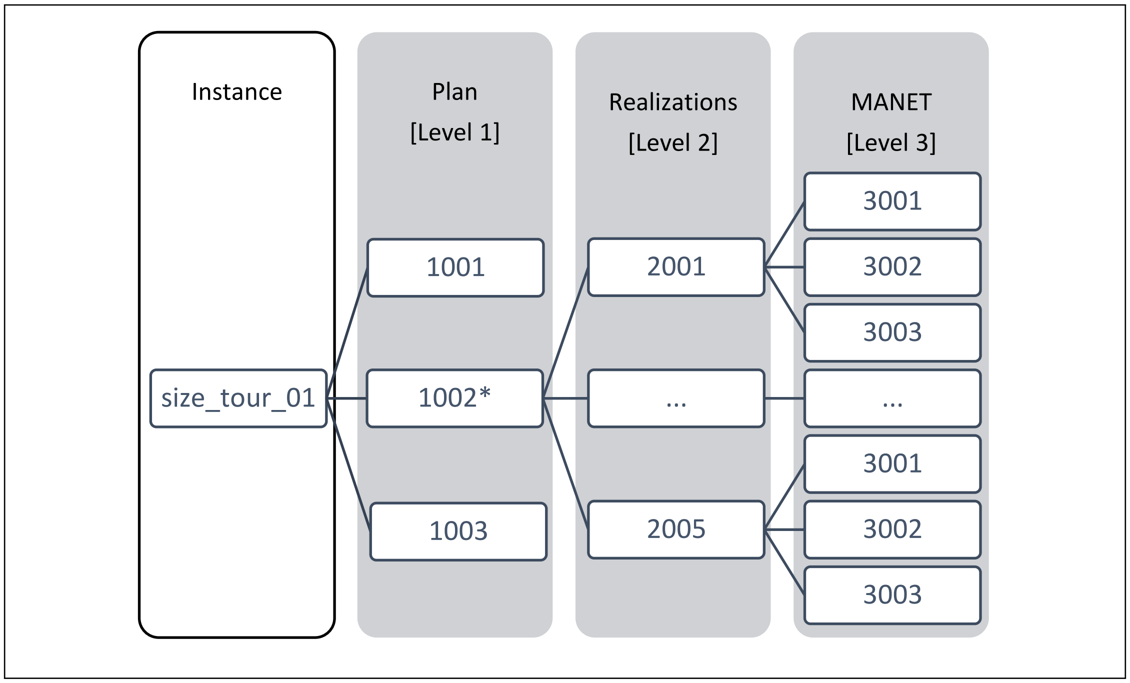
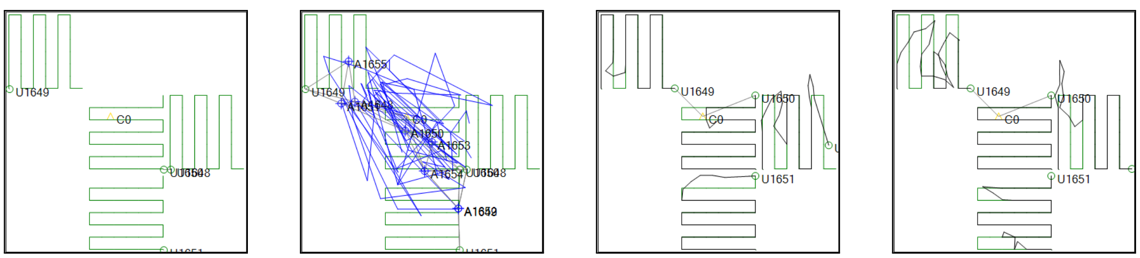


Figure 5.2: Levels of Randomness



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

Figure 5.3: Instance S_SAR_01 a.) user node tours b.) pre-plan and assignment solution 1001 c.) realization 2001 d.) realization 2002

of the set of agent nodes can be increased or decreased. The control node will then manage agent node tours and locations.

5.4.1 Simulation with Re-plan Option

Define MANET as the application of the best Level 1 Plan replication in a stochastic environment with the option to re-plan given user node deviations exceed a given threshold. For every time step, the control node will update its internal model of the network, forecast user node locations, and then determine agent node positions. The macro-level stochastic environment logic was presented in Heuristic 3. Heuristic 9 expounds upon the control node's logic given the network G_t at each time step t. The number of forecast time steps was defined to be $s = 6$ in section 3.1.1. Define g_t, the control node's copy of G_t used to model the control node's limited knowledge of the system when not fully connected, node movement, and determine connecting solutions.

Heuristic 9: Control Node Decision Logic in a Simulation

Input: time t, network G_t, Plan solution
Output: updated agent node tour, g_{t+s}

```
foreach u ∈ U_t do
    if u is connected to C_t then
        UPDATE g_t[u] position to u
    else
        MODEL g_t[u] movement to t
    RECORD g_t[u] position
    FORECAST movements at g_{t+s}[u]
foreach a ∈ A_t do
    FORECAST movements at g_{t+s}[a]
UPDATE connectivity of g_{t+s}
MEASURE deviations of g_{t+s}
if percentage of deviating user nodes ≥ Δ_replan then
    DETERMINE agent node tours with GA(RTI(g_t)) ∀(t, T)
else
    DETERMINE agent node in g_{t+1} position with Plan
return g_{t+s}
```

Model, Update, and Forecast Network Locations

With the assumption that the control node can only determine locations of connected nodes in G_t, if a node is disconnected, its location is assumed based on adherence to node tours. The locations of all nodes are recorded and then used to forecast locations, g_{t+s}. The connectivity state of g_{t+s} is then determined. The g_{t+s} is used to determine the degree of deviation in the forecast network.

Measuring Deviations

At each time step, comparing g_{t+s} and the original user node's planned tour, G_t, the distance from the planned tour can be determined. The distance between the user node's current position, i, and the original tour location, c, at a given time, is indicated by d_{ic}. Define d^{dev} as the threshold for a user node's deviation to be considered significant. It can be reasoned to be a percentage of the connection radius where deviations beyond this threshold would disconnect the network. Define Δ_{replan} as the threshold (percentage of deviating user nodes) to trigger re-planning agent node solutions. Its value should be comparable to the service constraint requirement w (Constraint 3.16d). For example, given $\Delta_{replan} = 0.3$, $\frac{|U_t : d_{ic} > d^{dev}|}{|U_t|} > \Delta_{replan}$, $t = 7$ and T=12, the GA(RTI) method will be used to determine agent node tours from $t = 8$ to 12.

Additionally, the network connected state based on user node location forecasts is considered. If at $t + s$, the network is forecast to be connected, there is no need to re-plan. Agent nodes will continue to follow their assigned tours.

Define I_{DMMST}, I_{GA} to be the number of iterations for the DMMST used in the re-planning (RTI) and assignment search (genetic algorithm). The run length of each method is dependent on problem size. Specifically, for I_{DMMST}, the size of the area of operation and number of agent nodes factor into determining the number of iterations. The value of I_{GA} is dependent on the number of agent nodes and the T$-t$, the size of the re-plan assignment matrix. For both, I_{DMMST} and I_{GA}, their values should allow convergence. Define a 5-tuple for the parameters used in the re-preplanning method:

$$< d^{dev}, \Delta_{replan}, I_{\text{DMMST}}, I_{\text{GA}}, s > \tag{5.3}$$

The evaluation time is limited to $t = \{0,\text{T}\}$ defined by original tour for a consistent basis when comparing to the deterministic environment. Thus, when deviations delay a user node's tour, it will not finish its tour.

One assumption was made for the search and rescue (SAR) and random (RAN) tour types that do not start at the control node. The positions of the agent nodes at $t = 0$ will be those determined by the pre-plan method. All other methods will have all nodes originate at the control node.

Determine Agent Node Tour Modifications

The logic of the control node used to modify agent node tours is the main contribution to the MANET path planning research area. It is what differentiates itself from a reactive approach. In reference to the Heuristic 9, lines 10-15 deal with agent node positioning. This logic is also used in the deterministic case allowing only Plan solution positions to be returned.

If it is determined that the percentage of deviating user nodes is more than the threshold Δ_{replan}, then g_t is modified before being passed to the re-plan method. Only time steps t to T are considered, excluding points at earlier times, 0 to t-1. The control node then uses the RTI pre-plan method , Heuristic 5.1, to determine connecting points. These points are assigned to agent nodes by the genetic algorithm search. Unlike the pre-plan process, the re-plan GA does not allow removal or addition of agent nodes.

Update Agent Node Tours

Concluding each time step, the control node will attempt to send the locations of its internal model, g_t. Depending on the network's state, this network is either in accordance with the agent node's previously assigned tour or a modified tour. Note, the case may exist where a modification was communicated at a previous time step, and the successive issued commands are in accordance with the new tour, not the original. These cases are dependent on an agent node being network connected to the control node.

When an agent node is not network connected, it will continue following its tour. It would not receive any modifications to its tour until reconnected. Generally speaking, each agent node will have a tour that covers all time steps, $t = 0$ to T. Thus, with no additional instruction, the agent node will continue to operate, potentially with outdated instructions, until the end of the simulation.

5.4.2 Reactive

Define the Reactive method as the use of the deterministic DMMST method for determining agent node positions. Similar to the MANET method, for each time step, the control node will update g_t, forecast user node locations at time $t + s$, then determine agent node locations.

Inherently, using a reactive method results in a limited span of instructions of only one time step. An additional instruction is needed for disconnected networks. If an agent node becomes disconnected, and there is no instruction for the $t+1$ time step, it will move towards the control node. If it is not connected and there are points in its tour to follow, continue to follow the tour. If it is connected, its tour is overwritten from t to $t + s$ and it moves according to this new tour.

5.5 Chapter Summary

The dynamic deterministic and dynamic stochastic problems, have been presented with solution methods for each. In a deterministic environment, the RTI and genetic algorithm method were developed to pre-plan and assign agent node tours, respectively. As a decision maker, after deterministic environment agent node tour assignment, the best plan was selected. Realizations of the operation of the network in a stochastic environment were generated to test the usage of these agent node assignments. Methods for measuring and responding to user node deviations by re-planning where necessary were presented.

Chapter 6

Validation, Experimentation, and Analysis

This dissertation seeks to provide a method to use prior knowledge of user node tours to pre-plan and re-plan if necessary. When testing the static problem, the validation process attempts to determine if the performance of the Reduction method is as good as a mathematical model for small size problems. The Reduction method is then used to investigate if it performs as well as the existing research of Lin and Xue [40].

In the dynamic problem the performance in deterministic and stochastic environments is evaluated. With perfect knowledge in a deterministic environment, the Plan (pre-plan and assign) method is validated by comparing it to a mathematical model, limited to small problems. The benefit of the Plan solution is then compared to the Reactive method. In the stochastic environment, the MANET (using the Plan method with possible re-planning) is evaluated to determine its benefit over the Reactive only positioning method.

6.1 Static Problem

Given any static graph, during the planning stage the objective is to return a connected network with the minimum number of added points and, subsequently, the minimum number of agent nodes. Connected networks can always be found if there is an unlimited number of points/agents that can be placed. It is assumed that there is no limit but the number of agent nodes should be minimized.

To compare the performance of this dissertation with the literature, Lin and Xue's MST method was used as a basis. This method provides an easily computed upper bound for the objective of minimizing $|A_t|$. The Reduction method uses repetitive DMMST calculations to direct movable points to a connecting solution. Because the method has a potential random removal of a node as part of the search, a run of four replications (varying seeds) was used in the analysis.

6.1.1 Validation

To validate these solution methods they were compared to the optimal solution obtained from the mathematical model (Math). Define the null hypothesis:

$$H_0 : \mu_{|P|_{Reduction}} = |P|_{Math} \tag{6.1}$$

For a run of the Reduction method, this tests if the average size of the set of additional points, $\mu_{|P|_{Reduction}}$, is comparable to the size of the set found by the mathematical model, $|P|_{Math}$.

The math model was developed for CPLEX on Auburn University's High Performance Cluster (HPC). Some preliminary experimentation was conducted on the NEOS server [16, 19, 27]. The problem instances used for validation were only S class or smaller due to math model computation time requirements. A total of 33 problems where $|U| = 5$ and 27 additional problems where $|U| = 10$ were evaluated. Seven ($|U| = 10$) problems did not finish (dnf) within thirty hours of computation time on Auburn's HPC.

A one sample two-tailed t-test was used to determine statistical differences between the values of $\mu_{|P|_{Reduction}}$ and $|P|_{Math}$. Table 6.1 list a sample of the results of these tests with ($\alpha = 0.95$, d.f. $= 2$) $t_{critical} = 4.30$. Appendix Table C.1 fully lists these results. The t-stat could only be determined for runs where the standard deviation of the number of connecting points $\sigma_{|P|_{Reduction}} > 0$. In all but one run of the Reduction method $\sigma_{|P|_{Reduction}} = 0$ making the t-statistic incalculable. Only 1.7% (1/60) runs could be determined to be statistically equivalent. Contradictory to the desired outcome, there was only one run where $|P|_{Math} < \mu_{|P|_{Reduction}}$; 1.7% (1/60) cases had a better math solution.

Since there were so few runs that could be compared with the t-test, the values of $|P|$ were compared via ratios. Most important of these, the $\frac{Reduction}{Math}$ ratio shows that the Reduction method performs as well as the math model by finding optimal solutions in 86.7% (52/60) test problems. If the proportion of problems where the math model did not finish is not considered then the Reduction method finds the optimal solution in 98.1% (52/53) of the runs. Similarly, if the Reduction method can be considered to perform better than the math model for problems where the math model did not finish, then the Reduction method performs as well as the math model in 98.3% (59/60) of all runs. Figure 6.1a shows the proportion of runs where $\mu_{|P|_{Reduction}} = |P|_{Math}$. Assuming that this performance scales to larger problems, then the Reduction method is suitable to connect networks at discrete time steps of a dynamic problem.

6.1.2 Experimentation and Analysis

Table 6.1 also lists $|P|_{MST}$ and $|P|_{MMST}$. In all cases, the MMST finds a solution with less than or equal to the number of additional points of a MST solution. Similarly, comparing the Reduction, MST, and MMST solutions, shows that the Reduction method can always find a solution with less than or equal to the number of additional points required by the MST or the MMST method. Figure 6.1b shows the proportion of runs where $|P|_{MST} > \mu_{|P|_{Reduction}}$. This verifies the improvements of this dissertation over the Lin and Xue [40] MST method.

Table 6.1: Sample Static Problem Results and Analysis of $|P|$

Graph	$\lvert P\rvert_{Math}$	$\lvert P\rvert_{MST}$	$\lvert P\rvert_{MMST}$	$\mu_{\lvert P\rvert_{Reduction}}$	$\sigma_{\lvert P\rvert_{Reduction}}$	**t-stat**
5-000	2	2	2	2.00	-	-
5-008	3	3	3	3.00	-	-
5R-00	4	4	4	4.00	-	-
5R-01	4	4	4	4.00	-	-
10-13	dnf	7	6	6.00	-	-
10-34	6	8	7	6.25	0.5	1

In approximately 37% (22/60) of the runs $|P|_{MST} = \mu_{|P|_{Reduction}}$. In these runs, there was no way to reduce $|P|$ due to the topology of the problem instance. Figure 6.2 is an example of this, showing that the MST, MMST, and Reduction methods result in the same

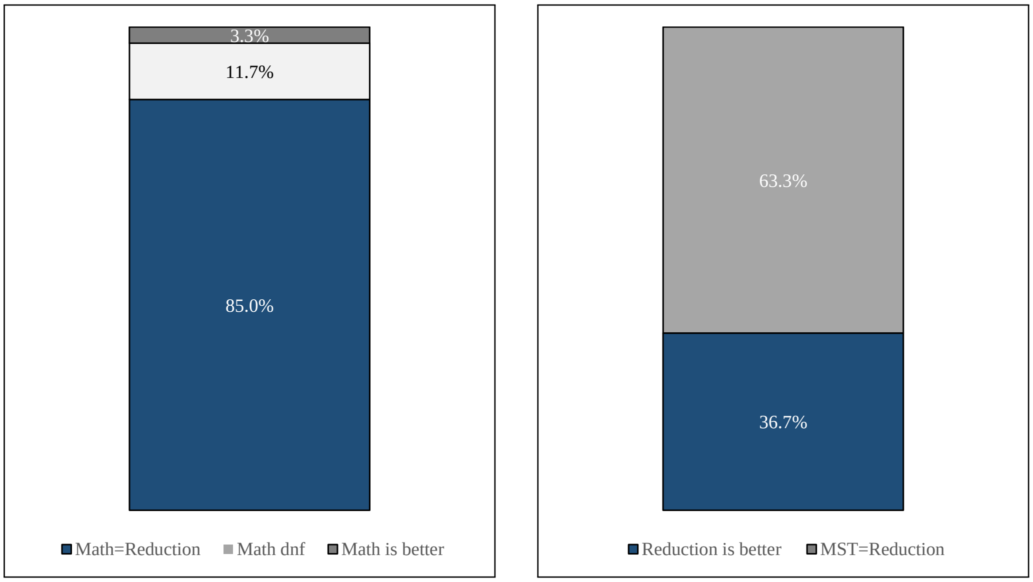
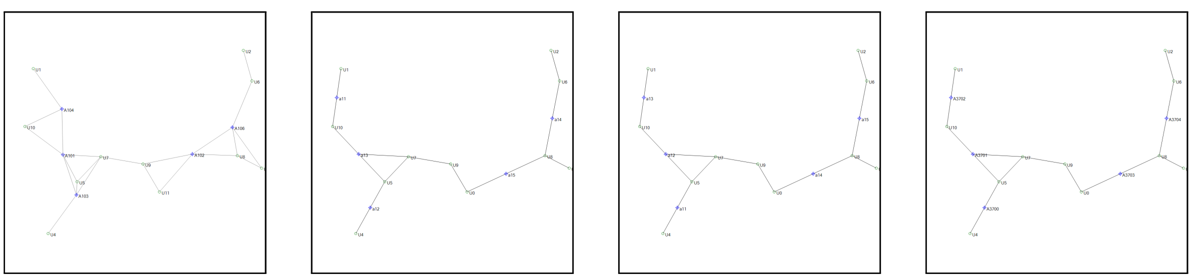


Figure 6.1: Proportion of runs where a.) the Reduction method performs as well as the math model and b.) the Reduction method performs better than the MST method

topology. The mathematical model's solution supports the idea that there is no improvement as almost all these runs also have $|P|_{Math} = |P|_{MST} = \mu_{|P|_{Reduction}}$. This is a testament to the goodness of the MST, MMST, and Reduction methods, as they can sometimes find an optimal $|P|$ connected solution.



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

Figure 6.2: Similar solutions a.) mathematical model solution with $|A|_{Math} = 5$ b.) MST solution with $|A| = 5$ c.) MMST solution with $|A| = 5$ d.) Reduction solution with $|A| = 5$

The Reduction method determined a connected network solution significantly faster than the mathematical model. Each run of four replications of the Reduction method, run on a personal computer (PC), was faster than an exact search on the HPC. Table 6.2

compares the total computation time of the two methods for a few problem instances where $|U| = 10$. The full table of computation times is found in Appendix Table C.2.

Table 6.2: Sample Static Problem Computation Time

Graph	$\lvert U \rvert$	**Math** (HPC hh:mm:ss)	**Reduction** (PC seconds)
5-000	5	0:00:02	6.830
5-008	5	0:00:19	9.833
5R-00	5	0:00:35	3.608
5R-01	5	0:02:35	4.862
10-13	10	>30:00:00	0.724
10-34	10	1:11:35	1.627

6.2 Dynamic Problem

Network metrics, $|A_t|$, $ANCU$, and B_t (when applicable), were recorded at each time step. Each metric with the addition of the weighted objective function, OFV, was used to evaluate the performance of the network. To validate using the Plan method in the deterministic environment, its performance was compared to a mathematical model solution based on the weighted sum of $|A_t|$ and $ANCU$. Recall, that B_t was not included in the mathematical model. The resultant deterministic environment Plan solutions were then compared to the Reactive method. The Reactive method defined in this dissertation is a deterministic solution process, see Section 5.4.2, requiring only one replication for a given instance/realization. Lastly, the performance of the usage of the two methods in a stochastic environment was compared. Table 6.3 shows the experimental design for the dynamic problem.

6.2.1 Deterministic Validation

A mathematical model was developed for CPLEX on Auburn University's HPC for validation of dynamic deterministic problems. Its performance was compared with the Plan method. The goal of this experiment is to determine if the Plan method can perform as well

Table 6.3: Dynamic Experimentation Design

Size_tour	**Deterministic** Math	Plan	Reactive	**Stochastic** MANET	Reactive
S_RAN	⋄	⋄			
S_SAR		•	•	○	□
S_TRA		•	•	○	□
M_PAT		•	•	○	□
M_POL		•	•	○	□
M_RAN		•	•	○	□
M_SAR		•	•	○	□
M_TRA		•	•	○	□
L_SAR		•	•	○	□
X_POL		•	•	○	□

⋄Validation only : $|U_t| < 8$. •10 runs of 3 replications.
○For best Plan solution per run, 5 realizations of 3 replications.
□For best Plan solution per run, 5 realizations. 1 deterministic replication

as the mathematical model. Define the hypothesis:

$$H_0 : \mu_{OFV_{Plan}} = OFV_{Math} \tag{6.2}$$

where $OFV = \gamma_1|A_t| - \gamma_2 ANCU$. The tested problems were small to accommodate the mathematical model's computation time requirements. The dynamic problem's mathematical model is more computationally difficult than a similar sized static problem. The mathematical model must solve multiple static problems with the inclusion of the movement constraints, interdependence, and precedence of assignments between time steps.

Fifteen problems were evaluated using the math model and the Plan method. For these problems, each user node had a random tour of length three with a random starting location in a 1000×1000 km area of operation. The movement rate (used to define the polar tour) and connection radius varied between problems. The number of user nodes ranged from three to ten.

Table 6.4 lists a sample of the results of the Plan and math model solutions. The complete listing of validation results is provided in the Appendix, Table C.3. Each run of the Plan method (consisting of three replications) had a low standard deviation for each metric. This is indicative of the convergence of the pre-plan with RTI and genetic algorithm assignment methods to good solutions within a run. For twelve of the fifteen runs $\sigma_{|A_t|_{Plan}} = \sigma_{ANCU_{Plan}} = \sigma_{OFV_{Plan}} = 0$; the t-statistic could not be calculated. For the three runs where $\sigma_{OFV_{Plan}} > 0$, the t-statistic determined statistical differences.

Table 6.4: Sample Dynamic Deterministic Validation Results

Graph	$\mu_{\|A_t\|}$	$\sigma_{\|A_t\|}$	μ_{ANCU}	σ_{ANCU}	μ_{OFV}	σ_{OFV}	$\|A\|_{Math}$	$ANCU_{Math}$	OFV_{Math}
v_01	2.0	0.0	3.0	0.0	-25.0	0.0	2	3	-25
v_02	5.0	0.0	3.0	0.0	-22.0	0.0	5	3	-22
v_15	7.0	1.0	5.0	0.0	-38.0	1.0	4	4	-32
v_16	10.0	1.0	7.0	0.0	-53.0	1.0	8	8	-64
v_21	8.3	0.6	5.0	0.0	-36.7	0.6	9	5	-36

When $\sigma_{OFV_{Plan}} = 0$, then $\mu_{OFV_{Plan}}$ was assumed to be constant and compared directly to OFV_{Math}; they were only equal or not equal. The result of each of the fifteen runs of the Plan method when compared to the math model fit one of five categories:

Outcome 1: Reject H_0 and the math model performed better

Outcome 2: Do not reject H_0

Outcome 3: $\sigma_{OFV_{Plan}} = 0$ and Plan performs as well as the math model

Outcome 4: $\sigma_{OFV_{Plan}} = 0$ and Plan performs better

Outcome 5: $\sigma_{OFV_{Plan}} = 0$ and the math model performs better

Cases 2, 3, and 4 support the null hypothesis that the Plan method performs as well as or better than the math model. Figure 6.3 shows the proportion of runs of each outcome. The bracket to the right of each metric groups these run outcomes that support the null hypothesis. The OFV outcomes show that the Plan method does as well or better than the math model in 66.7% (10/15) of the runs. Though the OFV is the primary metric for comparing solution methods, it is promising to see that the Plan method did well in minimizing $|A_t|$ and maximizing $ANCU$ when compared to the math model. In a majority of the runs, the Plan method did perform as well as the math model.

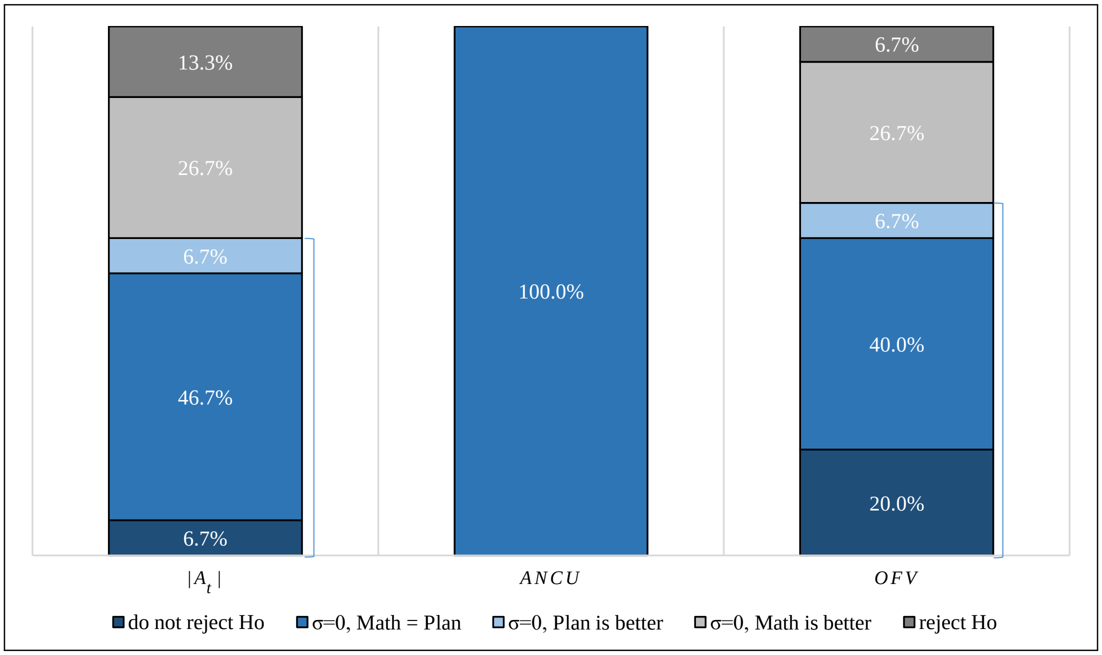


Figure 6.3: Dynamic Deterministic Validation Results

Alternatively, because in many runs the variation in metric scores within a run of the Plan method was zero, the ratio $\frac{Plan}{Math}$ provides a means for comparison of each metric. Figure 6.4 shows the proportion of runs that validate the Plan's performance compared to the math model. Here, there are three outcomes; outcomes 1 and 2 support the null hypothesis.

Outcome 1: Plan performed better ($\frac{Plan}{Math} < 1$)

Outcome 2: Plan performed as good as the math model ($\frac{Plan}{Math} = 1$)

Outcome 3: Math model performed better ($\frac{Plan}{Math} > 1$)

The bracket to the right of each metric bar groups the run outcomes that support the null hypothesis. The primary objective to minimize $|A_t|$ was better or equal to the math model in 60% (9/12) of runs ($\frac{\mu_{|A_t|_{\text{Plan}}}}{|A_t|_{\text{Math}}} \leq 1$). The $ANCU$ is the secondary objective and the Plan method found solutions as good as the math model in all runs ($\frac{\mu_{ANCU_{\text{Plan}}}}{ANCU_{\text{Math}}} = 1$). The OFV is the primary metric by which the solution methods are compared had a resultant 80.0%

(12/15) of the runs where $\frac{\mu_{OFV_{\text{Plan}}}}{OFV_{\text{Math}}} \leq 1$. In many cases the performance of the Plan method is just as good as the math model.

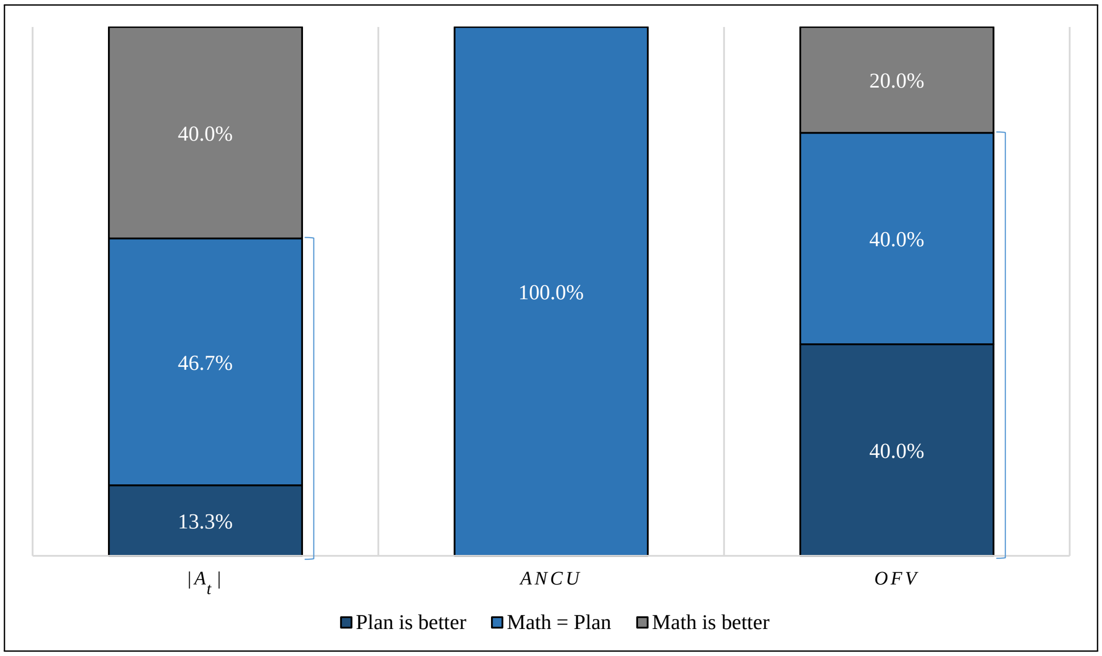


Figure 6.4: Dynamic Deterministic Validation Results : Ratios

The computation time required to determine math and Plan solutions are shown in Table 6.5. The computation time for the math method was recorded from Auburn University's HPC. The time recorded for the Plan method are the sum of three replications on a PC. It can be seen that the computation time of the Plan method is significantly lower, yielding comparable results (as shown in the previous section). Even in cases where the Plan solution requires more time than the math solution, the computer power difference between an HPC and PC should mitigate this difference.

Note that the large variation in the math model computation time, ex. graph v_21, is a result of problem size. The resultant solution for graph v_21 required almost double the number of agent nodes needed for any other graph. But the increase in search space, number of variables, and constraints is non-linear with number of user/agent nodes.

When comparing the primary and secondary objectives, in a majority of the runs, the Plan method performed as well as the math model. Comparison of the OFV metric also

Table 6.5: Dynamic Deterministic Validation Computation Time

Graph	Math (h:mm:ss)	Plan (m:ss)
v_01	0:00:08	1:09
v_02	3:21:25	1:35
v_05	1:06:06	1:36
v_06	0:00:01	1:18
v_07	2:33:14	1:34
v_08	0:10:55	1:35
v_09	0:09:53	1:53
v_10	0:02:44	1:59

Graph	Math (h:mm:ss)	Plan (m:ss)
v_11	0:00:13	2:26
v_12	0:34:33	1:52
v_15	0:07:46	0:02
v_16	0:26:16	0:11
v_19	1:27:06	3:03
v_20	0:12:12	2:16
v_21	11:34:27	0:03

shows that the Plan method performs as well as the math model in most runs. The Plan method was shown to have a significantly lower computation time requirement. With the combination of good comparative performances and significantly lower computation time, the Plan method is a sufficient solution process.

6.2.2 Deterministic Experimentation and Analysis

An average of the runs of the Plan method in a deterministic environment was compared to the single replications of the Reactive method to determine if planning is beneficial. In such a case, the null hypothesis should be rejected. Define:

$$H_0 : \mu_{OFV_{Plan}} = OFV_{Reactive} \tag{6.3}$$

where $OFV = \gamma_1|A_t| - \gamma_2 ANCU$. Because the DMMST method used as the reactive positioning is a deterministic solution method, only a single replication is required. However, each replication of the Plan method potentially yields a different size $|A_t|$. For a fair comparison, $|A_t|_{\text{Reactive}}$ was set to the size of $|A_t|_{\text{Plan}}$ and the Reactive method was evaluated for each replication of the Plan. The Reactive method was evaluated for each replication of the Plan.

To compare the Plan and Reactive method, because of the aforementioned factors, only the $ANCU$ and B_t metrics were considered. The OFV was the primary measure of comparison and was defined as the weighted sum of these two values. Similar to the OFV for the assignment problem, $|A_t|$ and $noia$ which are not applicable in the simulation environment were excluded. For each run for each metric, the mean and standard deviation, μ and σ were determined. Table 6.6 lists a sample of the obtained values from both methods. The entire list of results for the dynamic deterministic experimentation is found in Table C.1.

Table 6.6: Sample Dynamic Deterministic Experimentation Results

	Plan						**Reactive**					
	$ANCU$		B_t		OFV		$ANCU$		B_t		OFV	
Graph	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
S_SAR_1002	4	0	0.90	0.01	-567.7	0.23	3.98	0.01	0.95	0.0	-567.9	1.83
S_SAR_1003	4	0	0.88	0.01	-566.9	0.66	3.92	0.06	0.94	0.0	-559.2	7.14
M_POL_1001	10	0	0.92	0.01	-773.1	0.15	9.19	0.14	0.97	0.0	-742.2	10.88
M_POL_1002	10	0	0.92	0.01	-772.9	0.15	9.40	0.04	0.97	0.0	-758.2	3.45
X_POL_1009	60	0	0.91	0.00	-4703.8	0.05	58.37	0.08	0.98	0.0	-4578.6	6.25
X_POL_1010	60	0	0.92	0.00	-4704.1	0.11	58.23	0.10	0.98	0.0	-4567.6	7.94

Five outcomes were defined based on the results of a t-test. Figure 6.5 shows the proportion of each outcome.

Outcome 1: Reject H_0 and MANET is better

Outcome 2: Do not reject H_0

Outcome 3: Reject H_0 and Reactive is better

Outcome 4: $\sigma_{OFV_{Plan}} = 0$ and Plan performs better

Outcome 5: $\sigma_{OFV_{Plan}} = 0$ and Reactive performs better

There was a statistical difference in $ANCU$, favoring the Plan method, in 63.3% (57/90) of the runs. Contrarily, there were 33.3% (30/90) of runs where the $ANCU$ for the Plan and Reactive method were statistically equivalent but with 27/30 of the runs where the Reactive solution was not connected at all time steps and the Plan solution was. For all runs the Reactive method had a better robustness score, but was not as well connected. Recall that

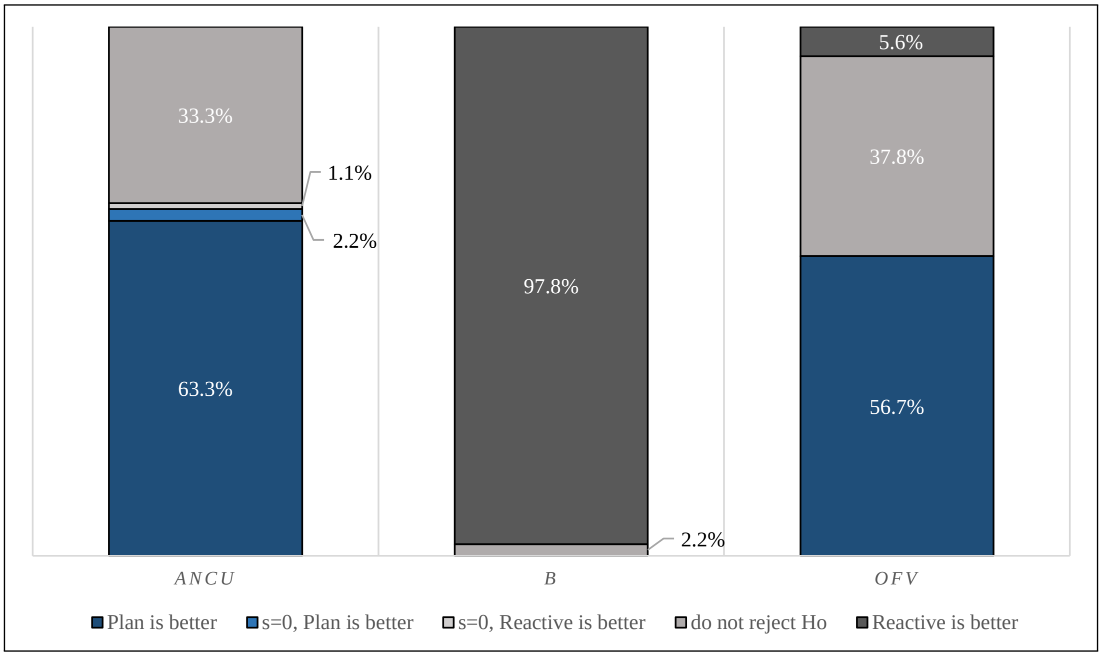


Figure 6.5: Dynamic Deterministic Experimentation Results

the Assortativity metric, used here to measure robustness, has a few cases that lead to false positive-like results. For example, a network with several disconnected sub-networks may have a better B_t score but would have poor overall network connectivity.

The Plan method performed (strictly) better for only 56.7% (51/90) of runs when comparing the OFV metric. There is a large proportion of runs, 37.8% (34/90), that are statistically equivalent. But, in 29/40 of these runs the Plan method maintains full connectivity and the Reactive method does not. Similarly, 5.6% (5/90) of runs resulted in statistically equivalent Plan and Reactive solutions but in 4/5 runs the Reactive solution was not connected at all time steps and the Plan solution was. If the number of runs where the Plan and Reactive methods are statistically equivalent but the Reactive solution is sub-optimal ($ANCU < |U_t|$) is combined with the number of runs where the Plan is statistically better, then the MANET performs better in 93.3% (84/90) of all runs.

There were 4.5% (4/90) of the runs where all three replications resulted in infeasible solutions, $ANCU < |U_t|$. For these, the genetic algorithm did not converge to a feasible solution within the alloted 10,000 iterations. Improving solutions were found well into the

9,000s. Additional time was given but the algorithm did not converge to a feasible solution. To determine if there is a solution given $|A_t|$ a combinatorial assignment mathematical model could be developed. If there is a solution to the mathematical model, the the genetic algorithm has failed. If there is no feasible solution with $|A_t|$ an infeasible solution could be fixed by adding agent nodes until a feasible solution could be found.

Though there were a few cases where the genetic algorithm was unable to determine a connected solution, approximately 77% (69/90) of runs resulted in $\sigma_{ANCU_{Plan}} = 0$ and $ANCU = |U_t|$ indicating the genetic algorithm converged to a fully connected solution. This is expected given the 100% service requirement. The maximum coefficient of variation for the robustness metric was small, $c.v._{B_{t\,Plan}} = 0.07$. Similarly for the OFV, incorporating incorporating both the $ANCU$ and B_t, the maximum coefficient of variation was also small $c.v._{OFV_{Plan}} = -0.4$. The low degree of variation within a run shows the genetic algorithm's convergence to good solutions.

Generally, for each problem type-instance, the Plan method required significantly more computation time than the Reactive method. For the comparison of the Plan and Reactive methods, the computational effort is noted but not used to determine the superiority of either method. All runs in this comparison were done on a high-end personal computer [54]. The average computation time per run for each problem instance and each solution method is provided in Appendix Table C.5.

For a decision maker, the computation time could be the deciding factor in determining which method to use. As was discussed previously, there is a large portion of runs where there was no statistical difference in $ANCU$ or OFV but they were infeasible. The decision maker would have to determine if it is worth the compromise to use the Reactive method over the Plan method. That is to say, if 100% connectivity is required, then the computational time required of the MANET method is worth its usage.

It can be concluded that the Plan method performed better than the Reactive at the cost of computational effort. The improvement in performance is even more pronounced because the same mechanism used in the Reactive method, the DMMST, is an underlying function of the Plan method.

6.2.3 Stochastic

After obtaining a solution from the Plan method, its usage in the stochastic environment is evaluated. Recall that 4.4% (4/90) of the runs were infeasible, $|ANCU| < |U_t|$, and were not considered in the following experimentation.

Comparison of the Plan and Reactive Methods

In the stochastic environment where user nodes can deviate from their tours, it is possible for the network to become disconnected. The control node attempts to mitigate this by modifying the assigned agent node tours. It is expected that MANET will perform better than the Reactive method in the stochastic environment, *i.e.* reject H_0 (Equation 6.3).

The best found replication per run (90 Plan solutions [Level 1]) was used to generate five stochastic instances of the given map [Level 2] and tested across three replications of the MANET method [Level 3]. The Reactive method is deterministic, thus it only needs to be run once per realization.

The $ANCU$, B_t, and OFV metrics are compared. The $|A_t|$ is constant within a run. Thus, the $ANCU$ metric is supplemented with the robustness metric to determine its value. Table 6.7 provides a sample of the results. Here, the graph label is the concatenation of the problem type, instance, best Plan solution (per run), and stochastic path identification numbers.

A one sample two-tailed t-test was used to compare the three replications of the MANET (Level 3: 3001-3003) with the Reactive method. There were five outcomes when the MANET runs were compared with a Reactive solution.

Outcome 1: Reject H_0 and MANET performs better

Outcome 2: Do not reject H_0

Outcome 3: Reject H_0 and Reactive performs better

Outcome 4: $\sigma_{OFV_{MANET}} = 0$ and MANET performs better

Outcome 5: $\sigma_{OFV_{MANET}} = 0$ and Reactive performs better

Figure 6.6 shows the proportion of runs of each of these outcomes when comparing the MANET and Reactive methods. Outcomes 1 and 4 support the rejection of the null hypothesis, that the MANET method performs better than the Reactive method. The bracket to the right of each metric bar groups these run outcomes.

Comparing the $ANCU$, in 77.4% (333/430) of runs the MANET performed better. This shows that using this pre-plan, assign, and possible re-planning is beneficial in keeping the network connected a majority of the runs. The MANET had a better value of B_t for 68.8% (296/430) of the runs. Using this number alone can be misleading when there are highly connected subnetworks and not a fully connected network. The primary metric, OFV, shows that the MANET method performed better in 79.5% (342/430) of all runs. Because the weighted OFV has $ANCU$ as a sub-component, the performance rate of these two are very similar. The portion of runs where the Reactive method does as well as the MANET model (outcome 2) can be based on the inability of MANET method maintain full connectivity in a stochastic environment which can be expected.

However, there remains 14.2% (61/430) of the runs where the Reactive method performs better than the MANET when comparing the OFV. These results do not support the

Table 6.7: Sample Dynamic Stochastic Simulation Results

	MANET						**REACT**		
	$ANCU$		B_t		OFV				
Graph	μ	σ	μ	σ	μ	σ	$ANCU$	B_t	OFV
S_SAR_01-1002-1403296479	3.86	0	0.944	0	-551.54	0	-548.03	0.93	3.84
S_SAR_01-1002-241471907	4.00	0	0.944	0	-569.53	0	-562.99	0.93	3.95
S_SAR_01-1002-253537267	3.82	0	0.945	0	-545.59	0	-551.12	0.93	3.86
M_TRA_01-1003-1397660399	20.00	0	0.946	0	-853.25	0	-853.28	0.95	20.00
M_TRA_01-1003-147115108	19.36	0	0.947	0	-826.25	0	-826.27	0.95	19.36
X_POL_10-1001-90240878	60.00	0	0.997	0	-4705.91	0	-4663.87	0.99	59.46
X_POL_10-1001-963671391	59.77	0	0.997	0	-4687.92	0	-4624.87	0.99	58.96

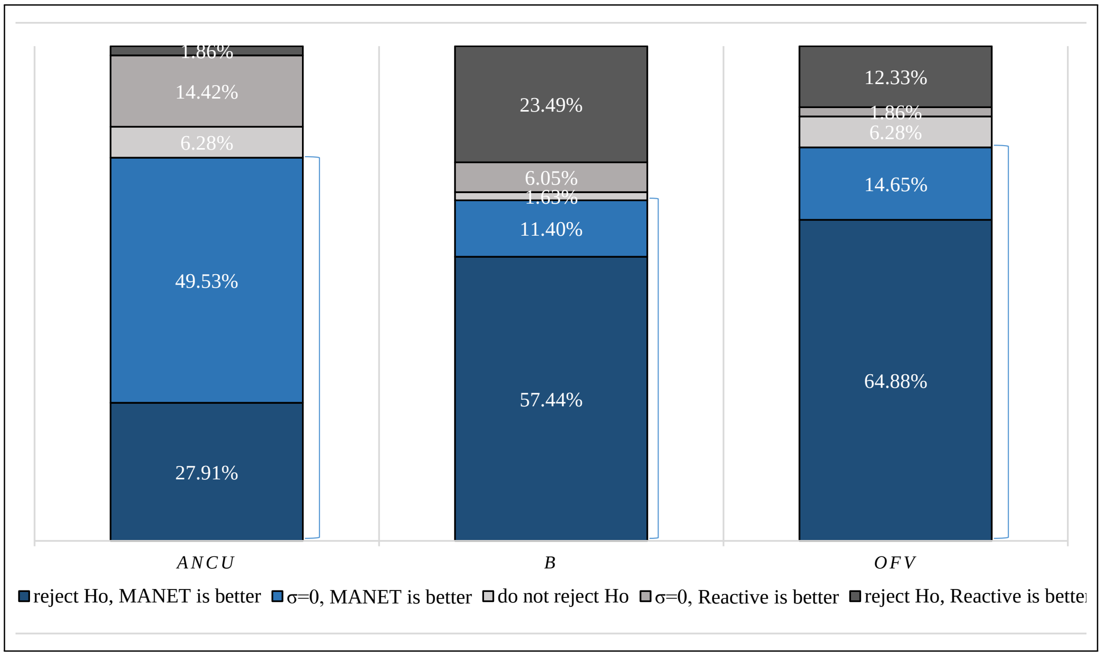


Figure 6.6: Dynamic Stochastic Experimentation Results

objective of this dissertation to assert that the MANET method performs better than the Reactive method. These runs also have $ANCU_{Reactive} > ANCU_{MANET}$, a loss of connectivity of user nodes. Figure 6.7 shows the proportion of runs that had either outcomes 3 or 5, divided by type. It is evident that the M_TRA problem type is difficult for the MANET to solve.

For the M_TRA problem type 62% (31/50) of its runs result in the Reactive outperforming the MANET method. And, it accounts for 50.8% (31/61) of all runs where Reactive performs better. These maps have a medium number of user nodes ($|U_t| = 20$) in a small area (3.2×3.2km). The mission time averages about 18 time steps. The resultant agent node set is small ($|A_t| \sim 5$). This type may be problematic in analyzing the difference between the MANET and Reactive methods in a stochastic environment because the network remains well connected most of the time, even with multiple node deviations.

When the network is determined to be deviating, the deviation parameters have an effect on the responsiveness of the MANET method. The Δ_{replan} parameter had a large effect on these poor performing runs. For any value of $\Delta_{replan} > 0\%$ it is possible to not re-plan for a disconnected forecast network state if the amount of deviation does not exceed this threshold. Also, the re-plan cool down rule (see Heuristic 3) makes it possible to have a disconnected network before allowing the next re-plan. Not allowing consecutive re-plans was incorporated to prevent re-planning at each step, which could be interpreted as a Reactive method.

In a similar way, the M_SAR type resulted in 16.4% (10/61) of the runs in the third case. These maps have a medium number of user nodes ($|U_t| = 20$) in a small area (6.9 x 6.9 km). The resultant number of agent nodes is approximately 40. This network is also somewhat dense and deviations are absorbed because of this. Further analysis of the runs where the Reactive method is statistically better seems to indicate that the problem type and selected re-planning parameters are important.

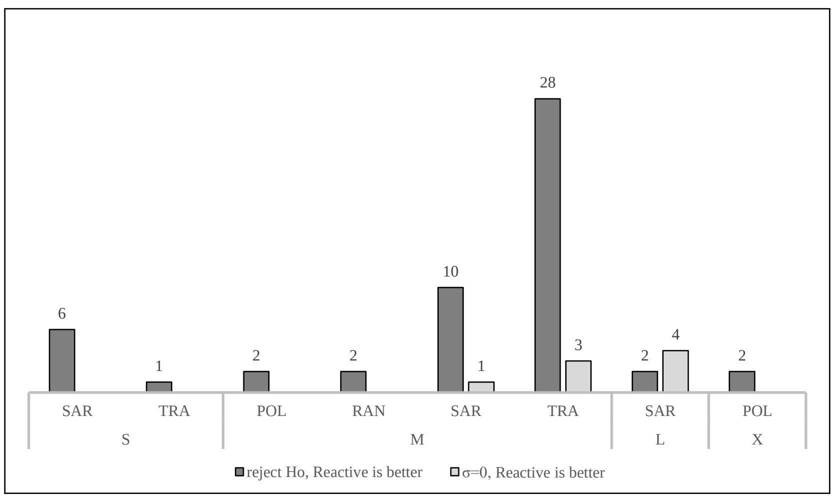


Figure 6.7: Dynamic Stochastic Experimentation Where Reactive is Superior by Type

Computation time for each stochastic environment type is found in the Appendix, Table C.7. It lists the average computation time of the runs (three replications) of five realizations of the best found Plan solution. The type of problem affects the computation time. Generally, when solving the SAR type problems (of S, M, and L size class) the computation time of the MANET and Reactive methods are comparable. The MANET method computation time is approximately ten times that of the Reactive method for TRA type problems (regardless of size). The M_PAT problem type also shows that the MANET method requires about ten times more computation time. The remaining types require at most about five times more computation time to solve with the MANET compared to the Reactive method. This shows that for some problem types the Reactive method should not be discounted, as it does provide good results for a significant portion of the stochastic environment runs.

For a decision maker, the computation effort required to use the MANET has to be compared with the connectivity requirements of the system. If 100% connectivity is required, the increase in computation time would be justified by typically better network management. Additionally, the problem type may influence the selection of solution method. Recall the definition of the time-steps t and its relationship to a real-world time interval τ. When the computation time of a positioning method is less than τ, its use is justified. For example, search and rescue missions may suffice with large values of τ, such as hourly updates, allowing for complex positioning methods like the MANET. Alternatively, military missions (TRA, PAT, and POL types) may require smaller values of τ requiring fast generation of solutions.

6.3 Chapter Summary

The effectiveness of the Reduction method as a static problem solution method has been validated with a mathematical model and shown to be better than methods provided in the literature. For the dynamic deterministic environment problem, the Plan method, pre-plan with the RTI method and assign with a genetic algorithm, has been show to efficiently determine minimum $|A_t|$ connected solutions. Also validated with a mathematical model, the Plan method maintains network connectivity better than the Reactive method. In the instance where the Plan method returns an infeasible solution and the service level requirement is 100%, the decision maker can manually add agent nodes assigning them the locations that connect the network. With user node deviations in the stochastic environment, the MANET method, using the Plan method with the possibility to re-plan, was shown in most runs to perform better than the Reactive method, though at a higher computational cost.

Chapter 7

Conclusions and Research Extensions

The primary research objective for this dissertation was to develop a method for and evaluate the effectiveness of using pre-deployment planning in a mobile ad-hoc network. This was done by measuring/evaluating a network's state at discrete time steps as it progressed. Using planning information was hypothesized to result in better performance when compared to a purely reactive method.

7.1 Static Problem Research Objectives

The static problem was to determine agent node locations to connect a disconnected network while minimizing $|P|$. The Reduction method starts by connecting the network with the MMST solution. It iteratively removes a random connecting node and repositions the remaining nodes using the DMMST. The resultant network is either the MMST solution or one with fewer $|P|$ that connects the network.

To validate the Reduction method a modified version of Konak *et al.*'s ALOC [34] formulation of a network as a maximum flow problem was solved in CPLEX. This method yielded optimal minimum $|P|$ solutions but at the cost of computation time. In almost all runs the Reduction performed as well as the exact mathematical model to connect a given network and minimize $|P|$. In addition, a run of the Reduction method is significantly faster than the mathematical model.

A secondary objective of this dissertation was to provide a better static problem position solution than was found in the literature (Lin and Xue's method [40]). From the testing in this dissertation, the Reduction method performed better than the Lin and Xue's [40].

7.2 Dynamic Problem Research Objectives

Based on the Reduction method, the Reduction per Time Interval (RTI) method solved sequential static problems to determine a set of connecting points for each time step, P_t. A genetic algorithm determined assignments of points in P_t to the set of agent nodes A_t (where $|A_t| = \max_t P_t$). The genetic algorithm measures $|A_t|$, $ANCU$, B_t, and $noia$ (number of infeasible assignments). The B_t metric is defined by a normalized Assortativity score. Most of the assignment solutions had a minimized $|A_t|$, equal to the mathematical model solutions, and always connected the network ($ANCU = |U_t|$) with only 4.5% (4/90) of the runs where no feasible solution was found ($ANCU < |U_t|$).

The Reactive method forecasts user node positions and directs agent nodes to connecting points determined by the DMMST method. It does not have the ability to determine the size of the agent node set. Thus, for the Reactive method, $|A_t|$ is set a priori. In this dissertation, for an equal basis of comparison, the Reactive method experimentation was conducted after the Plan method so that $|A_t|_{Reactive}$ could be set to $|A_t|_{Plan}$.

In the stochastic environment, in addition to the aforementioned metrics, user node deviations were also measured. Re-planning was done for future time steps when the forecast network topology was disconnected and deviating. Using the Plan with the possibility of re-planning was defined as the MANET method. For a majority of the runs, the MANET method outperformed the Reactive method.

There was a portion of runs where the Reactive method performed better than the MANET method. Part of this was attributed to the problem type. Networks that had dense node clumping, even during user node deviating events, remained well (though not necessarily fully) connected. As a result, the MANET method had a low rate of re-planning (approximately once during a mission). The Reactive method, however, reacts in accordance with the forecast network at each time step. The inability of the MANET method to maintain a better connected network than the Reactive method can be attributed to the limitations defined for re-planning (*i.e.* re-plan threshold, re-plan cooldown, no re-plan if forecast network is connected). Also, the potential deficiency in the genetic algorithm implementation could be a factor, considering its inability to find feasible solutions for the 4.4% of the dynamic deterministic runs.

Barring this unforeseen anomaly caused by the combination of the problem type definition and re-plan parameters, a majority of the stochastic environment simulation runs showed a benefit from planning and re-planning. This benefit is further supported by the fact that the Plan method relies on the DMMST method that is used as the Reactive method. Using pre-deployment planning to maintain network connectivity has been shown to be beneficial when compared to the Reactive positioning method.

7.3 Future Work

There are several modifications to the existing dissertation that could be investigated. The genetic algorithm could benefit from modifications to population reproduction, mutation and culling methods. A few parameters could be changed or eliminated to investigate the performance of the MANET. For example, the re-plan three time step cool-down parameters could be eliminated allowing for re-plans as often as possible. As was discovered post-experimentation, the user node tour definition can negatively impact the performance of the planning searches. New tour types should be developed or alternatively obtain empirical mobile network data.

Incorporating a reactive method into the MANET may be beneficial. This would require an additional threshold, ex. $\Delta_{reactive}$, that would define when the agent nodes should be explicitly directed instead of following a (modified) tour. For example, if the network forecast is not fully connected and is either not deviating enough for re-plan or in re-plan cool-down, the Reactive method could be employed. The re-plan method could be used when the network deviates beyond the threshold Δ_{replan}, or if the Reactive method is used too frequently indicating a need to re-plan. Combining a reactive method with agent node tours, could mitigate the shortcomings of having a re-plan cool-down time.

This dissertation seeks to determine the minimum number of agent nodes to connect the network over the course of the simulation. This is based on the assumption that an unlimited, though minimized, number of agent nodes can be obtained. If $|A_t|$ is limited then several modifications to the existing formulation would be needed. Initially, determining the maximum value of $|P_t|$ needs to be addressed. The deterministic environment pre-plan methodology would then seek to determine locations of connecting points to maximize connectivity with a constraint on P_t. The constraint requiring 100% connectivity has to be replaced with a minimum service goal. The assignment method would then be very similar to what has been defined but with a constraint on $|A_t|$.

As an example, consider a network with a tight constraint on $|A_t|$. The likelihood of partially connected networks is high. If the resultant $ANCU$ is about 50% of all nodes, then each solution is likely to have a different assignment of points in P_t to Q_t. The assignment method solutions would likely have a higher variability between replication. Similarly, evaluating the MANET method in the stochastic environment could result in higher variability per replication. The run size (number of replications) should be adjusted for both the assignment and MANET methods to accommodate this likely increase in variability.

Bibliography

[1] Fireteam. online. last modified: 5 June 2013 at 22:14, last accessed: 2013-06-20 13:30.

[2] Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2(1):1–22, 2004.

[3] Hamid Ali, Waseem Shahzad, and Farrukh Aslam Khan. Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization. *Applied Soft Computing*, 12(7):1913–1928, 2012.

[4] Gianluca Antonelli, Filippo Arrichiello, Stefano Chiaverini, and Roberto Setola. A self-configuring MANET for coverage area adaptation through kinematic control of a platoon of mobile robots. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1332–1337. IEEE, 2005.

[5] Gianluca Antonelli, Filippo Arrichiello, Stefano Chiaverini, and Roberto Setola. Coordinated control of mobile antennas for ad-hoc networks in cluttered environments. In *IAS*, pages 235–242, 2006.

[6] Christian Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 80–91. ACM, 2002.

[7] Christian Bettstetter. On the connectivity of ad hoc networks. *The Computer Journal*, 47(4):432, 2004.

[8] Satyabrata Chakrabarti and Amitabh Mishra. *Quality of Service in Mobile Ad Hoc Networks*, chapter 3, pages 3–1 through 3–29. CRC Press, 2003.

[9] Shampa Chakraverty, Arvind Batra, and Aman Rathi. Directed convergence heuristic: A fast novel approach to Steiner tree construction. In *Very Large Scale Integration, 2006 IFIP International Conference on*, pages 255 –260, oct. 2006.

[10] Donghui Chen, Ding-Zhu Du, Xiao-Dong Hu, Guo-Hui Lin, Lusheng Wang, and Guoliang Xue. Approximations for Steiner trees with minimum number of Steiner points. *Journal of Global Optimization*, 18(1):17–33, 2000.

[11] Hui Cheng and Shengxiang Yang. Genetic algorithms with elitism-based immigrants for dynamic shortest path problem in mobile ad hoc networks. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 3135–3140. IEEE, 2009.

[12] Younchol Cho. *Optimizing military tactical MANETs efficiently using PSO*. PhD thesis, Auburn University, 2009.

[13] Younchol Cho, Jeffrey S. Smith, and Alice E. Smith. Optimizing tactical military MANETs with a specialized pso. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–6, July 2010.

[14] Cem Şafak Şahin. Self-deployment of mobile agents in manets for military applications. Technical report, Defense Technical Information Center, 2008.

[15] Cem Şafak Şahin, Elkin Urrea, M. Ümit Uyar, Michael Conner, Giorgio Bertoli, and Christian Pizzo. Design of genetic algorithms for topology control of unmanned vehicles. *International Journal of Applied Decision Sciences*, 3(3):221–238, 2010.

[16] Joseph Czyzyk, Michael P. Mesnier, and Jorge J. Moré. The neos server. *IEEE Journal on Computational Science and Engineering*, 5(3):68–75, 1998.

[17] Orhan Dengiz. *Maximizing Connectivity and Performance In Mobile Ad-Hoc Networks Using Mobile Agents*. PhD thesis, Auburn University, 2007.

[18] Orhan Dengiz, Abdullah Konak, and Alice E. Smith. Connectivity management in mobile ad hoc networks using particle swarm optimization. *Ad Hoc Networks*, 9(7):1312–1326, 2011.

[19] Elizabeth D. Dolan. The neos server 4.0 administrative guide. Technical Memorandum ANL/MCS-TM-250, Mathematics and Computer Science Division, Argonne National Laboratory, 2001.

[20] Ding-Zhu Du. On Steiner ratio conjectures. *Annals of Operations Research*, 33:437–449, 1991. 10.1007/BF02071981.

[21] FDNY and McKinsey Report. *Executive Summary*. Fire Department of New York, 2002.

[22] Robert H. Frenkiel. Cellular radiotelephone system stuctured for flexible use of differnt cell sizes, 1979.

[23] M.R. Garey, R.L. Graham, and D.S. Johnson. The complexity of computing Steiner minimal trees. *SIAM Journal on Applied Mathematics*, pages 835–859, 1977.

[24] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):pp. 1–29, 1968.

[25] Michel X. Goemans and Young-soo Myung. A catalog of Steiner tree formulations. *Networks*, 23(1):19–28, 1993.

[26] R.L. Graham and Pavol Hell. On the history of the minimum spanning tree problem. *IEEE Annals of the History of Computing*, 7:43–57, 1985.

[27] William Gropp and Jorge J. Moré. Optimization environments and the neos server. In Martin D. Buhman and Arieh Iserles, editors, *Approximation Theory and Optimization*, pages 167 – 182. Cambridge University Press, 1997.

[28] Kyle Guan, Reza Ghanadan, Sintayehu Dehnie, and Shaomin Mo. Optimal platform placement and configuration in networked electronic warfare (EW). In *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, pages 1019 –1024, 31 2010-nov. 3 2010.

[29] Zhu Han, A. Lee Swindlehurst, and K.J. Ray Liu. Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles. *Vehicular Technology, IEEE Transactions on*, 58(7):3533–3546, 2009.

[30] M. Ani Hsieh, Anthony Cowley, Vijay Kumar, and Camillo J. Taylor. Maintaining network connectivity and performance in robot teams. *Journal of Field Robotics*, 25(1-2):111–131, 2008.

[31] Robert A. Hunjet, Andrew Coyle, and Matthew Sorell. Enhancing mobile adhoc networks through node placement and topology control. In *Wireless Communication Systems (ISWCS), 2010 7th International Symposium on*, pages 536–540. IEEE, 2010.

[32] Seong Hwan Kim, Byoung Hoon Jung, and Dan Keun Sung. Adaptive beamforming antenna scheme to minimize the interference in a unmanned aerial vehicle (UAV) MANET. In *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pages 813–817. IEEE, 2009.

[33] Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. *Wireless Networks*, 6:307–321, July 2000.

[34] Abdullah Konak, George E. Buchert, and James Juro. A flocking-based approach to maintain connectivity in mobile wireless ad hoc networks. *Applied Soft Computing*, 13(2):1284 – 1291, 2013.

[35] Abdullah Konak, Orhan Dengiz, and Alice E. Smith. Improving network connectivity in ad hoc networks using particle swarm optimization and agents. In Jeff Kennington, Eli Olinick, and Dinesh Rajan, editors, *Wireless Network Design*, volume 158 of *International Series in Operations Research & Management Science*, pages 247–267. Springer New York, 2011. 10.1007/978-1-4419-6111-2_11.

[36] Charalampos Konstantopoulos, Damianos Gavalas, and Grammati Pantziou. Clustering in mobile ad hoc networks through neighborhood stability-based mobility prediction. *Computer Networks*, 52(9):1797–1824, 2008.

[37] Raghavendra V. Kulkarni, Ganesh K. Venayagamoorthy, Ann Miller, and Cihan H. Dagli. Network-centric localization in MANETs based on particle swarm optimization. In *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*, pages 1–6. IEEE, 2008.

[38] Janusz Kusyk, Elkin Urrea, Cem Şafak Şahin, and M. Ümit Uyar. Game theory and genetic algorithm based approach for self positioning of autonomous nodes. *Ad Hoc & Sensor Wireless Networks*, 16(1-3):93–118, 2012.

[39] Janusz Kusyk, Elkin Urrea, Cem Şafak Şahin, M. Ümit Uyar, Giorgio Bertoli, and Christian Pizzo. Resilient node self-positioning methods for MANETS based on game theory and genetic algorithms. In *The 2010 Military Communications Conference - Unclassified Program - Networking Protocols and Performance Track*, 2010.

[40] Guo-Hui Lin and Guoliang Xue. Steiner tree problem with minimum number of Steiner points and bounded edge-length. *Information Processing Letters*, 69(2):53–57, 1999.

[41] Jenn-Wei Lin and Shih-Chieh Tang. Coverage improvement for target tracking in hybrid sensor networks. In *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, volume 4, pages 126 –130, feb. 2010.

[42] Qing Ling and Zhi Tian. Minimum node degree and k-connectivity of a wireless multihop network in bounded area. In *Global Telecommunications Conference, 2007. GLOBE-COM'07. IEEE*, pages 1296–1301. IEEE, 2007.

[43] Priya Mahadevan, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Xenofontas Dimitropoulos, KC Claffy, and Amin Vahdat. The internet as-level topology: Three data sources and one definitive metric. *ACM SIGCOMM Computer Communication Review*, 36:17–26, 2006.

[44] B.S. Manoj, Archana Sekhar, and C. Siva Ram Murthy. On the use of limited autonomous mobility for dynamic coverage maintenance in sensor networks. *Computer Networks*, 51(8):2126 – 2143, 2007.

[45] Mark Manzano, Eusebi Calle, Víctor Torres-Padrosa, Juan Segovia, and David Harle. Endurance: A new robustness measure for complex networks under multiple failure scenarios. *Computer Networks*, 57:3641–3653, 2013.

[46] Jishan Mehedi and MK Naskar. Fuzzy based algorithms for connectivity maintenance in manet: Fuzzy based connectivity management in MANET. 2012.

[47] Ion I. Măndoiu and Alexander Z. Zelikovsky. A note on the MST heuristic for bounded edge-length Steiner trees with minimum number of Steiner points. *Information Processing Letters*, 75(4):165–167, 2000.

[48] Stephen Mussmann and Jennifer Neville John Moore, Joseph J. Pfeiffer III. Assortativity in Chung Lu random graph models. *ACM*, 2014.

[49] Dragoş Niculescu and Badri Nath. Ad hoc positioning system (APS). In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, volume 5, pages 2926–2931 vol.5, 2001.

[50] Dragoş Niculescu and Badri Nath. DV based positioning in ad hoc networks. *Telecommunication Systems*, 22:267–280, 2003. 10.1023/A:1023403323460.

[51] Cevher D ogan, C. Şafak Şahin, M. Ümit Uyar, and Elkin Urrea. Testbed for node communication in MANETs to uniformly cover unknown geographical terrain using genetic algorithms. In *Adaptive Hardware and Systems, 2009. AHS 2009. NASA/ESA Conference on*, pages 273–280. IEEE, 2009.

[52] Cevher D ogan, M. Ümit Uyar, Elkin Urrea, C. Şafak Şahin, and Ibrahim Hokelek. Testbed implementation of genetic algorithms for self spreading nodes in MANETs. This work has been partially supported by the National Science Foundation grants ECS-0421159 and CNS-0619577. Copyright c 2008 City University of New York. All rights reserved., 2008.

[53] Potomac Mangate Group Inc. *A Simple Guild to Donduction Fround Search and Rescue Detection Experiments: Volume I.* U.S. Department of Homeland Security: United States Coast Guard, May 2006. Prepared for U.S. Department of Homeland Security United States Coast Guard.

[54] PPI Engineering & Construction Services, LLC. Computational resources, 2018.

[55] D.G. Reina, S.L. Toral Marn, N. Bessis, F. Barrero, and E. Asimakopoulou. An evolutionary computation approach for optimizing connectivity in disaster response scenarios. *Applied Soft Computing*, 13(2):833 – 845, 2013.

[56] Debdutta Barman Roy, Rituparna Chaki, and Nabendu Chaki. A new cluster-based wormhole intrusion detection algorithm for mobile ad-hoc networks. *Computing Research Repository*, abs/1004.0587, 2010.

[57] Waseem Shahzad, Farrukh Aslam Khan, and Abdul Basit Siddiqui. Clustering in mobile ad hoc networks using comprehensive learning particle swarm optimization (CLPSO). *Communication and Networking*, pages 342–349, 2009.

[58] Waseem Shahzad, Farrukh Aslam Khan, and Abdul Basit Siddiqui. Weighted clustering using comprehensive learning particle swarm optimization for mobile ad hoc networks. *International Journal of Future Generation Communication and Networking*, 3(1), March 2010.

[59] Anand Srinivas, Gil Zussman, and Eytan Modiano. Construction and maintenance of wireless mobile backbone networks. *IEEE-ACM Transactions on Networking*, 17(1):239–252, February 2009.

[60] Ivan Stojmenovic. Position-based routing in ad hoc networks. *Communications Magazine, IEEE*, 40(7):128–134, July 2002.

[61] A. Sydney, C. Scoglio, P. Schumm, and R. E. Kooij. Elasticity: Topological characterization of robustness in complex networks. *Bionetics*, 08:25–28, 2008.

[62] Chai K Toh. *Ad hoc mobile wireless networks: protocols and systems.* Pearson Education, 2001.

[63] Patrick Ulam and Ronald C Arkin. When good communication go bad: communications recovery for multi-robot teams. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 3727–3734. IEEE, 2004.

[64] U.S. Air Force. U.S. Air Force fact sheet: MQ-1B Predator. Technical report, Air Combat Command, Public Affairs Office, http://www.af.mil/information/factsheets/factsheet.asp?fsID=122, 2010. Last Accessed: June 29, 2011.

[65] U.S. Army. *FM 3-21.20 The Infantry Battalion*. Department of the Army Headquaters.

[66] U.S. Army. *U.S. Army Unmanned Aircraft Systems Roadmap 2010-2035: The Implementation of Network-Centric Warfare*. U.S. Army Office of Force Transformation.

[67] U.S. Army. *FM 7-8 Infantry Rifle Platoon and Squad*. Department of the Army Headquaters, April 1992.

[68] U.S. Army. *p10-1 Organization of the United States Army*. Department of the Army Headquaters, June 1994.

[69] U.S. Army. *FM 7-10 The Infantry Rifle Company*. Department of the Army Headquaters, October 2000.

[70] U.S. Army. *FM 6-02.53 Tactical Radio Operations*. Department of the Army Headquaters, August 2009.

[71] U.S. Army. *U. S. Army Roadmap for Unmanned Aircraft Systems - "Eyes of the Army"*. Army UAS CoE, 2010.

[72] U.S. Census Bureau, 2010. http://www.census.gov/.

[73] U.S. Marine Corp. *MCWP 3-11.3 Scouting and Patrolling*. U. S. Marine Corps, April 2000.

[74] Jose Vazquez and Chris Malcolm. Distributed multirobot exploration maintaining a mobile network. In *Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference*, volume 3, pages 113–118. IEEE, 2004.

[75] Jianping Wang, Eseosa Osagie, Parimala Thulasiraman, and Ruppa K. Thulasiram. HOPNET: A hybrid ant colony optimization routing algorithm for mobile ad hoc network. *Ad Hoc Networks*, 7(4):690–705, 2009.

[76] Shu-Ching Wang, Hsin-Hung Pan, Kuo-Qin Yan, and Yu-Lung Lo. A unified framework for cluster manager election and clustering mechanism in mobile ad hoc networks. *Computer Standards & Interfaces*, 30(5):329–338, 2008.

[77] Shun-Sheng Wang, Kuo-Qin Yan, and Shu-Ching Wang. An optimal solution for byzantine agreement under a hierarchical cluster-oriented mobile ad hoc network. *Computers & Electrical Engineering*, 36(1):100–113, 2010.

[78] Wei Wang, Vikram Srinivasan, and Kee-Chaing Chua. Coverage in hybrid mobile sensor networks. *Mobile Computing, IEEE Transactions on*, 7(11):1374 –1387, nov. 2008.

[79] Douglas B. West. *Introduction to Graph Theory.* Prentice-Hall , Inc., second edition, 2001.

[80] Jim Whitehead. *National Land Search Operations Manual.* The National Search and Rescue Council, 7 edition, 2010. Police, State/Territory Emergency Services, the Australian Defence Force and the Australian Maritime Safety Authority.

[81] Bang Ye Wu and Kun-Mao Chao. *Spanning Trees and Optimization Problems*, chapter Minimum Spanning Trees, pages 1–11. Chapman & Hall/CRC Press, USA, 2004.

[82] Kejian Xia, Fei Liu, Jie Zou, and Jie He. Fast neighbor discovery protocol based on MANET-RTLS. In *Advances in Wireless Sensor Networks*, pages 113–121. Springer, 2014.

[83] F. Xue and P.R. Kumar. The number of neighbors needed for connectivity of wireless networks. *Wireless networks*, 10(2):169–181, 2004.

[84] Jin-Tai Yan, Tzu-Ya Wang, and Yu-Cheng Lee. Timing-driven Steiner tree construction based on feasible assignment of hidden Steiner points. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 1370 – 1373 Vol. 2, may 2005.

[85] Zhenqiang Ye, Srikanth V. Krishnamurthy, and Satish K. Tripathi. A routing framework for providing robustness to node failures in mobile ad hoc networks. *Ad Hoc Networks*, 2(1):87 – 107, 2004.

[86] Abedalmotaleb Zadin and Thomas Fevens. Maintaining path stability with node failure in mobile ad hoc networks. *Procedia Computer Science*, 19(0):1068 – 1073, 2013. The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013), the 3rd International Conference on Sustainable Energy Information Technology (SEIT-2013).

Appendix A

Pseudo Code

A.1 Grouping

Multiple unconnected nodes that have overlapping coverage areas, $(di_{ij} < 2 \times r)$ for $i, j \in N$, can be considered as a group. When identified, an additional centroid placed node can be determined to connect these nodes. The following details the logic to determine a group. The resultant matrix of nodes, Λ, is indexed by the group number g and node number in that group i.

Heuristic 10: FindGroups

Input: List of nodes N, Connection radius R
Output: List of groups $\Lambda = \{n_{gi}\}$

define $\Lambda = \{\}$
for $g = 0$ **to** $|N|$ **do**
    add N_g to Λ_g
    for $i = 0$ **to** $|N|$ **do**
        for $j = 0$ **to** $|\Lambda_g|$ **do**
            if $0 < d_{N_g,\Lambda_{gj}} < 2 \times R$ &
            $N_g \neq \Lambda_{gj}$
            & N_g is not connected to Λ_{gj} **then**
                add N_i to Λ_g

return Λ

A.2 Find Centroids

The method FindCentroids identifies clusters nodes that would benefit by a centroid placed point. By looking at all combinations of size 3 to 5 nodes, three tests are sequentially performed to determine the location of a centroid node: average coordinate, midpoint, and circumcenter.

Heuristic 11: FindCentroids

Data: List of nodes N, Connection radius R
Input: Matrix of groups of nodes Λ = FindGroups(N, R)
Output: Connecting centroid location $n = (n_x, n_y)$

```
define n = (0, 0)
for h = 0 to |Λ| do
    for i = 0 to |Λ_h| do
        for j = i to |Λ_h| do
            for k = j to |Λ_h| do
                for l = k to |Λ_h| do
                    for m = l to |Λ_h| do
                        n ← average(Λ_hi, Λ_hj, Λ_hk, Λ_hl, Λ_hm)
                        If n connects Λ_h then return n
                        n ← midpoint(Λ_hi, Λ_hj, Λ_hk, Λ_hl, Λ_hm)
                        If n connects Λ_h then return n
                    n ← average(Λ_hi, Λ_hj, Λ_hk, Λ_hl)
                    If n connects Λ_h then return n
                    n ← midpoint(Λ_hi, Λ_hj, Λ_hk, Λ_hl)
                    If n connects Λ_h then return n
                n ← average(Λ_hi, Λ_hj, Λ_hk)
                If n connects Λ_h then return n
                n ← midpoint(Λ_hi, Λ_hj, Λ_hk)
                If n connects Λ_h then return n
                n ← circumcenter(Λ_hi, Λ_hj, Λ_hk)
                If n connects Λ_h then return n
```

Appendix B

Checkpoint Definition

This section discusses the influence of the magnitude of user node movement rate, v, on tour definition.

B.1 Small Variation Between Checkpoints (Linear)

A smaller magnitude of v results in a smaller degree in angular change between checkpoints. This will result in a more linear set of checkpoints. At $v = 0$, the checkpoints will be created in a ray, starting at the control node, with a distance of S between them.

B.2 Large Variation Between Checkpoints (Random)

Increasing the magnitude of v increases the randomness of checkpoint definition. Each checkpoint will still have a distance of S between them, but can result in any polygonal arrangement. In such a case, there is a higher likelihood of user node's paths crossing. Instead of a network expanding with each checkpoint, it could be folded in on itself.

Appendix C

Experimental Results

Table C.1: Static Problem Experimentation Results

Graph	$\lvert P\rvert_{Math}$	$\lvert P\rvert_{MST}$	$\lvert P\rvert_{MMST}$	$\mu_{\lvert P\rvert_{Reduction}}$	$\sigma_{\lvert P\rvert_{Reduction}}$	$\frac{\lvert P\rvert_{MST}}{\lvert P\rvert_{Math}}$	$\frac{\lvert P\rvert_{MMST}}{\lvert P\rvert_{Math}}$	$\frac{\mu_{\lvert P\rvert_{Reduction}}}{\lvert P\rvert_{Math}}$
5-000	2	2	2	2.0	0.0	1.00	1.00	1.00
5-008	3	3	3	3.0	0.0	1.00	1.00	1.00
5-012	4	4	4	4.0	0.0	1.00	1.00	1.00
5-028	3	3	3	3.0	0.0	1.00	1.00	1.00
5-031	3	3	3	3.0	0.0	1.00	1.00	1.00
5-033	2	2	2	2.0	0.0	1.00	1.00	1.00
5-051	3	3	3	3.0	0.0	1.00	1.00	1.00
5-071	4	4	4	4.0	0.0	1.00	1.00	1.00
5-077	4	4	4	4.0	0.0	1.00	1.00	1.00
5-081	2	2	2	2.0	0.0	1.00	1.00	1.00
5-092	3	3	3	3.0	0.0	1.00	1.00	1.00
5-102	3	3	3	3.0	0.0	1.00	1.00	1.00
5-104	3	3	3	3.0	0.0	1.00	1.00	1.00
5-107	2	2	2	2.0	0.0	1.00	1.00	1.00
5-122	2	2	2	2.0	0.0	1.00	1.00	1.00
5-124	3	3	3	3.0	0.0	1.00	1.00	1.00
5-126	4	4	4	4.0	0.0	1.00	1.00	1.00
5-130	3	3	3	3.0	0.0	1.00	1.00	1.00
5-150	3	3	3	3.0	0.0	1.00	1.00	1.00
5-153	3	3	3	3.0	0.0	1.00	1.00	1.00
5-154	3	3	3	3.0	0.0	1.00	1.00	1.00
5-181	2	2	2	2.0	0.0	1.00	1.00	1.00
5-191	3	3	3	3.0	0.0	1.00	1.00	1.00
5-193	2	2	2	2.0	0.0	1.00	1.00	1.00
5-199	2	2	2	2.0	0.0	1.00	1.00	1.00
5R-00	4	4	4	4.0	0.0	1.00	1.00	1.00
5R-01	4	4	4	4.0	0.0	1.00	1.00	1.00
5R-02	3	3	3	3.0	0.0	1.00	1.00	1.00
5R-03	3	3	3	3.0	0.0	1.00	1.00	1.00

Continued on next page

Table C.1 – continued from previous page

Graph	$\lvert P\rvert_{Math}$	$\lvert P\rvert_{MST}$	$\lvert P\rvert_{MMST}$	$\mu_{\lvert P\rvert_{Reduction}}$	$\sigma_{\lvert P\rvert_{Reduction}}$	$\frac{\lvert P\rvert_{MST}}{\lvert P\rvert_{Math}}$	$\frac{\lvert P\rvert_{MMST}}{\lvert P\rvert_{Math}}$	$\frac{\mu_{\lvert P\rvert_{Reduction}}}{\lvert P\rvert_{Math}}$
5R-04	4	4	4	4.0	0.0	1.00	1.00	1.00
5R-05	4	6	6	4.0	0.0	1.50	1.50	1.00
5R-06	4	4	4	4.0	0.0	1.00	1.00	1.00
5R-08	3	3	3	3.0	0.0	1.00	1.00	1.00
10-02	dnf	6	7	6.0	0.0	-	-	-
10-06	5	7	7	5.0	0.0	1.40	1.40	1.00
10-07	6	7	7	6.0	0.0	1.17	1.17	1.00
10-09	dnf	7	6	5.0	0.0	-	-	-
10-10	4	5	6	4.0	0.0	1.25	1.50	1.00
10-11	5	6	5	5.0	0.0	1.20	1.00	1.00
10-12	dnf	7	7	6.0	0.0	-	-	-
10-13	dnf	7	6	6.0	0.0	-	-	-
10-20	4	6	5	4.0	0.0	1.50	1.25	1.00
10-21	4	6	4	4.0	0.0	1.50	1.00	1.00
10-22	5	6	5	5.0	0.0	1.20	1.00	1.00
10-23	dnf	4	4	4.0	0.0	-	-	-
10-24	3	5	3	3.0	0.0	1.67	1.00	1.00
10-25	dnf	5	5	5.0	0.0	-	-	-
10-26	5	6	6	6.0	0.0	1.20	1.20	1.20
10-27	5	7	7	5.0	0.0	1.40	1.40	1.00
10-28	dnf	7	6	6.0	0.0	-	-	-
10-29	5	6	5	5.0	0.0	1.20	1.00	1.00
10-30	5	8	5	5.0	0.0	1.60	1.00	1.00
10-31	5	6	5	5.0	0.0	1.20	1.00	1.00
10-32	6	8	6	6.0	0.0	1.33	1.00	1.00
10-33	4	5	4	4.0	0.0	1.25	1.00	1.00
10-34	6	8	7	6.3	0.5	1.33	1.17	1.04
10-36	4	6	6	4.0	0.0	1.50	1.50	1.00
10-37	4	4	4	4.0	0.0	1.00	1.00	1.00
10-38	5	5	5	5.0	0.0	1.00	1.00	1.00
10-39	4	5	4	4.0	0.0	1.25	1.00	1.00

Table C.2: Static Problem Validation and Experimentation Computation Time

Map	$\|U\|$	**Math** (HPC hh:mm:ss)	**Reduction** (PC seconds)
5-000	5	0:00:02	2.611
5-008	5	0:00:19	40.254
5-012	5	0:00:18	20.099
5-028	5	0:00:04	2.718
5-031	5	0:00:15	20.827
5-033	5	0:00:04	2.625
5-051	5	0:01:57	2.473
5-071	5	0:04:52	3.04
5-077	5	0:05:55	2.23
5-081	5	0:00:03	1.627
5-092	5	0:00:23	2.692
5-102	5	0:00:25	4.002
5-104	5	0:00:25	2.598
5-107	5	0:00:06	2.161
5-122	5	0:00:06	2.231
5-124	5	0:01:18	0.959
5-126	5	0:01:22	1.391
5-130	5	0:00:06	1.032
5-150	5	0:00:10	0.969
5-153	5	0:00:06	1.031
5-154	5	0:01:15	0.837
5-181	5	0:00:02	0.734
5-191	5	0:01:20	0.771
5-193	5	0:00:00	0.739
5-199	5	0:00:02	0.724
5R-00	5	0:00:35	1.962
5R-01	5	0:02:35	1.788
5R-02	5	0:00:06	1.366
5R-03	5	0:00:04	1.843
5R-04	5	0:00:25	1.996
5R-05	5	0:00:08	2.34
5R-06	5	0:00:10	2.199
5R-08	5	0:00:08	0.941
10-02	10	>30:00:00	6.83
10-06	10	1:01:46	9.833
10-07	10	1:32:32	8.124

Continued on next page

Table C.2 – continued from previous page

Map	$\|U\|$	Math (HPC hh:mm:ss)	Reduction (PC seconds)
10-09	10	>30:00:00	8.324
10-10	10	0:26:41	4.217
10-11	10	16:32:00	5.12
10-12	10	>30:00:00	6.823
10-13	10	>30:00:00	6.315
10-20	10	4:01:47	4.676
10-21	10	0:32:20	4.414
10-22	10	7:20:59	6.009
10-23	10	>30:00:00	3.782
10-24	10	0:02:56	3.434
10-25	10	>30:00:00	5.178
10-26	10	14:44:19	6.393
10-27	10	2:12:24	6.009
10-28	10	>30:00:00	6.929
10-29	10	0:10:58	5.097
10-30	10	5:09:46	5.825
10-31	10	15:31:42	5.727
10-32	10	1:56:43	6.164
10-33	10	3:02:58	4.629
10-34	10	1:11:35	8.018
10-36	10	3:26:00	6.346
10-37	10	17:07:42	3.608
10-38	10	9:40:27	4.862
10-39	10	10:08:17	4.414
Mean		6:47:05	4.882
Std.Dev		13:26:52	6.002

”>30:00:00” treated as 30 hours when calculating mean and std.dev.

Table C.3: Dynamic Deterministic Validation Results

Graph	$\mu_{\|A_t\|}$	$\sigma_{\|A_t\|}$	μ_{ANCU}	σ_{ANCU}	μ_{OFV}	σ_{OFV}	$\|A\|_{Math}$	$ANCU_{Math}$	OFV_{Math}
v_01	2.0	0.0	3.0	0.0	-25.0	0.0	2	3	-25
v_02	5.0	0.0	3.0	0.0	-22.0	0.0	5	3	-22
v_05	5.0	0.0	3.0	0.0	-22.0	0.0	5	3	-22
v_06	3.0	0.0	3.0	0.0	-24.0	0.0	2	3	-25
v_07	5.0	0.0	3.0	0.0	-22.0	0.0	5	3	-22
v_08	3.0	0.0	5.0	0.0	-42.0	0.0	4	5	-41
v_09	5.0	0.0	5.0	0.0	-40.0	0.0	4	5	-41
v_10	5.0	0.0	5.0	0.0	-40.0	0.0	4	5	-41
v_11	3.0	0.0	5.0	0.0	-42.0	0.0	3	5	-42
v_12	5.0	0.0	5.0	0.0	-40.0	0.0	5	5	-40
v_15	7.0	1.0	5.0	0.0	-38.0	1.0	4	4	-32
v_16	10.0	1.0	7.0	0.0	-53.0	1.0	8	8	-64
v_19	7.0	0.0	7.0	0.0	-56.0	0.0	6	7	-57
v_20	6.0	0.0	5.0	0.0	-39.0	0.0	6	5	-39
v_21	8.3	0.6	5.0	0.0	-36.7	0.6	9	5	-36

Table C.4: Dynamic Deterministic Plan and Reactive Experimentation Results

	MANET						Reactive					
	ANCU		*B*		*OFV*		*ANCU*		*B*		*OFV*	
Graph-Replication	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
S_SAR_1002	4.00	0.00	0.90	0.01	-567.71	0.23	3.98	0.01	0.95	0.00	-567.91	1.83
S_SAR_1003	4.00	0.00	0.88	0.01	-566.85	0.66	3.92	0.06	0.94	0.01	-559.16	7.14
S_SAR_1004	4.00	0.00	0.89	0.01	-567.08	0.31	3.92	0.07	0.93	0.01	-558.91	9.75
S_SAR_1005	4.00	0.00	0.88	0.04	-566.70	1.63	3.96	0.03	0.95	0.02	-564.60	4.74
S_SAR_1006	4.00	0.00	0.86	0.02	-565.70	0.87	4.00	0.00	0.94	0.00	-569.18	0.15
S_SAR_1007	4.00	0.00	0.87	0.00	-566.43	0.10	3.87	0.03	0.94	0.01	-552.25	5.02
S_SAR_1008	4.00	0.00	0.88	0.02	-566.69	0.92	3.98	0.03	0.94	0.01	-567.21	3.53
S_SAR_1009	4.00	0.00	0.89	0.01	-567.25	0.56	3.96	0.03	0.95	0.00	-564.63	4.44
S_SAR_1010	4.00	0.00	0.87	0.06	-566.44	2.71	3.95	0.02	0.95	0.00	-563.96	2.89
S_TRA_1001	10.00	0.00	0.95	0.00	-1888.17	0.17	8.16	0.27	0.99	0.00	-1578.37	50.40
S_TRA_1002	9.79	0.37	0.95	0.00	-1909.74	69.45	6.04	0.65	0.99	0.00	-1222.18	124.13
S_TRA_1003	10.00	0.00	0.95	0.00	-1825.83	0.10	6.42	0.25	0.99	0.00	-1214.27	45.44
S_TRA_1004	9.89	0.19	0.95	0.01	-1867.91	34.13	5.76	2.20	0.99	0.00	-1132.14	409.09
S_TRA_1005	10.00	0.00	0.94	0.01	-1887.13	0.52	5.62	1.67	0.99	0.00	-1107.26	310.66
S_TRA_1006	10.00	0.00	0.95	0.00	-1950.14	0.21	5.95	0.40	0.99	0.00	-1206.27	77.63
S_TRA_1007	10.00	0.00	0.96	0.00	-1919.36	0.07	7.62	0.26	0.99	0.00	-1502.20	49.60
S_TRA_1008	10.00	0.00	0.94	0.01	-2011.40	0.47	7.28	0.14	0.99	0.00	-1506.16	27.03
S_TRA_1009	10.00	0.00	0.94	0.01	-1918.23	0.53	6.45	0.41	0.99	0.00	-1281.29	77.13
S_TRA_1010	10.00	0.00	0.95	0.01	-1950.02	0.32	6.69	0.52	0.99	0.00	-1347.24	99.89
M_PAT_1001	9.96	0.04	0.95	0.00	-1727.04	5.89	7.99	0.23	0.99	0.00	-1423.35	38.70
M_PAT_1002	9.90	0.18	0.94	0.01	-1470.11	25.98	7.56	0.11	0.99	0.00	-1159.40	16.52
M_PAT_1003	9.98	0.04	0.94	0.00	-1728.92	7.01	7.71	0.04	0.99	0.00	-1375.36	7.55
M_PAT_1004	9.82	0.07	0.94	0.00	-1551.11	10.65	7.09	0.37	0.99	0.00	-1157.34	58.06
M_PAT_1005	9.99	0.02	0.93	0.00	-1575.66	3.32	8.56	0.08	0.99	0.00	-1387.33	12.49
M_PAT_1006	9.90	0.09	0.95	0.00	-1655.18	15.15	7.04	0.34	0.99	0.00	-1193.42	37.51
M_PAT_1007	9.78	0.12	0.95	0.00	-1665.98	19.69	7.09	0.29	0.99	0.00	-1246.36	49.11
M_PAT_1008	9.90	0.13	0.94	0.00	-1715.81	22.03	7.20	0.11	0.99	0.00	-1287.37	18.34
M_PAT_1009	9.98	0.04	0.94	0.01	-1697.71	6.93	7.53	0.10	0.99	0.00	-1320.39	16.56
M_PAT_1010	9.83	0.09	0.94	0.00	-1734.59	15.13	7.94	0.17	0.99	0.00	-1438.96	15.71
M_POL_1001	10.00	0.00	0.92	0.01	-773.10	0.15	9.19	0.14	0.97	0.00	-742.17	10.88
M_POL_1002	10.00	0.00	0.92	0.01	-772.92	0.15	9.40	0.04	0.97	0.00	-758.21	3.45
M_POL_1003	10.00	0.00	0.92	0.01	-772.90	0.20	8.88	0.28	0.97	0.00	-718.24	21.69
M_POL_1004	10.00	0.00	0.88	0.01	-771.93	0.14	8.64	0.21	0.97	0.00	-699.10	16.58
M_POL_1005	9.97	0.02	0.91	0.01	-770.92	1.96	8.69	0.14	0.97	0.00	-703.12	10.91
M_POL_1006	10.00	0.00	0.90	0.01	-772.40	0.21	8.95	0.10	0.97	0.00	-723.19	7.57
M_POL_1007	10.00	0.00	0.91	0.00	-772.70	0.04	8.71	0.02	0.97	0.00	-704.15	1.72
M_POL_1008	10.00	0.00	0.93	0.01	-773.15	0.16	8.59	0.21	0.97	0.00	-695.15	16.52
M_POL_1009	10.00	0.00	0.91	0.02	-772.64	0.48	9.13	0.08	0.96	0.00	-737.06	6.25
M_POL_1010	10.00	0.00	0.92	0.01	-772.91	0.28	9.18	0.12	0.97	0.00	-741.16	9.67
M_RAN_1001	9.99	0.02	0.92	0.01	-772.14	1.38	8.91	0.10	1.00	0.00	-720.99	7.55
M_RAN_1002	9.72	0.39	0.92	0.01	-751.72	29.65	8.47	0.24	1.00	0.00	-686.99	19.05
M_RAN_1003	10.00	0.00	0.92	0.01	-772.95	0.30	9.03	0.22	1.00	0.00	-730.00	17.06
M_RAN_1004	10.00	0.00	0.93	0.01	-773.31	0.16	8.55	0.72	1.00	0.00	-692.99	56.47
M_RAN_1005	9.99	0.02	0.93	0.01	-772.20	1.63	9.45	0.09	1.00	0.00	-762.97	6.93
M_RAN_1006	9.95	0.09	0.92	0.02	-769.19	6.57	8.86	0.24	1.00	0.00	-716.99	18.33
M_RAN_1007	9.47	0.04	0.93	0.01	-733.85	3.19	9.28	0.08	1.00	0.00	-750.00	6.25
M_RAN_1008	9.96	0.04	0.93	0.01	-770.43	3.09	9.28	0.08	1.00	0.00	-749.99	6.25
M_RAN_1009	10.00	0.00	0.94	0.01	-773.44	0.13	8.49	0.06	1.00	0.00	-687.99	4.58
M_RAN_1010	9.79	0.36	0.93	0.01	-757.93	26.53	9.01	0.61	1.00	0.00	-729.00	47.84
M_SAR_1001	20.00	0.00	0.91	0.00	-2619.27	0.15	19.92	0.05	1.00	0.00	-2309.02	633.88

Continued on next page

Table C.4 – continued from previous page

	MANET						Reactive					
	ANCU		*B*		*OFV*		*ANCU*		*B*		*OFV*	
Graph-Replication	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
M_SAR_1002	20.00	0.00	0.92	0.01	-2619.46	0.46	19.89	0.08	1.00	0.00	-2669.95	10.54
M_SAR_1003	20.00	0.00	0.93	0.00	-2619.90	0.18	19.75	0.06	1.00	0.00	-2650.97	7.94
M_SAR_1004	20.00	0.00	0.90	0.01	-2618.73	0.35	19.76	0.28	1.00	0.00	-2651.97	37.27
M_SAR_1005	20.00	0.00	0.90	0.01	-2618.78	0.53	19.58	0.15	1.00	0.00	-2627.95	20.42
M_SAR_1006	20.00	0.00	0.90	0.00	-2618.65	0.16	19.43	0.30	1.00	0.00	-2608.97	39.01
M_SAR_1007	20.00	0.00	0.92	0.01	-2619.35	0.26	19.85	0.09	1.00	0.00	-2663.96	12.49
M_SAR_1008	20.00	0.00	0.92	0.00	-2619.61	0.15	19.70	0.11	1.00	0.00	-2644.96	15.01
M_SAR_1009	20.00	0.00	0.92	0.01	-2619.65	0.25	19.60	0.17	1.00	0.00	-2630.95	22.12
M_SAR_1010	20.00	0.00	0.90	0.01	-2618.81	0.25	19.67	0.09	1.00	0.00	-2639.96	12.13
M_TRA_1001	20.00	0.00	0.94	0.00	-1096.92	0.06	18.95	1.05	0.96	0.00	-1098.33	60.01
M_TRA_1002	20.00	0.00	0.94	0.00	-1036.06	0.01	19.26	1.28	0.96	0.00	-1057.36	69.28
M_TRA_1003	20.00	0.00	0.93	0.00	-1096.79	0.06	18.95	0.00	0.96	0.00	-1098.25	0.01
M_TRA_1004	20.00	0.00	0.95	0.01	-1036.12	0.09	18.15	1.70	0.96	0.00	-997.36	91.65
M_TRA_1005	20.00	0.00	0.95	0.00	-1097.07	0.04	16.14	0.61	0.97	0.00	-938.38	34.65
M_TRA_1006	20.00	0.00	0.95	0.01	-1097.18	0.10	17.89	0.00	0.97	0.00	-1038.44	0.01
M_TRA_1007	20.00	0.00	0.95	0.01	-1097.11	0.10	15.79	3.16	0.97	0.00	-918.41	180.00
M_TRA_1008	20.00	0.00	0.95	0.00	-1157.98	0.06	16.67	2.52	0.97	0.00	-1019.35	151.01
M_TRA_1009	20.00	0.00	0.95	0.00	-1158.02	0.06	17.95	2.65	0.97	0.00	-1096.32	158.73
M_TRA_1010	20.00	0.00	0.94	0.00	-1096.94	0.06	15.79	0.00	0.96	0.00	-918.29	0.01
L_SAR_1001	41.00	0.00	0.91	0.03	-5452.01	1.24	40.39	0.24	1.00	0.00	-5375.00	31.61
L_SAR_1002	41.00	0.00	0.92	0.00	-5452.59	0.22	40.33	0.21	1.00	0.00	-5367.00	27.87
L_SAR_1003	41.00	0.00	0.92	0.01	-5452.48	0.32	40.85	0.13	1.00	0.00	-5436.00	17.58
L_SAR_1004	41.00	0.00	0.91	0.01	-5451.87	0.42	40.82	0.04	1.00	0.00	-5432.00	5.20
L_SAR_1005	41.00	0.00	0.91	0.01	-5452.11	0.58	40.42	0.27	1.00	0.00	-5379.99	35.79
L_SAR_1006	41.00	0.00	0.91	0.02	-5452.19	0.81	40.55	0.10	1.00	0.00	-5397.00	13.86
L_SAR_1007	41.00	0.00	0.93	0.01	-5452.74	0.30	40.81	0.21	1.00	0.00	-5431.00	27.87
L_SAR_1008	41.00	0.00	0.91	0.01	-5451.94	0.60	40.42	0.34	1.00	0.00	-5380.00	45.13
L_SAR_1009	41.00	0.00	0.91	0.00	-5451.95	0.13	40.48	0.25	1.00	0.00	-5388.00	32.36
L_SAR_1010	41.00	0.00	0.92	0.00	-5452.43	0.14	40.77	0.18	1.00	0.00	-5426.00	24.00
X_POL_1001	60.00	0.00	0.90	0.00	-4703.52	0.01	58.44	0.16	0.99	0.00	-4583.64	12.48
X_POL_1002	59.99	0.02	0.92	0.00	-4703.01	1.71	58.17	0.14	0.99	0.00	-4562.64	10.54
X_POL_1003	60.00	0.00	0.91	0.00	-4703.78	0.11	58.73	0.17	0.98	0.00	-4606.59	13.08
X_POL_1004	60.00	0.00	0.92	0.01	-4703.81	0.16	57.19	0.24	0.98	0.00	-4486.56	18.73
X_POL_1005	60.00	0.00	0.91	0.00	-4703.73	0.12	58.17	0.36	0.98	0.00	-4562.58	28.36
X_POL_1006	60.00	0.00	0.92	0.00	-4703.92	0.05	58.74	0.31	0.98	0.00	-4607.61	24.42
X_POL_1007	60.00	0.00	0.92	0.01	-4703.97	0.14	57.95	0.06	0.98	0.00	-4545.59	4.55
X_POL_1008	60.00	0.00	0.92	0.01	-4703.90	0.14	57.71	0.23	0.98	0.00	-4526.59	17.57
X_POL_1009	60.00	0.00	0.91	0.00	-4703.78	0.05	58.37	0.08	0.98	0.00	-4578.58	6.25
X_POL_1010	60.00	0.00	0.92	0.00	-4704.03	0.11	58.23	0.10	0.98	0.00	-4567.55	7.94

Table C.5: Dynamic Deterministic Experimentation Computation Time per Instance

Instance	**RTI** (hh:mm:ss)	**GA** (hh:mm:ss)	**Reactive** (hh:mm:ss)
S_SAR_1001	00:00:34	00:44:41	00:00:41
S_SAR_1002	00:00:31	00:44:29	00:00:22
S_SAR_1003	00:00:26	00:40:31	00:00:00
S_SAR_1004	00:00:53	00:42:10	00:00:26
S_SAR_1005	00:00:55	00:51:31	00:00:05
S_SAR_1006	00:00:07	00:34:02	00:00:19
S_SAR_1007	00:00:22	00:36:19	00:00:41
S_SAR_1008	00:00:41	00:37:53	00:00:58
S_SAR_1009	00:00:19	00:45:19	00:00:48
S_SAR_1010	00:00:02	00:35:10	00:00:46
S_TRA_1001	00:07:50	06:53:24	00:01:14
S_TRA_1002	00:09:14	07:39:24	00:01:29
S_TRA_1003	00:07:10	06:14:36	00:01:00
S_TRA_1004	00:06:46	04:28:24	00:01:34
S_TRA_1005	00:06:14	05:15:00	00:01:22
S_TRA_1006	00:08:19	06:09:24	00:01:26
S_TRA_1007	00:07:02	05:48:12	00:01:00
S_TRA_1008	00:09:41	06:58:48	00:01:43
S_TRA_1009	00:06:46	04:54:48	00:00:48
S_TRA_1010	00:07:34	06:40:00	00:01:48
M_SAR_1001	00:18:07	08:20:48	00:02:38
M_SAR_1002	00:10:55	06:03:00	00:01:19
M_SAR_1003	00:18:12	08:51:36	00:03:12
M_SAR_1004	00:18:17	08:05:24	00:01:43
M_SAR_1005	00:13:58	06:34:12	00:02:41
M_SAR_1006	00:17:55	07:44:12	00:02:24
M_SAR_1007	00:12:12	06:17:24	00:01:46
M_SAR_1008	00:12:07	06:45:00	00:02:53
M_SAR_1009	00:14:10	06:15:48	00:02:48
M_SAR_1010	00:10:00	05:28:24	00:01:22
M_TRA_1001	00:01:19	00:00:17	00:00:43
M_TRA_1002	00:01:55	00:00:02	00:00:24
M_TRA_1003	00:01:55	00:01:02	00:00:43
M_TRA_1004	00:01:38	00:28:38	00:00:29
M_TRA_1005	00:02:55	00:25:02	00:00:24
M_TRA_1006	00:01:43	00:00:48	00:00:14

Continued on next page

Table C.5 – continued from previous page

M_TRA_1007	00:02:14	00:37:24	00:00:17
M_TRA_1008	00:02:26	01:01:29	00:00:31
M_TRA_1009	00:02:53	01:02:38	00:00:38
M_TRA_1010	00:01:02	00:33:24	00:00:19
M_POL_1001	00:01:31	02:08:38	00:00:48
M_POL_1002	00:01:36	01:47:38	00:00:07
M_POL_1003	00:02:53	02:12:02	00:00:17
M_POL_1004	00:01:26	01:18:24	00:00:17
M_POL_1005	00:01:38	01:48:12	00:00:24
M_POL_1006	00:01:10	01:16:29	00:00:50
M_POL_1007	00:02:41	01:56:10	00:00:26
M_POL_1008	00:02:22	01:37:43	00:00:12
M_POL_1009	00:00:41	01:11:24	00:00:17
M_POL_1010	00:01:31	01:54:00	00:00:34
M_RAN_1001	00:30:05	10:46:24	00:02:41
M_RAN_1002	00:52:14	10:56:00	00:03:38
M_RAN_1003	00:58:41	15:06:12	00:04:12
M_RAN_1004	00:33:50	08:05:24	00:03:10
M_RAN_1005	00:07:26	03:39:12	00:00:17
M_RAN_1006	00:19:55	05:21:00	00:01:17
M_RAN_1007	00:46:58	13:13:48	00:04:10
M_RAN_1008	00:30:12	08:57:36	00:02:14
M_RAN_1009	00:29:19	06:59:48	00:02:17
M_RAN_1010	00:36:50	08:14:24	00:03:41
M_PAT_1001	00:48:36	05:22:12	00:02:22
M_PAT_1002	01:05:41	14:35:24	00:04:10
M_PAT_1003	01:04:02	08:12:12	00:04:12
M_PAT_1004	00:52:34	03:33:48	00:04:43
M_PAT_1005	00:15:05	13:51:12	00:01:02
M_PAT_1006	00:36:19	20:49:48	00:03:48
M_PAT_1007	00:42:02	23:38:00	00:04:58
M_PAT_1008	00:53:46	02:25:12	00:04:24
M_PAT_1009	00:37:14	19:56:24	00:03:41
M_PAT_1010	00:44:12	23:14:12	00:03:17
L_SAR_1001	04:43:48	15:10:00	00:31:43
L_SAR_1002	04:29:12	13:04:00	00:30:24
L_SAR_1003	03:38:24	12:49:00	00:20:24
L_SAR_1004	03:30:24	09:18:00	00:19:48
L_SAR_1005	03:20:24	13:34:00	00:17:50

Continued on next page

Table C.5 – continued from previous page

L_SAR_1006	03:51:36	15:50:00	00:27:29
L_SAR_1007	03:48:24	13:03:00	00:22:58
L_SAR_1008	03:55:24	10:09:00	00:22:29
L_SAR_1009	03:23:12	10:45:00	00:19:50
L_SAR_1010	04:00:12	13:32:00	00:24:14
X_POL_1001	00:50:10	10:35:00	00:04:29
X_POL_1002	00:55:41	09:53:24	00:07:31
X_POL_1003	00:43:53	10:05:24	00:05:38
X_POL_1004	00:56:07	10:43:24	00:06:50
X_POL_1005	00:47:26	09:37:12	00:04:00
X_POL_1006	00:40:22	08:43:48	00:03:46
X_POL_1007	00:55:38	09:58:00	00:04:26
X_POL_1008	00:51:05	09:40:12	00:05:36
X_POL_1009	00:48:34	10:24:00	00:05:26
X_POL_1010	00:59:38	09:57:12	00:06:46
Mean	00:43:29	10:49:50	00:04:56
Std.Dev.	01:10:40	12:00:55	00:07:23

Table C.6: Dynamic Stochastic MANET and Reactive Experimentation Results

	MANET						Reactive		
	ANCU		*B*		*OFV*				
Graph-Replication	μ	σ	μ	σ	μ	σ	*ANCU*	*B*	*OFV*
S_SAR_01-1002-1403296479	3.86	0.000	0.944	0.000	-551.54	0.000000	3.84	0.93	-548.03
S_SAR_01-1002-241471907	4.00	0.000	0.944	0.000	-569.53	0.000000	3.95	0.93	-562.99
S_SAR_01-1002-253537267	3.82	0.000	0.945	0.000	-545.59	0.000000	3.86	0.93	-551.12
S_SAR_01-1002-580778438	3.95	0.000	0.939	0.000	-563.34	0.000000	3.89	0.93	-553.87
S_SAR_01-1002-972075082	3.91	0.000	0.944	0.000	-557.56	0.000000	4.00	0.93	-568.97
S_SAR_02-1002-106190266	4.00	0.000	0.944	0.000	-569.52	0.000000	3.98	0.95	-566.79
S_SAR_02-1002-1103914701	3.98	0.000	0.943	0.000	-566.50	0.000000	3.84	0.95	-548.62
S_SAR_02-1002-317592740	3.86	0.000	0.942	0.000	-551.43	0.000000	3.86	0.94	-551.48
S_SAR_02-1002-696824024	4.00	0.000	0.941	0.000	-569.42	0.000000	3.89	0.94	-554.56
S_SAR_02-1002-724683417	4.00	0.000	0.941	0.000	-569.39	0.000000	3.98	0.94	-566.54
S_SAR_03-1003-1367307288	4.00	0.000	0.933	0.000	-569.07	0.000000	3.98	0.93	-566.07
S_SAR_03-1003-1774397965	4.00	0.000	0.934	0.000	-569.07	0.000000	4.00	0.93	-569.08
S_SAR_03-1003-217548076	4.00	0.000	0.934	0.000	-569.10	0.000000	4.00	0.93	-568.99
S_SAR_03-1003-6145602	3.98	0.000	0.930	0.000	-565.92	0.000000	3.93	0.93	-560.12
S_SAR_03-1003-976010644	4.00	0.000	0.928	0.000	-568.82	0.000000	3.98	0.93	-565.88
S_SAR_04-1001-1028000757	3.93	0.000	0.944	0.000	-560.56	0.000000	3.89	0.92	-553.53
S_SAR_04-1001-1618634515	3.86	0.000	0.943	0.000	-551.48	0.000000	3.80	0.92	-541.49
S_SAR_04-1001-1646493908	3.61	0.000	0.941	0.000	-518.39	0.000000	3.57	0.92	-511.60
S_SAR_04-1001-2025725192	3.89	0.000	0.943	0.000	-554.51	0.000000	3.77	0.92	-538.39
S_SAR_04-1001-468875303	3.98	0.000	0.946	0.000	-566.61	0.000000	3.95	0.92	-562.61
S_SAR_05-1001-1139358567	3.98	0.000	0.967	0.000	-567.53	0.000000	3.89	0.95	-554.69
S_SAR_05-1001-1670624628	3.95	0.000	0.967	0.000	-564.57	0.000000	3.84	0.94	-548.55
S_SAR_05-1001-1698484021	4.00	0.000	0.965	0.000	-570.45	0.000000	3.95	0.94	-563.43
S_SAR_05-1001-520865416	3.98	0.000	0.964	0.000	-567.42	0.000000	3.95	0.94	-563.33
S_SAR_05-1001-912162060	3.95	0.000	0.966	0.000	-564.49	0.000000	3.80	0.94	-542.47
S_SAR_06-1001-1031936319	4.00	0.000	0.931	0.000	-568.98	0.000000	3.98	0.93	-565.73
S_SAR_06-1001-1343383457	4.00	0.000	0.935	0.000	-569.13	0.000000	4.00	0.93	-568.82
S_SAR_06-1001-1435298323	3.98	0.010	0.936	0.000	-567.19	1.720000	4.00	0.93	-568.93
S_SAR_06-1001-1463157716	4.00	0.000	0.933	0.000	-569.07	0.000000	3.98	0.93	-565.77
S_SAR_06-1001-46277244	3.98	0.000	0.933	0.000	-566.03	0.000000	3.98	0.92	-565.67
S_SAR_07-1003-1108057152	3.70	0.000	0.930	0.000	-529.92	0.000000	3.68	0.92	-526.67
S_SAR_07-1003-1135916545	4.00	0.000	0.935	0.000	-569.12	0.000000	3.86	0.93	-550.88
S_SAR_07-1003-1515147829	3.65	0.010	0.931	0.000	-522.94	1.740000	3.61	0.92	-517.66
S_SAR_07-1003-517423394	4.00	0.000	0.931	0.000	-568.96	0.000000	3.86	0.92	-550.60
S_SAR_07-1003-728825868	4.00	0.000	0.928	0.000	-568.83	0.000000	3.84	0.92	-547.62
S_SAR_08-1001-1227831411	3.98	0.000	0.934	0.000	-566.11	0.000000	3.89	0.92	-553.61
S_SAR_08-1001-1607062695	4.00	0.000	0.931	0.000	-568.98	0.000000	3.89	0.92	-553.41
S_SAR_08-1001-441509450	3.80	0.010	0.935	0.000	-543.13	1.730000	3.59	0.92	-514.55
S_SAR_08-1001-820740734	3.91	0.000	0.937	0.000	-557.21	0.000000	3.93	0.92	-559.66
S_SAR_08-1001-848600127	3.91	0.000	0.934	0.000	-557.09	0.000000	3.84	0.92	-547.52
S_SAR_09-1002-1563695920	4.00	0.000	0.947	0.000	-569.65	0.000000	3.95	0.94	-563.26
S_SAR_09-1002-1591555313	3.95	0.000	0.946	0.000	-563.63	0.000000	4.00	0.94	-569.41
S_SAR_09-1002-386077315	3.91	0.000	0.946	0.000	-557.63	0.000000	3.91	0.94	-557.14
S_SAR_09-1002-593830878	4.00	0.000	0.947	0.000	-569.67	0.000000	3.91	0.94	-557.36
S_SAR_09-1002-765308599	3.80	0.000	0.946	0.000	-542.63	0.000000	3.86	0.94	-551.30
S_SAR_10-1002-1264314142	4.00	0.000	0.960	0.000	-570.26	0.000000	3.95	0.94	-563.41
S_SAR_10-1002-1854947900	3.73	0.000	0.963	0.000	-534.39	0.000000	3.45	0.94	-497.53
S_SAR_10-1002-857223465	4.00	0.000	0.959	0.000	-570.20	0.000000	3.86	0.94	-551.39
S_SAR_10-1002-885082858	4.00	0.000	0.962	0.000	-570.32	0.000000	3.86	0.94	-551.40
S_SAR_10-1002-98760897	4.00	0.000	0.960	0.000	-570.26	0.000000	3.95	0.94	-563.41

Continued on next page

Table C.6 – continued from previous page

	MANET						Reactive		
	ANCU		*B*		*OFV*				
Graph-Replication	μ	σ	μ	σ	μ	σ	*ANCU*	*B*	*OFV*
S_TRA_01-1001-1079990669	9.26	0.000	0.991	0.000	-1783.42	0.000000	9.23	0.99	-1777.21
S_TRA_01-1001-1412661031	8.51	0.260	0.991	0.000	-1643.43	47.650000	8.65	0.99	-1669.14
S_TRA_01-1001-1591061572	9.48	0.000	0.991	0.000	-1825.44	0.000000	9.44	0.99	-1816.22
S_TRA_01-1001-1855947788	8.75	0.190	0.991	0.000	-1688.43	35.540000	8.56	0.99	-1654.31
S_TRA_01-1001-907019319	8.78	0.010	0.991	0.000	-1695.43	1.730000	8.95	0.99	-1726.23
S_TRA_02-1003-1004531489	9.28	0.080	0.991	0.000	-1845.44	15.590000	8.23	0.99	-1644.12
S_TRA_02-1003-1155943389	9.66	0.010	0.992	0.000	-1918.46	1.730000	8.19	0.99	-1635.27
S_TRA_02-1003-1371865526	9.49	0.010	0.992	0.000	-1885.48	1.730000	9.28	0.99	-1845.28
S_TRA_02-1003-575426598	9.40	0.010	0.991	0.000	-1867.45	1.730000	7.91	0.99	-1581.06
S_TRA_02-1003-627248598	9.38	0.000	0.992	0.000	-1863.46	0.000000	5.69	0.98	-1155.03
S_TRA_03-1003-1103439179	9.53	0.000	0.990	0.000	-1775.42	0.000000	7.70	0.98	-1445.00
S_TRA_03-1003-153521420	8.70	0.000	0.990	0.000	-1625.42	0.000000	7.68	0.98	-1442.07
S_TRA_03-1003-2046789732	9.38	0.000	0.990	0.000	-1748.38	0.000000	7.78	0.98	-1459.94
S_TRA_03-1003-407736755	9.30	0.000	0.991	0.000	-1733.45	0.000000	7.80	0.98	-1463.08
S_TRA_03-1003-661952090	9.32	0.000	0.990	0.000	-1736.42	0.000000	8.20	0.98	-1535.09
S_TRA_04-1002-1499640246	9.60	0.000	0.990	0.000	-1846.37	0.000000	8.85	0.99	-1708.12
S_TRA_04-1002-209170202	9.69	0.010	0.989	0.000	-1863.34	1.730000	8.89	0.99	-1714.17
S_TRA_04-1002-2091767633	9.44	0.060	0.990	0.000	-1816.38	10.820000	8.56	0.99	-1654.14
S_TRA_04-1002-449925698	9.65	0.000	0.990	0.000	-1855.36	0.000000	9.31	0.99	-1792.18
S_TRA_04-1002-536411373	9.42	0.020	0.990	0.000	-1813.37	3.000000	7.77	0.99	-1507.14
S_TRA_05-1003-1242922604	9.19	0.000	0.991	0.000	-1771.42	0.000000	7.47	0.99	-1450.32
S_TRA_05-1003-1251793817	9.20	0.040	0.991	0.000	-1772.43	6.930000	7.77	0.99	-1507.11
S_TRA_05-1003-1689651383	8.94	0.000	0.990	0.000	-1723.40	0.000000	7.53	0.99	-1462.23
S_TRA_05-1003-558880351	8.81	0.000	0.991	0.000	-1699.42	0.000000	8.34	0.99	-1612.29
S_TRA_05-1003-737280892	9.30	0.080	0.990	0.000	-1791.40	14.800000	8.79	0.99	-1696.32
S_TRA_06-1002-1020987175	9.02	0.010	0.992	0.000	-1795.49	1.730000	8.36	0.99	-1668.32
S_TRA_06-1002-224548247	9.04	0.020	0.992	0.000	-1798.49	3.460000	8.67	0.99	-1728.39
S_TRA_06-1002-440470384	9.11	0.020	0.992	0.000	-1813.49	3.460000	8.59	0.99	-1713.36
S_TRA_06-1002-466381384	8.83	0.000	0.992	0.000	-1758.49	0.000000	8.23	0.99	-1644.40
S_TRA_06-1002-843664275	8.79	0.010	0.992	0.000	-1750.48	1.730000	7.84	0.99	-1569.34
S_TRA_07-1003-1359254841	8.84	0.000	0.994	0.000	-1733.61	0.000000	7.02	0.99	-1388.29
S_TRA_07-1003-1504653449	8.93	0.030	0.993	0.000	-1749.58	6.240000	8.71	0.99	-1709.35
S_TRA_07-1003-1650052057	9.62	0.000	0.993	0.000	-1880.55	0.000000	7.59	0.99	-1496.30
S_TRA_07-1003-418494946	9.70	0.000	0.993	0.000	-1895.57	0.000000	8.90	0.99	-1745.34
S_TRA_07-1003-794700196	8.89	0.010	0.993	0.000	-1743.54	1.730000	8.44	0.99	-1658.31
S_TRA_08-1003-1095446266	9.56	0.000	0.993	0.000	-1958.51	0.000000	8.21	0.99	-1691.26
S_TRA_08-1003-1902002161	9.33	0.010	0.993	0.000	-1941.54	3.000000	8.55	0.99	-1785.31
S_TRA_08-1003-208170113	9.33	0.010	0.993	0.000	-1941.51	3.000000	7.91	0.99	-1656.27
S_TRA_08-1003-342126238	9.25	0.000	0.993	0.000	-1926.53	0.000000	8.21	0.99	-1716.33
S_TRA_08-1003-997518108	9.13	0.000	0.993	0.000	-1902.54	0.000000	8.03	0.99	-1680.37
S_TRA_09-1003-1307303504	9.23	0.010	0.989	0.000	-1806.31	1.730000	8.32	0.99	-1634.11
S_TRA_09-1003-1452702112	9.41	0.000	0.989	0.000	-1841.31	0.000000	8.71	0.99	-1709.09
S_TRA_09-1003-1556227574	9.55	0.010	0.989	0.000	-1867.28	1.730000	7.54	0.99	-1487.10
S_TRA_09-1003-1701626182	9.19	0.000	0.989	0.000	-1799.30	0.000000	7.76	0.99	-1529.08
S_TRA_09-1003-991672929	9.86	0.100	0.989	0.000	-1926.30	19.050000	8.05	0.99	-1583.07
S_TRA_10-1001-1123967312	9.30	0.010	0.994	0.000	-1849.61	1.730000	8.63	0.99	-1719.38
S_TRA_10-1001-1233506066	9.80	0.000	0.994	0.000	-1944.60	0.000000	9.14	0.99	-1818.35
S_TRA_10-1001-124294484	8.96	0.010	0.994	0.000	-1783.59	1.730000	8.80	0.99	-1752.42
S_TRA_10-1001-1285328066	9.56	0.010	0.994	0.000	-1898.59	1.730000	9.00	0.99	-1791.36
S_TRA_10-1001-2029944994	9.34	0.010	0.994	0.000	-1856.61	1.730000	9.23	0.99	-1836.41
M_PAT_01-1001-1151405807	9.72	0.020	0.999	0.000	-1718.92	3.000000	8.18	1.00	-1454.84

Continued on next page

Table C.6 – continued from previous page

	MANET						Reactive		
	$ANCU$		B		OFV				
Graph-Replication	μ	σ	μ	σ	μ	σ	$ANCU$	B	OFV
M_PAT_01-1001-1399793545	9.43	0.020	0.999	0.000	-1668.92	3.470000	7.95	1.00	-1415.84
M_PAT_01-1001-1787977610	9.63	0.000	0.999	0.000	-1703.92	0.000000	8.40	1.00	-1493.83
M_PAT_01-1001-430383267	9.07	0.020	0.999	0.000	-1607.92	3.000000	7.58	1.00	-1352.84
M_PAT_01-1001-931132120	9.49	0.030	0.999	0.000	-1678.92	4.580000	8.26	1.00	-1469.83
M_PAT_02-1003-1025450143	9.46	0.080	0.999	0.000	-1439.95	12.120000	8.04	1.00	-1230.85
M_PAT_02-1003-1067491402	9.51	0.040	0.999	0.000	-1446.94	6.000000	7.92	1.00	-1212.86
M_PAT_02-1003-1492747276	9.34	0.010	0.999	0.000	-1421.95	1.730000	7.73	1.00	-1185.85
M_PAT_02-1003-1802077908	9.26	0.080	0.999	0.000	-1409.95	12.120000	7.88	1.00	-1206.85
M_PAT_02-1003-758160770	9.08	0.040	0.999	0.000	-1383.94	5.200000	7.92	1.00	-1212.86
M_PAT_03-1002-144105714	8.31	0.140	0.999	0.000	-1477.94	24.250000	7.58	1.00	-1352.85
M_PAT_03-1002-1480179383	9.49	0.020	0.999	0.000	-1678.94	3.460000	8.33	1.00	-1481.85
M_PAT_03-1002-2092707011	9.23	0.090	0.999	0.000	-1635.94	15.100000	7.96	1.00	-1418.85
M_PAT_03-1002-302106014	8.95	0.060	0.999	0.000	-1587.94	9.640000	8.04	1.00	-1430.85
M_PAT_03-1002-323497351	9.02	0.030	0.999	0.000	-1598.94	5.200000	8.56	1.00	-1520.84
M_PAT_05-1002-1122507549	9.34	0.090	0.998	0.000	-1508.91	14.800000	7.87	1.00	-1278.83
M_PAT_05-1002-1239423887	9.22	0.020	0.998	0.000	-1490.91	3.460000	8.15	1.00	-1323.79
M_PAT_05-1002-2128696202	9.51	0.040	0.998	0.000	-1534.91	6.240000	8.04	1.00	-1305.82
M_PAT_05-1002-654219320	9.27	0.000	0.998	0.000	-1497.91	0.000000	8.02	1.00	-1302.82
M_PAT_05-1002-98128893	9.53	0.110	0.998	0.000	-1538.91	17.320000	7.90	1.00	-1284.82
M_PAT_06-1001-1113468223	9.32	0.130	0.998	0.000	-1591.91	22.110000	7.84	1.00	-1347.83
M_PAT_06-1001-1261438101	9.33	0.100	0.998	0.000	-1594.92	17.320000	7.80	1.00	-1341.82
M_PAT_06-1001-1936904785	9.02	0.020	0.998	0.000	-1542.92	3.000000	7.53	1.00	-1296.82
M_PAT_06-1001-435941795	9.04	0.020	0.998	0.000	-1545.92	3.000000	7.82	1.00	-1344.83
M_PAT_06-1001-802189198	9.04	0.130	0.998	0.000	-1545.92	20.780000	7.49	1.00	-1290.82
M_PAT_08-1001-1387689409	9.31	0.020	0.999	0.000	-1648.94	3.460000	6.98	1.00	-1250.87
M_PAT_08-1001-1489712249	8.95	0.030	0.999	0.000	-1586.94	5.200000	7.84	1.00	-1397.88
M_PAT_08-1001-1540178950	9.26	0.070	0.999	0.000	-1639.94	12.490000	7.51	1.00	-1340.87
M_PAT_08-1001-730794917	9.11	0.010	0.999	0.000	-1614.94	1.730000	7.53	1.00	-1343.87
M_PAT_08-1001-929677267	9.52	0.050	0.999	0.000	-1684.94	8.660000	7.75	1.00	-1382.86
M_PAT_09-1001-1436228507	9.18	0.060	0.999	0.000	-1597.92	10.390000	6.89	1.00	-1213.83
M_PAT_09-1001-167149800	9.07	0.050	0.999	0.000	-1579.92	7.940000	7.45	1.00	-1306.82
M_PAT_09-1001-2144826461	8.98	0.040	0.999	0.000	-1563.92	6.930000	7.25	1.00	-1273.83
M_PAT_09-1001-495893593	9.71	0.000	0.999	0.000	-1687.92	0.000000	7.63	1.00	-1336.83
M_PAT_09-1001-677311175	9.65	0.030	0.999	0.000	-1676.92	4.580000	7.43	1.00	-1303.83
M_POL_01-1001-1228650629	9.50	0.000	0.992	0.000	-766.78	0.000000	8.77	0.98	-709.58
M_POL_01-1001-1383711440	9.92	0.000	0.992	0.000	-799.78	0.000000	9.15	0.98	-739.51
M_POL_01-1001-1821232780	9.85	0.000	0.991	0.000	-793.77	0.000000	9.46	0.98	-763.54
M_POL_01-1001-266331284	8.92	0.000	0.992	0.000	-721.78	0.000000	8.42	0.98	-682.59
M_POL_01-1001-439757398	9.77	0.000	0.992	0.000	-787.78	0.000000	8.81	0.98	-712.58
M_POL_02-1003-1674340543	9.96	0.000	0.988	0.000	-802.68	0.000000	9.35	0.98	-754.58
M_POL_02-1003-1847766657	9.77	0.000	0.987	0.000	-787.67	0.000000	9.50	0.98	-766.52
M_POL_02-1003-662602355	9.77	0.000	0.987	0.000	-787.67	0.000000	9.46	0.98	-763.52
M_POL_02-1003-667122018	9.88	0.000	0.988	0.000	-796.68	0.000000	9.58	0.98	-772.50
M_POL_02-1003-840548132	9.77	0.000	0.987	0.000	-787.67	0.000000	9.04	0.98	-730.47
M_POL_03-1002-1585906475	9.50	0.000	0.991	0.000	-766.78	0.000000	9.27	0.98	-748.60
M_POL_03-1002-1590426138	9.81	0.000	0.992	0.000	-790.78	0.000000	9.23	0.98	-745.60
M_POL_03-1002-386896533	9.95	0.020	0.991	0.000	-801.77	1.730000	9.31	0.99	-751.62
M_POL_03-1002-391416196	9.38	0.000	0.992	0.000	-757.78	0.000000	9.42	0.99	-760.64
M_POL_03-1002-943905481	9.81	0.000	0.991	0.000	-790.77	0.000000	9.27	0.99	-748.62
M_POL_04-1003-1443533901	9.73	0.000	0.987	0.000	-784.67	0.000000	9.23	0.97	-745.23
M_POL_04-1003-1621479678	9.81	0.000	0.987	0.000	-790.67	0.000000	9.12	0.98	-736.43

Continued on next page

Table C.6 – continued from previous page

	MANET						Reactive		
	ANCU		*B*		*OFV*				
Graph-Replication	μ	σ	μ	σ	μ	σ	*ANCU*	*B*	*OFV*
M_POL_04-1003-1799425455	9.40	0.020	0.988	0.000	-758.68	1.730000	8.81	0.98	-712.35
M_POL_04-1003-262889262	9.38	0.000	0.986	0.000	-757.65	0.000000	9.12	0.97	-736.27
M_POL_04-1003-855471413	9.77	0.000	0.987	0.000	-787.67	0.000000	9.23	0.98	-745.45
M_POL_05-1002-1764929893	9.77	0.000	0.992	0.000	-787.78	0.000000	9.31	0.98	-751.59
M_POL_05-1002-214548060	9.38	0.000	0.991	0.000	-757.78	0.000000	9.15	0.98	-739.56
M_POL_05-1002-41121946	9.88	0.000	0.991	0.000	-796.77	0.000000	9.73	0.98	-784.55
M_POL_05-1002-584285254	9.86	0.020	0.992	0.000	-794.78	1.730000	9.19	0.98	-742.60
M_POL_05-1002-825495514	9.96	0.000	0.992	0.000	-802.78	0.000000	9.58	0.98	-772.59
M_POL_06-1002-1088433337	9.77	0.000	0.989	0.000	-787.73	0.000000	9.23	0.98	-745.58
M_POL_06-1002-139959632	9.27	0.000	0.989	0.000	-748.72	0.000000	8.65	0.99	-700.63
M_POL_06-1002-1690341465	9.92	0.000	0.989	0.000	-799.72	0.000000	9.08	0.99	-733.62
M_POL_06-1002-1699380791	10.00	0.000	0.990	0.000	-805.74	0.000000	9.04	0.99	-730.64
M_POL_06-1002-1872806905	8.51	0.020	0.990	0.000	-689.74	1.730000	8.58	0.99	-694.64
M_POL_07-1001-1009325246	9.85	0.000	0.992	0.000	-793.78	0.000000	9.62	0.98	-775.54
M_POL_07-1001-1013844909	9.69	0.000	0.992	0.000	-781.79	0.000000	9.35	0.98	-754.58
M_POL_07-1001-1018364572	9.19	0.000	0.992	0.000	-742.79	0.000000	9.12	0.98	-736.59
M_POL_07-1001-1606427060	9.81	0.000	0.992	0.000	-790.79	0.000000	9.54	0.98	-769.54
M_POL_07-1001-1962318614	9.88	0.000	0.992	0.000	-796.80	0.000000	9.15	0.98	-739.57
M_POL_08-1001-1464054486	10.00	0.000	0.994	0.000	-805.85	0.000000	9.38	0.99	-757.75
M_POL_08-1001-1833791680	10.00	0.000	0.994	0.000	-805.85	0.000000	9.42	0.99	-760.74
M_POL_08-1001-2056636637	9.77	0.000	0.994	0.000	-787.85	0.000000	9.15	0.99	-739.74
M_POL_08-1001-278890184	10.00	0.000	0.994	0.000	-805.85	0.000000	9.88	0.99	-796.70
M_POL_08-1001-871472335	9.92	0.000	0.994	0.000	-799.84	0.000000	9.38	0.99	-757.74
M_POL_09-1002-1206713967	9.88	0.000	0.986	0.000	-796.64	0.000000	9.54	0.99	-769.61
M_POL_09-1002-1794776455	9.92	0.040	0.986	0.000	-799.63	3.000000	9.73	0.98	-784.57
M_POL_09-1002-1799296118	9.27	0.000	0.987	0.000	-748.66	0.000000	9.08	0.98	-733.54
M_POL_09-1002-595766513	9.69	0.000	0.987	0.000	-781.67	0.000000	9.38	0.99	-757.62
M_POL_09-1002-614131816	9.50	0.000	0.985	0.000	-766.62	0.000000	9.35	0.98	-754.45
M_POL_10-1002-1183829001	9.50	0.000	0.994	0.000	-766.83	0.000000	8.77	0.99	-709.65
M_POL_10-1002-1357255115	9.38	0.000	0.994	0.000	-757.83	0.000000	8.46	0.99	-685.64
M_POL_10-1002-1429558924	9.96	0.000	0.993	0.000	-802.83	0.000000	9.00	0.99	-727.69
M_POL_10-1002-1771891489	9.76	0.020	0.993	0.000	-786.83	1.730000	9.27	0.99	-748.64
M_POL_10-1002-760153301	9.88	0.000	0.994	0.000	-796.83	0.000000	9.23	0.99	-745.66
M_RAN_01-1002-142114914	9.65	0.000	1.000	0.000	-778.99	0.000000	9.08	1.00	-733.99
M_RAN_01-1002-378805511	9.09	0.020	1.000	0.000	-734.99	1.730000	8.88	1.00	-718.99
M_RAN_01-1002-383325174	9.54	0.000	1.000	0.000	-769.99	0.000000	8.85	1.00	-715.99
M_RAN_01-1002-556751288	9.54	0.000	1.000	0.000	-769.99	0.000000	8.31	1.00	-673.99
M_RAN_01-1002-784115908	9.12	0.000	1.000	0.000	-736.99	0.000000	8.27	1.00	-670.99
M_RAN_02-1001-112139015	9.73	0.000	1.000	0.000	-785.00	0.000000	9.04	1.00	-731.00
M_RAN_02-1001-1123877203	9.63	0.020	1.000	0.000	-777.00	1.730000	8.92	1.00	-722.00
M_RAN_02-1001-1538513577	9.08	0.000	1.000	0.000	-734.00	0.000000	9.04	1.00	-731.00
M_RAN_02-1001-1716459354	9.65	0.000	1.000	0.000	-779.00	0.000000	9.00	1.00	-728.00
M_RAN_02-1001-334983972	9.33	0.020	1.000	0.000	-754.00	1.730000	8.42	1.00	-683.00
M_RAN_03-1002-1023832539	9.05	0.020	1.000	0.000	-732.00	1.730000	8.27	1.00	-671.00
M_RAN_03-1002-1201778316	9.35	0.000	1.000	0.000	-755.00	0.000000	8.42	1.00	-683.00
M_RAN_03-1002-1744941624	9.81	0.000	1.000	0.000	-791.00	0.000000	9.04	1.00	-731.00
M_RAN_03-1002-190040128	7.68	0.160	1.000	0.000	-625.00	12.120000	7.50	1.00	-611.00
M_RAN_03-1002-381831545	10.00	0.000	1.000	0.000	-806.00	0.000000	9.04	1.00	-731.00
M_RAN_04-1001-1644896960	9.50	0.000	1.000	0.000	-767.00	0.000000	8.58	1.00	-694.99
M_RAN_04-1001-405794152	9.88	0.000	1.000	0.000	-797.00	0.000000	9.50	1.00	-766.99
M_RAN_04-1001-445887018	9.46	0.000	1.000	0.000	-764.00	0.000000	8.50	1.00	-688.99

Continued on next page

Table C.6 – continued from previous page

	MANET						Reactive		
	ANCU		*B*		*OFV*				
Graph-Replication	μ	σ	μ	σ	μ	σ	*ANCU*	*B*	*OFV*
M_RAN_04-1001-459732658	8.40	0.090	1.000	0.000	-681.00	6.930000	7.50	1.00	-610.99
M_RAN_04-1001-583739929	9.88	0.000	1.000	0.000	-797.00	0.000000	9.00	1.00	-727.99
M_RAN_05-1002-1319772295	9.58	0.000	0.999	0.000	-772.97	0.000000	9.58	1.00	-772.98
M_RAN_05-1002-371298590	9.77	0.000	0.999	0.000	-787.97	0.000000	9.73	1.00	-784.98
M_RAN_05-1002-553764030	8.74	0.040	0.999	0.000	-707.97	3.460000	8.77	1.00	-709.98
M_RAN_05-1002-727190144	9.12	0.000	0.999	0.000	-736.97	0.000000	8.81	1.00	-712.98
M_RAN_05-1002-963880741	9.85	0.000	0.999	0.000	-793.97	0.000000	9.58	1.00	-772.98
M_RAN_06-1001-1165789125	9.85	0.000	1.000	0.000	-793.99	0.000000	8.35	1.00	-676.99
M_RAN_06-1001-1289796396	9.68	0.020	1.000	0.000	-780.99	1.730000	8.77	1.00	-709.99
M_RAN_06-1001-154050937	9.19	0.000	1.000	0.000	-742.99	0.000000	8.35	1.00	-676.99
M_RAN_06-1001-2074169964	8.96	0.000	1.000	0.000	-724.99	0.000000	8.42	1.00	-682.99
M_RAN_06-1001-711059885	9.50	0.000	1.000	0.000	-766.99	0.000000	8.23	1.00	-667.99
M_RAN_08-1001-124075038	9.87	0.020	1.000	0.000	-795.99	1.730000	9.38	1.00	-758.00
M_RAN_08-1001-1866248288	9.91	0.020	1.000	0.000	-798.99	1.730000	9.42	1.00	-761.00
M_RAN_08-1001-1870767951	9.72	0.020	1.000	0.000	-783.99	1.730000	9.00	1.00	-727.99
M_RAN_08-1001-2044194065	9.77	0.000	1.000	0.000	-787.99	0.000000	9.27	1.00	-749.00
M_RAN_08-1001-716657189	9.96	0.000	1.000	0.000	-802.99	0.000000	9.42	1.00	-761.00
M_RAN_09-1002-1281237400	9.81	0.000	1.000	0.000	-790.99	0.000000	9.00	1.00	-727.99
M_RAN_09-1002-381895887	9.31	0.000	1.000	0.000	-751.99	0.000000	8.81	1.00	-712.99
M_RAN_09-1002-672238339	9.92	0.000	1.000	0.000	-799.99	0.000000	9.04	1.00	-730.99
M_RAN_09-1002-676758002	9.42	0.040	1.000	0.000	-760.99	3.000000	8.81	1.00	-712.99
M_RAN_09-1002-962580791	9.90	0.020	1.000	0.000	-797.99	1.730000	9.38	1.00	-757.99
M_RAN_10-1003-1054470091	9.81	0.000	1.000	0.000	-791.00	0.000000	9.46	1.00	-764.00
M_RAN_10-1003-1103888934	8.99	0.020	1.000	0.000	-727.00	1.730000	8.85	1.00	-716.00
M_RAN_10-1003-1281834711	9.19	0.000	1.000	0.000	-743.00	0.000000	9.46	1.00	-764.00
M_RAN_10-1003-2066208279	9.85	0.000	1.000	0.000	-794.00	0.000000	9.38	1.00	-758.00
M_RAN_10-1003-319515366	9.54	0.000	1.000	0.000	-770.00	0.000000	9.19	1.00	-743.00
M_SAR_01-1001-1024494192	19.66	0.000	0.999	0.000	-2638.97	0.000000	19.75	1.00	-2650.97
M_SAR_01-1001-144595623	19.98	0.010	0.999	0.000	-2681.97	1.730000	20.00	1.00	-2683.97
M_SAR_01-1001-1601114197	19.73	0.000	0.999	0.000	-2647.97	0.000000	19.84	1.00	-2662.98
M_SAR_01-1001-1987268301	19.80	0.000	0.999	0.000	-2656.97	0.000000	19.98	1.00	-2680.97
M_SAR_01-1001-525775300	19.86	0.000	0.999	0.000	-2665.97	0.000000	19.77	1.00	-2653.97
M_SAR_02-1001-1067444979	19.73	0.000	0.999	0.000	-2647.94	0.000000	19.59	1.00	-2629.95
M_SAR_02-1001-1344228442	19.64	0.000	0.999	0.000	-2635.94	0.000000	19.73	1.00	-2647.95
M_SAR_02-1001-1758410052	19.89	0.000	0.999	0.000	-2668.94	0.000000	19.77	1.00	-2653.94
M_SAR_02-1001-1847921761	19.77	0.000	0.999	0.000	-2653.94	0.000000	19.59	1.00	-2629.94
M_SAR_02-1001-2030219088	20.00	0.000	0.999	0.000	-2683.94	0.000000	19.68	1.00	-2641.94
M_SAR_03-1002-1693483827	19.93	0.000	0.999	0.000	-2674.97	0.000000	19.93	1.00	-2674.97
M_SAR_03-1002-1965292863	20.00	0.000	0.999	0.000	-2683.97	0.000000	19.84	1.00	-2662.97
M_SAR_03-1002-2038219549	19.84	0.000	0.999	0.000	-2662.97	0.000000	19.75	1.00	-2650.97
M_SAR_03-1002-321502535	19.85	0.010	0.999	0.000	-2663.97	1.730000	19.66	1.00	-2638.97
M_SAR_03-1002-581700975	19.78	0.030	0.999	0.000	-2654.97	3.460000	19.68	1.00	-2641.97
M_SAR_04-1002-1050443968	19.89	0.000	0.999	0.000	-2668.97	0.000000	19.91	1.00	-2671.97
M_SAR_04-1002-217106321	18.67	0.010	0.999	0.000	-2507.97	1.730000	19.41	1.00	-2605.97
M_SAR_04-1002-551725076	19.73	0.000	0.999	0.000	-2647.97	0.000000	19.84	1.00	-2662.97
M_SAR_04-1002-8107004	20.00	0.000	0.999	0.000	-2683.97	0.000000	19.89	1.00	-2668.97
M_SAR_04-1002-908071394	19.84	0.000	0.999	0.000	-2662.97	0.000000	19.82	1.00	-2659.97
M_SAR_05-1001-102593140	19.85	0.010	0.999	0.000	-2663.96	1.730000	19.75	1.00	-2650.95
M_SAR_05-1001-1416739140	19.82	0.000	0.999	0.000	-2659.96	0.000000	19.77	1.00	-2653.96
M_SAR_05-1001-147492320	19.64	0.010	0.999	0.000	-2636.96	1.730000	19.66	1.00	-2638.95
M_SAR_05-1001-374402176	19.89	0.000	0.999	0.000	-2668.96	0.000000	19.59	1.00	-2629.96

Continued on next page

Table C.6 – continued from previous page

	MANET						Reactive		
	ANCU		*B*		*OFV*				
Graph-Replication	μ	σ	μ	σ	μ	σ	*ANCU*	*B*	*OFV*
M_SAR_05-1001-770673247	19.80	0.000	0.999	0.000	-2656.96	0.000000	19.73	1.00	-2647.95
M_SAR_06-1001-10094173	19.93	0.000	0.999	0.000	-2674.97	0.000000	19.91	1.00	-2671.97
M_SAR_06-1001-281903209	19.61	0.010	0.999	0.000	-2631.97	1.730000	19.55	1.00	-2623.97
M_SAR_06-1001-384351030	19.80	0.000	0.999	0.000	-2656.97	0.000000	19.80	1.00	-2656.97
M_SAR_06-1001-656160066	19.98	0.000	0.999	0.000	-2680.97	0.000000	20.00	1.00	-2683.97
M_SAR_06-1001-927969102	19.77	0.000	0.999	0.000	-2653.97	0.000000	19.68	1.00	-2641.97
M_SAR_07-1003-1057405564	19.86	0.000	0.999	0.000	-2665.96	0.000000	19.98	1.00	-2680.96
M_SAR_07-1003-1556124456	19.73	0.000	0.999	0.000	-2647.96	0.000000	19.73	1.00	-2647.96
M_SAR_07-1003-2099742528	19.98	0.000	0.999	0.000	-2680.97	0.000000	19.95	1.00	-2677.97
M_SAR_07-1003-2104716955	19.78	0.030	0.999	0.000	-2654.96	3.460000	19.77	1.00	-2653.97
M_SAR_07-1003-643223954	19.80	0.000	0.999	0.000	-2656.97	0.000000	19.68	1.00	-2641.96
M_SAR_08-1003-1428675163	19.65	0.010	0.999	0.000	-2637.96	1.730000	19.57	1.00	-2626.96
M_SAR_08-1003-1718394738	19.95	0.010	0.999	0.000	-2676.95	1.730000	19.80	1.00	-2656.95
M_SAR_08-1003-465901054	19.84	0.000	0.999	0.000	-2662.95	0.000000	19.68	1.00	-2641.95
M_SAR_08-1003-613248055	19.93	0.000	0.999	0.000	-2674.96	0.000000	19.75	1.00	-2650.95
M_SAR_08-1003-737710090	19.75	0.000	0.999	0.000	-2650.95	0.000000	19.84	1.00	-2662.96
M_SAR_09-1001-1523161299	19.90	0.010	0.999	0.000	-2670.95	1.730000	19.73	1.00	-2647.95
M_SAR_09-1001-1937342909	19.80	0.010	0.999	0.000	-2657.95	1.730000	19.34	1.00	-2596.95
M_SAR_09-1001-2084689910	19.93	0.000	0.999	0.000	-2674.94	0.000000	19.59	1.00	-2629.95
M_SAR_09-1001-623196909	19.91	0.000	0.999	0.000	-2671.94	0.000000	19.66	1.00	-2638.95
M_SAR_09-1001-832196226	19.96	0.010	0.999	0.000	-2678.94	1.730000	19.80	1.00	-2656.95
M_SAR_10-1001-1493185400	19.95	0.000	0.999	0.000	-2677.95	0.000000	19.66	1.00	-2638.95
M_SAR_10-1001-179039400	19.82	0.000	0.999	0.000	-2659.94	0.000000	19.45	1.00	-2611.95
M_SAR_10-1001-1917315864	19.84	0.000	0.999	0.000	-2662.94	0.000000	19.64	1.00	-2635.95
M_SAR_10-1001-455822863	19.89	0.000	0.999	0.000	-2668.95	0.000000	19.80	1.00	-2656.96
M_SAR_10-1001-802220327	19.91	0.000	0.999	0.000	-2671.95	0.000000	19.50	1.00	-2617.95
M_TRA_01-1003-1397660399	20.00	0.000	0.946	0.000	-853.25	0.000000	20.00	0.95	-853.28
M_TRA_01-1003-147115108	19.36	0.000	0.947	0.000	-826.25	0.000000	19.36	0.95	-826.27
M_TRA_01-1003-313450289	19.64	0.000	0.946	0.000	-838.25	0.000000	19.64	0.95	-838.29
M_TRA_01-1003-606195898	19.64	0.000	0.946	0.000	-838.24	0.000000	19.64	0.95	-838.26
M_TRA_01-1003-812169181	19.43	0.000	0.948	0.000	-829.28	0.000000	19.43	0.95	-829.32
M_TRA_02-1001-1035014138	19.06	0.060	0.940	0.000	-930.04	3.000000	19.13	0.95	-933.14
M_TRA_02-1001-1655455682	20.00	0.000	0.939	0.000	-975.02	0.000000	20.00	0.94	-975.07
M_TRA_02-1001-1791528313	19.19	0.000	0.940	0.000	-936.03	0.000000	19.06	0.94	-930.10
M_TRA_02-1001-2020841326	19.90	0.040	0.936	0.000	-969.97	1.730000	19.88	0.94	-969.06
M_TRA_02-1001-887499024	19.31	0.000	0.942	0.000	-942.07	0.000000	19.38	0.94	-945.11
M_TRA_03-1003-1105369554	19.83	0.000	0.948	0.000	-1088.06	0.000000	19.83	0.95	-1088.02
M_TRA_03-1003-1236754409	19.89	0.000	0.949	0.000	-1091.07	0.000000	19.89	0.95	-1091.09
M_TRA_03-1003-1765903964	19.67	0.000	0.952	0.000	-1079.13	0.000000	19.72	0.95	-1082.11
M_TRA_03-1003-829208968	19.96	0.030	0.948	0.000	-1095.07	1.730000	20.00	0.95	-1097.10
M_TRA_03-1003-906487204	19.78	0.000	0.951	0.000	-1085.11	0.010000	19.78	0.95	-1085.15
M_TRA_04-1002-1770916966	19.50	0.000	0.946	0.000	-1070.03	0.000000	19.72	0.95	-1082.09
M_TRA_04-1002-1899898664	19.54	0.030	0.951	0.000	-1072.11	1.730000	17.67	0.96	-971.21
M_TRA_04-1002-73188132	18.91	0.080	0.940	0.000	-1037.92	4.580000	19.11	0.95	-1049.10
M_TRA_04-1002-803623194	20.00	0.000	0.947	0.000	-1097.05	0.010000	20.00	0.95	-1097.14
M_TRA_04-1002-902006116	20.00	0.000	0.948	0.000	-1097.07	0.000000	20.00	0.95	-1097.14
M_TRA_05-1001-1513744560	19.47	0.000	0.949	0.000	-1128.04	0.000000	19.53	0.95	-1131.13
M_TRA_05-1001-1811009832	19.32	0.000	0.956	0.000	-1119.16	0.000000	19.58	0.96	-1134.23
M_TRA_05-1001-1822907079	20.00	0.000	0.951	0.000	-1158.07	0.000000	20.00	0.95	-1158.14
M_TRA_05-1001-1844011765	19.89	0.000	0.953	0.000	-1152.12	0.000000	19.89	0.96	-1152.19
M_TRA_05-1001-412949427	19.65	0.030	0.956	0.000	-1138.17	1.730000	17.05	0.96	-990.25

Continued on next page

Table C.6 – continued from previous page

	MANET						Reactive		
	$ANCU$		B		OFV				
Graph-Replication	μ	σ	μ	σ	μ	σ	$ANCU$	B	OFV
M_TRA_06-1003-1020000095	19.74	0.000	0.949	0.000	-1143.03	0.000000	19.95	0.95	-1155.09
M_TRA_06-1003-1317265367	19.63	0.000	0.956	0.000	-1137.16	0.000000	19.63	0.96	-1137.25
M_TRA_06-1003-1338370053	19.47	0.000	0.953	0.000	-1128.11	0.010000	19.58	0.96	-1134.17
M_TRA_06-1003-1647532572	19.63	0.000	0.949	0.000	-1137.04	0.000000	19.63	0.96	-1137.15
M_TRA_06-1003-237574920	19.84	0.000	0.954	0.000	-1149.13	0.000000	19.84	0.96	-1149.17
M_TRA_07-1001-1493262751	20.00	0.000	0.955	0.000	-1158.14	0.000000	20.00	0.96	-1158.15
M_TRA_07-1001-1802425270	20.00	0.000	0.961	0.000	-1158.26	0.000000	20.00	0.96	-1158.26
M_TRA_07-1001-1823529956	20.00	0.000	0.959	0.000	-1158.22	0.000000	20.00	0.96	-1158.21
M_TRA_07-1001-392467618	19.79	0.000	0.961	0.000	-1146.26	0.000000	19.79	0.96	-1146.26
M_TRA_07-1001-755736756	20.00	0.000	0.957	0.000	-1158.19	0.000000	20.00	0.96	-1158.22
M_TRA_08-1001-1109511804	19.85	0.000	0.962	0.000	-1210.23	0.000000	19.85	0.96	-1210.26
M_TRA_08-1001-1111914961	19.90	0.000	0.964	0.000	-1213.27	0.000000	18.90	0.96	-1153.29
M_TRA_08-1001-1320291401	19.25	0.050	0.963	0.001	-1174.26	3.010000	19.10	0.96	-1165.30
M_TRA_08-1001-1924316035	19.85	0.000	0.963	0.000	-1210.27	0.010000	18.90	0.96	-1153.28
M_TRA_08-1001-297110730	19.45	0.000	0.961	0.000	-1186.23	0.000000	19.50	0.96	-1189.25
M_TRA_09-1003-1184841647	19.90	0.000	0.949	0.000	-1212.98	0.000000	19.90	0.96	-1213.14
M_TRA_09-1003-1578086684	19.87	0.030	0.953	0.000	-1211.05	1.740000	19.85	0.96	-1210.16
M_TRA_09-1003-1746993135	19.60	0.000	0.949	0.000	-1194.98	0.000000	19.75	0.96	-1204.15
M_TRA_09-1003-763282453	19.85	0.000	0.952	0.000	-1210.04	0.000000	19.80	0.96	-1207.13
M_TRA_09-1003-974062050	20.00	0.000	0.950	0.000	-1219.00	0.010000	20.00	0.96	-1219.17
M_TRA_10-1001-1054079669	19.47	0.000	0.952	0.000	-1128.08	0.000000	19.47	0.96	-1128.16
M_TRA_10-1001-1363242188	19.68	0.000	0.950	0.000	-1140.05	0.000000	19.63	0.95	-1137.08
M_TRA_10-1001-1384346874	19.72	0.030	0.947	0.000	-1142.00	1.730000	19.74	0.95	-1143.03
M_TRA_10-1001-1770500978	19.61	0.030	0.948	0.001	-1136.01	1.750000	19.63	0.95	-1137.07
M_TRA_10-1001-2100768183	19.49	0.060	0.948	0.000	-1129.01	3.470000	19.53	0.95	-1131.10
L_SAR_01-1002-1636428977	40.45	0.000	1.000	0.000	-5384.00	0.000000	40.61	1.00	-5405.00
L_SAR_01-1002-1907615136	40.73	0.040	1.000	0.000	-5420.00	5.200000	40.20	1.00	-5351.00
L_SAR_01-1002-2121420767	40.61	0.000	1.000	0.000	-5405.00	0.000000	40.55	1.00	-5396.00
L_SAR_01-1002-212289459	40.84	0.000	1.000	0.000	-5435.00	0.000000	40.66	1.00	-5411.00
L_SAR_01-1002-322737741	40.95	0.000	1.000	0.000	-5450.00	0.000000	40.73	1.00	-5420.00
L_SAR_02-1002-318840955	40.83	0.010	1.000	0.000	-5433.00	1.730000	40.84	1.00	-5435.00
L_SAR_02-1002-384676708	40.68	0.000	1.000	0.000	-5414.00	0.000000	40.55	1.00	-5396.00
L_SAR_02-1002-82437009	40.57	0.000	1.000	0.000	-5399.00	0.000000	40.45	1.00	-5384.00
L_SAR_02-1002-859432993	40.86	0.010	1.000	0.000	-5437.00	1.730000	40.86	1.00	-5438.00
L_SAR_02-1002-974400938	40.80	0.010	1.000	0.000	-5430.00	1.730000	40.57	1.00	-5399.00
L_SAR_03-1002-1186258176	40.75	0.000	1.000	0.000	-5423.00	0.000000	40.75	1.00	-5423.00
L_SAR_03-1002-1270172581	40.77	0.130	1.000	0.000	-5425.00	17.320000	40.89	1.00	-5441.00
L_SAR_03-1002-1832778833	40.75	0.000	1.000	0.000	-5423.00	0.000000	40.64	1.00	-5408.00
L_SAR_03-1002-2127927599	40.80	0.020	1.000	0.000	-5429.00	3.000000	40.82	1.00	-5432.00
L_SAR_03-1002-958893556	40.86	0.000	1.000	0.000	-5438.00	0.000000	40.80	1.00	-5429.00
L_SAR_04-1002-1002299107	40.74	0.010	1.000	0.000	-5422.00	1.730000	40.68	1.00	-5414.00
L_SAR_04-1002-1198610187	40.16	0.020	1.000	0.000	-5345.00	3.000000	40.23	1.00	-5354.00
L_SAR_04-1002-1880704047	40.77	0.000	1.000	0.000	-5426.00	0.000000	40.70	1.00	-5417.00
L_SAR_04-1002-236290842	40.73	0.010	1.000	0.000	-5421.00	1.730000	40.66	1.00	-5411.00
L_SAR_04-1002-671577138	40.57	0.000	1.000	0.000	-5399.00	0.000000	40.80	1.00	-5429.00
L_SAR_05-1001-1518512550	40.75	0.000	1.000	0.000	-5423.00	0.000000	40.68	1.00	-5413.99
L_SAR_05-1001-249433843	40.80	0.000	1.000	0.000	-5429.00	0.000000	40.59	1.00	-5401.99
L_SAR_05-1001-492137732	40.48	0.000	1.000	0.000	-5387.00	0.000000	40.41	1.00	-5377.99
L_SAR_05-1001-532230598	40.82	0.000	1.000	0.000	-5432.00	0.000000	40.23	1.00	-5353.99
L_SAR_05-1001-827379364	40.28	0.010	1.000	0.000	-5361.00	1.730000	40.52	1.00	-5392.99
L_SAR_06-1001-1116228187	40.92	0.010	1.000	0.000	-5445.00	1.730000	40.66	1.00	-5411.00

Continued on next page

Table C.6 – continued from previous page

	MANET						Reactive		
	ANCU		*B*		*OFV*				
Graph-Replication	μ	σ	μ	σ	μ	σ	*ANCU*	*B*	*OFV*
L_SAR_06-1001-1151801390	40.84	0.000	1.000	0.000	-5435.00	0.000000	40.75	1.00	-5423.00
L_SAR_06-1001-469707530	40.93	0.000	1.000	0.000	-5447.00	0.000000	40.82	1.00	-5432.00
L_SAR_06-1001-573064879	40.71	0.010	1.000	0.000	-5418.00	1.730000	40.73	1.00	-5420.00
L_SAR_06-1001-608638082	40.95	0.000	1.000	0.000	-5450.00	0.000000	40.82	1.00	-5432.00
L_SAR_07-1002-1175763997	40.83	0.010	1.000	0.000	-5434.00	1.730000	40.55	1.00	-5396.00
L_SAR_07-1002-177871449	40.43	0.000	1.000	0.000	-5381.00	0.000000	40.77	1.00	-5426.00
L_SAR_07-1002-1822284654	40.80	0.040	1.000	0.000	-5429.00	5.200000	40.68	1.00	-5414.00
L_SAR_07-1002-1925642003	40.64	0.000	1.000	0.000	-5408.00	0.000000	40.57	1.00	-5399.00
L_SAR_07-1002-859965309	40.91	0.000	1.000	0.000	-5444.00	0.000000	40.57	1.00	-5399.00
L_SAR_08-1001-1415480628	40.52	0.000	1.000	0.000	-5393.00	0.000000	40.34	1.00	-5369.00
L_SAR_08-1001-1558930843	40.73	0.010	1.000	0.000	-5421.00	1.730000	40.75	1.00	-5423.00
L_SAR_08-1001-1642845248	40.84	0.000	1.000	0.000	-5435.00	0.000000	40.61	1.00	-5405.00
L_SAR_08-1001-1682938114	40.84	0.040	1.000	0.000	-5435.00	5.200000	40.34	1.00	-5369.00
L_SAR_08-1001-537075688	40.68	0.000	1.000	0.000	-5414.00	0.000000	40.41	1.00	-5378.00
L_SAR_09-1001-1128164210	40.65	0.010	1.000	0.000	-5410.00	1.730000	40.45	1.00	-5384.00
L_SAR_09-1001-1582893450	40.76	0.010	1.000	0.000	-5424.00	1.730000	40.77	1.00	-5426.00
L_SAR_09-1001-1706900721	40.84	0.000	1.000	0.000	-5435.00	0.000000	40.50	1.00	-5390.00
L_SAR_09-1001-81930460	40.83	0.010	1.000	0.000	-5433.00	1.730000	40.68	1.00	-5414.00
L_SAR_09-1001-847938725	40.95	0.000	1.000	0.000	-5450.00	0.000000	40.93	1.00	-5447.00
L_SAR_10-1001-1187700020	40.90	0.010	1.000	0.000	-5443.00	1.730000	40.50	1.00	-5390.00
L_SAR_10-1001-1211662627	40.86	0.000	1.000	0.000	-5438.00	0.000000	40.50	1.00	-5390.00
L_SAR_10-1001-1295577032	40.55	0.010	1.000	0.000	-5397.00	1.730000	40.55	1.00	-5396.00
L_SAR_10-1001-1439027247	40.86	0.020	1.000	0.000	-5438.00	3.000000	40.68	1.00	-5414.00
L_SAR_10-1001-565141970	40.89	0.000	1.000	0.000	-5441.00	0.000000	40.36	1.00	-5372.00
X_POL_01-1001-1290835010	59.81	0.000	0.997	0.000	-4690.92	0.000000	59.15	0.99	-4639.87
X_POL_01-1001-16781876	59.96	0.000	0.997	0.000	-4702.91	0.000000	59.35	0.99	-4654.87
X_POL_01-1001-349620351	59.69	0.000	0.997	0.000	-4681.91	0.000000	59.27	0.99	-4648.87
X_POL_01-1001-853145557	59.41	0.020	0.997	0.000	-4659.91	1.730000	58.92	0.99	-4621.87
X_POL_01-1001-890212389	59.88	0.000	0.997	0.000	-4696.91	0.000000	59.19	0.99	-4642.86
X_POL_02-1002-1195478122	59.92	0.000	0.997	0.000	-4699.92	0.000000	59.31	1.00	-4651.89
X_POL_02-1002-1717536744	59.65	0.000	0.997	0.000	-4678.92	0.000000	59.38	1.00	-4657.89
X_POL_02-1002-1947472634	59.96	0.000	0.997	0.000	-4702.92	0.000000	59.19	1.00	-4642.89
X_POL_02-1002-322047609	59.42	0.000	0.997	0.000	-4660.92	0.000000	58.96	1.00	-4624.89
X_POL_02-1002-862639647	59.96	0.000	0.997	0.000	-4702.92	0.000000	59.31	1.00	-4651.88
X_POL_03-1001-1100121234	59.96	0.000	0.997	0.000	-4702.91	0.000000	59.04	0.99	-4630.86
X_POL_03-1001-116242439	59.81	0.000	0.997	0.000	-4690.91	0.000000	58.31	0.99	-4573.86
X_POL_03-1001-1203023819	59.92	0.000	0.997	0.000	-4699.91	0.000000	58.77	0.99	-4609.86
X_POL_03-1001-1268859572	59.96	0.000	0.997	0.000	-4702.91	0.000000	59.08	0.99	-4633.85
X_POL_03-1001-765334366	59.73	0.000	0.997	0.000	-4684.91	0.000000	59.50	0.99	-4666.86
X_POL_04-1002-121839743	59.73	0.000	0.997	0.000	-4684.91	0.000000	59.04	0.99	-4630.86
X_POL_04-1002-1414426293	59.69	0.000	0.997	0.000	-4681.92	0.000000	58.38	0.99	-4579.86
X_POL_04-1002-206208912	59.92	0.000	0.997	0.000	-4699.92	0.000000	59.04	0.99	-4630.86
X_POL_04-1002-290578081	59.49	0.020	0.997	0.000	-4665.92	1.730000	58.58	0.99	-4594.86
X_POL_04-1002-627313342	59.81	0.040	0.997	0.000	-4690.92	3.000000	58.62	0.99	-4597.85
X_POL_05-1003-1052066683	59.49	0.310	0.997	0.000	-4665.92	24.250000	59.15	0.99	-4639.87
X_POL_05-1003-1154969268	59.65	0.070	0.997	0.000	-4678.92	5.200000	59.04	1.00	-4630.87
X_POL_05-1003-1257871853	59.64	0.060	0.997	0.000	-4677.92	4.580000	58.69	0.99	-4603.86
X_POL_05-1003-178636170	59.00	0.000	0.997	0.000	-4627.92	0.000000	58.96	0.99	-4624.87
X_POL_05-1003-427105476	59.46	0.000	0.997	0.000	-4663.92	0.000000	59.23	0.99	-4645.87
X_POL_06-1002-1162514965	59.83	0.020	0.997	0.000	-4692.91	1.730000	58.96	0.99	-4624.87
X_POL_06-1002-1246884134	60.00	0.000	0.997	0.000	-4705.91	0.000000	59.23	1.00	-4645.87

Continued on next page

Table C.6 – continued from previous page

	MANET						Reactive		
	$ANCU$		B		OFV				
Graph-Replication	μ	σ	μ	σ	μ	σ	$ANCU$	B	OFV
X_POL_06-1002-1476820024	59.54	0.000	0.996	0.000	-4669.91	0.000000	59.77	0.99	-4687.87
X_POL_06-1002-202766890	59.60	0.040	0.997	0.000	-4674.91	3.460000	59.19	0.99	-4642.87
X_POL_06-1002-2101781231	59.96	0.000	0.996	0.000	-4702.90	0.000000	59.12	0.99	-4636.87
X_POL_07-1001-184688238	60.00	0.000	0.996	0.000	-4705.90	0.000000	59.38	0.99	-4657.86
X_POL_07-1001-1886936735	59.50	0.000	0.997	0.000	-4666.91	0.000000	59.46	0.99	-4663.87
X_POL_07-1001-296630149	59.85	0.070	0.996	0.000	-4693.91	5.200000	59.23	0.99	-4645.86
X_POL_07-1001-380999318	59.17	0.020	0.997	0.000	-4640.91	1.730000	59.54	0.99	-4669.87
X_POL_07-1001-407078431	59.77	0.000	0.997	0.000	-4687.91	0.000000	58.62	0.99	-4597.87
X_POL_08-1001-1365332877	60.00	0.000	0.997	0.000	-4705.91	0.000000	59.00	0.99	-4627.86
X_POL_08-1001-1801073937	59.65	0.000	0.997	0.000	-4678.91	0.000000	58.88	0.99	-4618.86
X_POL_08-1001-1885443106	59.27	0.230	0.997	0.000	-4648.91	18.250000	58.27	0.99	-4570.86
X_POL_08-1001-861807671	59.73	0.000	0.997	0.000	-4684.91	0.000000	58.85	0.99	-4615.87
X_POL_08-1001-964710256	59.71	0.020	0.997	0.000	-4682.91	1.730000	58.08	0.99	-4555.86
X_POL_09-1002-1670598610	59.62	0.000	0.997	0.000	-4675.92	0.000000	58.23	1.00	-4567.88
X_POL_09-1002-2124873086	59.54	0.000	0.997	0.000	-4669.92	0.000000	59.23	1.00	-4645.88
X_POL_09-1002-602350646	59.40	0.020	0.997	0.000	-4658.92	1.730000	58.81	1.00	-4612.88
X_POL_09-1002-850819952	59.31	0.140	0.997	0.000	-4651.92	10.820000	59.19	1.00	-4642.88
X_POL_09-1002-98825440	59.85	0.000	0.997	0.000	-4693.92	0.000000	59.73	1.00	-4684.88
X_POL_10-1001-1799580308	59.85	0.040	0.997	0.000	-4693.91	3.000000	58.96	0.99	-4624.87
X_POL_10-1001-338710184	60.00	0.000	0.997	0.000	-4705.91	0.000000	59.04	0.99	-4630.87
X_POL_10-1001-842235390	59.73	0.000	0.997	0.000	-4684.91	0.000000	58.77	0.99	-4609.87
X_POL_10-1001-90240878	60.00	0.000	0.997	0.000	-4705.91	0.000000	59.46	0.99	-4663.87
X_POL_10-1001-963671391	59.77	0.000	0.997	0.000	-4687.92	0.000000	58.96	0.99	-4624.87

Table C.7: Dynamic Stochastic Experimentation Average Instance Computation Time

Instance (3 replications x 5 realizations)	**MANET** (h:mm:ss)	**Reactive** (h:mm:ss)
S_SAR_1001	0:00:06	0:00:06
S_SAR_1002	0:00:05	0:00:08
S_SAR_1003	0:00:09	0:00:08
S_SAR_1004	0:00:13	0:00:06
S_SAR_1005	0:00:08	0:00:06
S_SAR_1006	0:00:08	0:00:06
S_SAR_1007	0:00:10	0:00:10
S_SAR_1008	0:00:10	0:00:05
S_SAR_1009	0:00:11	0:00:11
S_SAR_1010	0:00:09	0:00:09
S_TRA_1001	0:30:47	0:02:53
S_TRA_1002	0:37:39	0:03:39
S_TRA_1003*	7:29:11	0:01:46
S_TRA_1004	0:31:02	0:02:50
S_TRA_1005	0:33:06	0:03:18
S_TRA_1006	0:56:24	0:04:22
S_TRA_1007	0:50:05	0:03:52
S_TRA_1008	0:53:03	0:05:02
S_TRA_1009	0:47:24	0:03:48
S_TRA_1010	0:46:56	0:03:52
M_PAT_1001	3:02:58	0:18:30
M_PAT_1002	4:15:51	0:24:59
M_PAT_1003	4:33:36	0:21:15
M_PAT_1005	1:29:08	0:10:09
M_PAT_1006	3:36:27	0:18:19
M_PAT_1008	4:48:50	0:33:35
M_PAT_1009	3:38:30	0:19:00
M_POL_1001	0:02:56	0:00:43
M_POL_1002	0:01:49	0:00:35
M_POL_1003	0:03:19	0:00:39
M_POL_1004	0:03:41	0:00:20
M_POL_1005	0:04:02	0:00:43
M_POL_1006	0:02:12	0:00:29
M_POL_1007	0:04:56	0:00:39
M_POL_1008	0:01:36	0:00:54
M_POL_1009	0:03:17	0:00:22

Continued on next page

Table C.7 – continued from previous page

Instance (3 replications x 5 realizations)	MANET (h:mm:ss)	Reactive (h:mm:ss)
M_POL_1010	0:03:51	0:00:38
M_RAN_1001	0:37:11	0:11:06
M_RAN_1002	0:43:30	0:20:29
M_RAN_1003	1:26:04	0:26:36
M_RAN_1004	1:10:05	0:22:10
M_RAN_1005	0:07:40	0:02:03
M_RAN_1006	0:17:15	0:08:20
M_RAN_1008	0:42:03	0:10:49
M_RAN_1009	1:02:37	0:16:12
M_RAN_1010	1:11:03	0:19:37
M_SAR_1001	0:08:41	0:07:14
M_SAR_1002	0:04:48	0:02:37
M_SAR_1003	0:08:17	0:07:36
M_SAR_1004	0:23:33	0:05:51
M_SAR_1005	0:06:43	0:04:26
M_SAR_1006	0:09:18	0:05:57
M_SAR_1007	0:06:04	0:03:47
M_SAR_1008	0:08:06	0:05:52
M_SAR_1009	0:09:53	0:05:44
M_SAR_1010	0:05:04	0:03:41
M_TRA_1001	0:01:17	0:00:10
M_TRA_1002	0:02:01	0:00:18
M_TRA_1003	0:02:51	0:00:49
M_TRA_1004	0:05:14	0:00:54
M_TRA_1005	0:03:41	0:00:25
M_TRA_1006	0:03:16	0:00:47
M_TRA_1007	0:01:07	0:00:51
M_TRA_1008	0:06:47	0:00:48
M_TRA_1009	0:04:03	0:01:36
M_TRA_1010	0:06:15	0:01:06
L_SAR_1001	2:56:11	2:22:44
L_SAR_1002	3:38:50	2:31:59
L_SAR_1003	2:19:31	2:13:15
L_SAR_1004	2:34:56	2:10:04
L_SAR_1005	2:12:37	1:50:55
L_SAR_1006	2:58:44	2:45:19
L_SAR_1007	2:09:24	2:07:19

Continued on next page

Table C.7 – continued from previous page

Instance (3 replications x 5 realizations)	**MANET** (h:mm:ss)	**Reactive** (h:mm:ss)
L_SAR_1008	2:03:57	2:12:23
L_SAR_1009	1:50:45	1:58:15
L_SAR_1010	2:40:25	2:13:31
X_POL_1001	3:11:18	0:18:39
X_POL_1002	3:00:49	0:28:41
X_POL_1003	2:21:20	0:34:13
X_POL_1004	4:48:53	0:39:59
X_POL_1005	2:15:43	0:27:02
X_POL_1006	2:55:15	0:19:56
X_POL_1007	3:00:47	0:19:09
X_POL_1008	2:26:44	0:25:22
X_POL_1009	3:32:45	0:33:59
X_POL_1010	2:07:57	0:28:59
Mean	1:12:51	0:23:21
Std.Dev	1:32:41	0:42:08

*Data was lost. Re-evaluated with different code base leading to increased computation time.