Design and Performance Analysis of Genetic Algorithms for Topology Control Problems

by

CEM ŞAFAK ŞAHİN

A Dissertation Submitted to the Graduate Faculty in Engineering
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy
The City University of New York

2010

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

Date	Prof. M. Ümit Uyar Chair of Examining Committee
Date	Prof. Mumtaz Kassir Executive Officer

Prof. M. Ümit Uyar (Mentor)

Prof. Michael Conner

Prof. Jizhong Xiao

Prof. Nelly Fazio

Dr. Ibrahim Hökelek

Giorgio Bertoli

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

Design and Performance Analysis of Genetic Algorithms for Topology Control Problems

by

Cem Şafak Şahin

Advisor: Prof. M. Ümit Uyar

In this dissertation, we present a bio-inspired decentralized topology control mechanism, called force-based genetic algorithm (FGA), where a genetic algorithm (GA) is run by each autonomous mobile node to achieve a uniform spread of mobile nodes and to provide a fully connected network over an unknown area. We present a formal analysis of FGA in terms of convergence speed, uniformity at area coverage, and Lyapunov stability theorem.

This dissertation emphasizes the use of mobile nodes to achieve a uniform distribution over an unknown terrain without a priori information and a central control unit. In contrast, each mobile node running our FGA has to make its own movement direction and speed decisions based on local neighborhood information, such as obstacles and the number of neighbors, without a centralized control unit or global knowledge.

We have implemented simulation software in Java and developed four different testbeds to study the effectiveness of different GA-based topology control frameworks for net-

work performance metrics including node density, speed, and the number of generations that GAs run.

The stochastic behavior of FGA, like all GA-based approaches, makes it difficult to analyze its convergence speed. We built metrically transitive homogeneous and inhomogeneous Markov chain models to analyze the convergence of our FGA with respect to the communication ranges of mobile nodes and the total number of nodes in the system. The Dobrushin contraction coefficient of ergodicity is used for measuring convergence speed for homogeneous and inhomogeneous Markov chain models of our FGA. Furthermore, convergence characteristic analysis helps us to choose the near-optimal values for communication range, the number of mobile nodes, and the mean node degree before sending autonomous mobile nodes to any mission.

Our analytical and experimental results show that our FGA delivers promising results for uniform mobile node distribution over unknown terrains. Since our FGA adapts to local environment rapidly and does not require global network knowledge, it can be used as a real-time topology controller for commercial and military applications.

Aileme:
sevgili annem Birdesen ve babam Cahit'e
ve
canım oğlum Emre, kızım Ela ve eşim Erica'ya

To My Family:
my mother Birdesen and my father Cahit
and
my son Emre, my daughter Ela, and my wife Erica

Acknowledgments

I would like to express my deep and sincere gratitude to all those people who have helped make this thesis possible.

It is difficult to overstate my gratitude to my Ph.D. supervisor, Prof. M. Ümit Uyar, for his inspiration, his guidance, and his great efforts to explain things clearly and simply throughout my doctoral studies. This work would not have been possible without his experience, his support, and his encouragement especially during the difficult moments of my research when nice results were not so visible. I would also like to thank Prof. Michael Conner for his guidance and help during my study. I am grateful to Prof. Jizhong Xiao, Prof. Nelly Fazio, Dr. İbrahim Hökelek, and Mr. Giorgio Bertoli for serving in my examination committee.

I would like to state my gratitude to my M.Sc. supervisor, Prof. F. Rüyal Ergül from the Middle East Technical University, Ankara, Turkey. It is an honor for me to thank Prof. Adnan Köksal from the Hacettepe University, Ankara, Turkey since he thought me how much fun the research is. I would like to address my thanks to Mr. Christian Pizzo from U.S. Army Communications-Electronics RD&E Center for his help. I would also like to show my gratitude to my managers Mr. Sami Pancaroğlu and Mr. Nevzat Yaraş from Mikes, Ankara, Turkey and Aroon Naidu from Elanti Systems, NJ, USA for their support and encouragement throughout my Ph.D. study.

I would like to acknowledge Prof. Uyar's support of my Ph.D study with the following grants: U.S. Army Communications-Electronics RD&E Center grant W15P7T-09-C-

S021 and the National Science Foundation grants ECS-0421159 and CNS-0619577.

I want to thank my friends and colleagues Elkin Urrea, Janusz Kusyk, Stephen Gundry, Cevher Doğan, Jumie Yuventi, and Yahao Chen of the City College of the City University of New York, and Dr. Suat Gümüşsoy from K.U. Leuven, Belgium for their valuable discussions and suggestions. Their help is very much appreciated. Dr. Larry Sheldon from BAE Systems, NJ, USA also deserves special mention for his help.

I wish to thank my grandparents Hüseyin and Hüsniye Takmaz, whom I consider my first teachers. I would also like to thank my uncles Ali Atilla and Ata Takmaz. They always believed in me.

Finally, I express my deepest gratitude and appreciation to my family, my mother Birdesen, my father Cahit, my son Emre, my daughter Ela, and my love Erica for always being there for me when I needed them, and always encouraging and supporting me in my studies. Without them, this work could not have been accomplished.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Autonomy in machines is defined as the capability of operating in the real-life environment without any centralized and external control unit. *Autonomous systems* represent a blend of software and machinery to create intelligent platforms for complex real-world problems without human control and guidance. These systems must be self-sufficient and capable of adapting their behavior to rapidly changing and most likely unfamiliar environments.

1.1 Motivation

A mobile ad hoc network (MANET) is a wireless network paradigm which encompasses autonomous systems of mobile nodes connected using wireless links with limited ranges. These mobile entities dynamically establish a network without any need of

pre-existing structure or an administrator. Each mobile node, in most circumstances, moves and operates in a distributed peer-to-peer mode, generating independent data and acting as a router to provide multi-hop communications (i.e., communication between far away mobile nodes can be established via multi-hop routing). Unlike the wired networks that typically have fixed network topology, the mobility of MANET nodes can lead to frequent and unpredictable topology changes due to mobile nodes leaving and/or joining the network without notice. The main goal of topology control in a MANET is to maintain network connectivity, optimize network lifetime and throughput, and make it possible to design power-efficient routing [1]. The optimal topology control and coordination in a MANET should be highly dynamic and adaptive due to a MANET's dynamic and unpredictable topology. A global optimal solution for the control and coordination of a MANET topology may be achieved by continuously monitoring all mobile nodes' status; however, this approach is not scalable due to the overhead required to obtain such information for a highly dynamic topology. Another approach may be the distribution of the topology control and coordination; it may be realizable and practical but the need and lack of reliable coordination may result in instability and the underutilization and waste of valuable resources such as bandwidth and energy. Furthermore, distributed topology control approach may not quickly adapt to dynamically changing environments. One way of maintaining topology control and a uniform distribution of mobile nodes over a terrain is to provide the nodes with the ability to adapt their speeds and movement directions based on their local neighbor nodes and surroundings (e.g., number of neighbors, neighbors locations, obstacles within node sensing range, etc.). In other words, each mobile

node uses a topology control algorithm as a self-distribution mechanism to decide its next speed and movement direction to obtain a uniform distribution.

Recent years have seen a considerable amount of research interest in MANETs since MANET applications span a wide spectrum ranging from commercial to military applications such as clearing mine-fields, spreading military assets in unknown areas (e.g., robots, mini-submarines etc.), controlling unmanned vehicles and transportation systems, emergency and rescue operations (e.g., hurricane, earthquake, tsunami, etc.), and environmental cleaning operations. These applications typically require a uniform distribution of autonomous mobile nodes controlled by active running software agents over an unknown geographical area [2–4]. Dynamically changing network topology, lack of centralized authority, nodes' selfishness, and unknown deployment terrain present some of the difficulties in self deployment algorithm for mobile entities in MANETs [5–7].

Evolutionary algorithms (EAs) [8] are a family of evolutionary processes which are inspired by the natural selection concept called *survival of the fittest* in biological systems. They promise outstanding adaptation, reliability, and robustness features in dynamic environments such as MANETs. EAs differ from the traditional optimization algorithms (e.g., hill-climbing [9], tabu search [10], and others) in that using a *population* of potential solutions instead of a single solution. EAs, intuitively, are proper choices for multi-optimum search problems due to their intrinsic parallel search mechanisms. Fig. 1.1 shows popular optimization methods, each of which typically has a different focus on its own suitable target problems. For problems with no

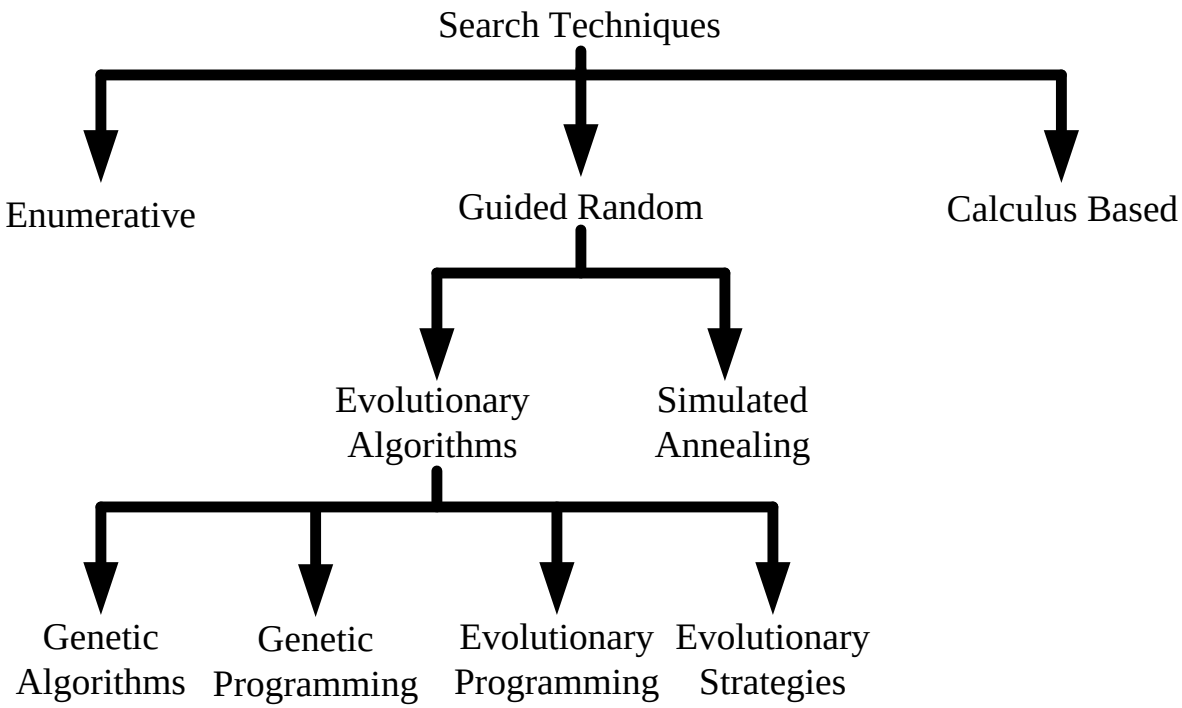


Figure 1.1: Classification of search techniques

known deterministic solutions or with solutions unfeasibly high computation complexity, there has been an increasing interest in applying evolutionary and hereditary principles based on special selection methodologies to generate approximate solutions. As shown in Fig. 1.1, genetic algorithms (called GAs) are a highly popular type of EAs using techniques motivated by evolutionary biology such as selection, mutation, and elitism [11]. GAs are suitable for solving a wide range of complex real-life problems requiring automated, adaptive, and self-learning computational techniques across military, commercial, and scientific applications. For example, decentralized intelligence is used where a large number of simple robots in multi-robot systems coordinate to obtain a collaborative behavior for completing a difficult task [12–16]. GAs are one of the most suitable approaches in finding solutions to these types of collaborative and complex task-oriented problems including NP-hard ones.

1.2 Our Approach

The techniques proposed in this dissertation provide an infrastructure for autonomous topology control in MANETs. Our coverage-centric mobile node deployment algorithm is called forced-based GA (FGA) ensures effective sensing coverage (i.e., communication range called R_{com}) for each mobile node after initial deployment. Our FGA is also an energy-aware self-organization framework since it manages the mobile nodes reducing energy consumption due to the unnecessary excessive movements. Fault-tolerant self-organization is another important feature of our GA-based topology control approach. Our FGA is resilient to losses and malfunction of mobile nodes improving the robustness of the MANET that it is operating in. Fig. 1.2 illustrates a single mobile node with its environment where mobile node 1 has five neighbors (Nodes 2, 4, 5, 6, and 7) and does not communicate with Node 3 due to an obstacle. Mobile node 1 in Fig. 1.2 runs our GA-based topology control framework to decide its next location (i.e., the next speed and movement direction) based on its neighboring nodes.

It is easy to envision many applications for GA-based topology control approach ranging from military to commercial applications, such as search and rescue missions (e.g., locating humans trapped in rubble after an earthquake), controlling unmanned vehicles, and transportation systems, clearing mine-fields, detecting hazardous materials, mapping, and spreading military assets (e.g., robots, mini-submarines, etc.) under harsh and bandwidth-limited conditions. All these applications include exploration of an unknown area that consists of identifying the locations of obstacles, objects,

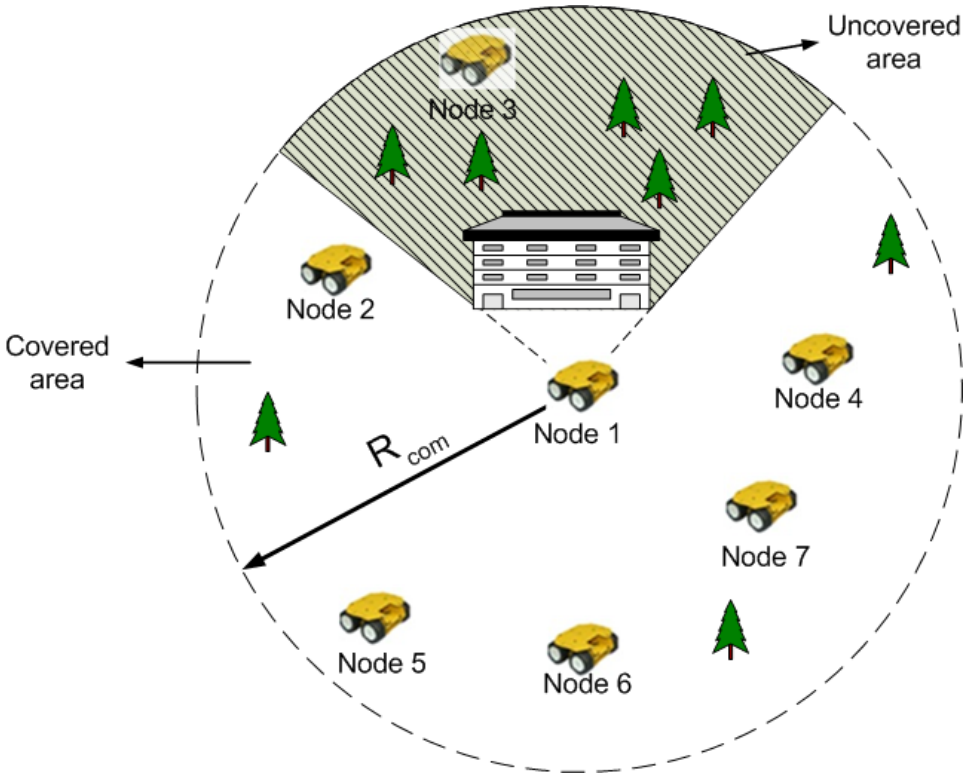


Figure 1.2: An autonomous mobile node with the communication range of R_{com}, its five neighbors, and typical obstacles in a terrain

and free spaces by sensing the environment. Exploration is a basis for many other applications. For example, to map an unknown area, mobile nodes need to explore the area. To search and rescue survivors after a disaster, mobile nodes have to explore the area to find survivors [17]. Exploration may also have several different objectives to optimize including exploration time minimization and the total energy consumption minimization. Furthermore, in these types of applications a large number of mobile nodes can self-spread and gather information from multiple viewpoints simultaneously, allowing them to share information and adapt to the environment quickly and comprehensively. One of the common objectives among these applications is the uniform distribution of autonomous mobile nodes operating on geographical areas

without a priori geographical terrain knowledge. The topology control of autonomous mobile nodes face extra challenges in MANETs since:

- due to mobility, local terrain may change dramatically and quickly in a short time-span during an operation,
- the number of mobile nodes may increase or decrease unpredictably due to malfunctions or hostile activity,
- mobile nodes may not have access to navigation maps or GPS devices, but can only have limited information collected from local neighbors,
- nodes may be deployed into a terrain from a single entry point (rather than simplified approaches using random or other types of initial distributions often seen in existing research).

GAs are adaptive heuristic search algorithms which have been demonstrated to be useful tools in a variety of search and optimization problems. GAs premise on the evolutionary ideas of natural selection and search for the best individuals within a population as the GA evolves toward the fittest solution or optimum result in an entire problem space [8, 11]. In [15, 16, 18–20], we introduce our FGA as a topology control mechanism in MANETs. In this framework, each mobile node runs FGA to decide its next speed and movement direction based on its current local information to obtain a uniform distribution. The objective function used in FGA is inspired by the equilibrium of the molecules in physics where each molecule tries to be in the balanced position and to spend minimum energy to protect its own position. Our

GA-based topology control framework uses the objective function (also called fitness function) to quantify the optimality of a solution (chromosome) and rank it against all the other chromosomes. We implemented simulation software to evaluate FGAs effectiveness and applicability to real-life problems. In addition, in the Bio-inspired Computing Laboratory at the City College of New York, we built several testbeds to study the convergence properties of various GA-based topology control mechanisms, including FGA, in MANETs. Our testbeds use different technologies and components, namely:

- FPGA Virtex-II™ with laptops and desktops,
- VMware™,
- small robots (iRobots™) controlled by gumstix™ processors,
- laptops and PDAs.

In this dissertation, the main objective for our evolutionary algorithm is to uniformly distribute mobile nodes over a two dimensional area while maintaining their communication connectivity. One of the best ways to maintain and improve network connectivity is to provide mobile nodes with the ability to adjust their speed and movement direction without a centralized control unit. However, these skills represent a challenging problem since:

- the conditions of geographical terrain may change dramatically in a short time span (although there are many factors that affect the performance and success

of mobile nodes distribution in an unknown terrain, the environmental configuration has one of the largest impact on the speed and success of mobile nodes separation),

- the number of mobile nodes may change (increase or decrease) dynamically due to malfunctions and/or loss of communications,
- mobile nodes may not have access to navigation maps or GPS devices, but have only limited information collected from local neighbors,
- mobile nodes may be deployed into a terrain from a single entry point (more realistic but more difficult than random or other types of initial distributions often seen in existing research),
- the real-world MANET applications may not have persistent and reliable network connections.

1.3 Thesis Outline

The remainder of this dissertation is organized as follows. In Chapter 2, we review prior research on the use of GAs on mobile node deployment, target localization in MANETs, swarm robotics, statistical and mathematical methods for EAs used in convergence evaluation. We also provide the differences between cited approaches and our GA-based topology control framework. Chapter 3 contains three sections, namely Genetic Algorithms (Section 3.1), MANETs (Section 3.2), and Mobility Model (Sec-

tion 3.3). In Section 3.1, we briefly introduce GAs using a simple problem as an example. Section 3.2 provides fundamentals about MANETs. Section 3.3 briefly explains our mobility model for MANETs. Chapter 4 outlines our FGA approach with formal proofs of convergence. Chapter 5 has two sections. In Section 5.1, our simulation software is introduced with experimental results. Our testbed implementations with experimental results are in Section 5.2. Chapter 6 presents the statistical analysis of our GA-based topology control approach using the results of simulation experiments for different network parameters including the node communication range (R_{com}), and the number of autonomous mobile nodes. To formally analyze the convergence speed of FGA, we build the ergodic homogeneous (in Section 7.2) and inhomogeneous (in Section 7.3) Markov chain models as explained in Chapter 7. The concluding remarks, future research directions for this work, and our refereed publications we generated during the course of this research are presented in Chapter 8.

Chapter 2

Related Work

Our FGA is inspired by the force-based distribution in physics where each molecule attempts to remain in balanced position and to spend minimum energy to protect its own position. FGA is run by each mobile node as a standalone topology control application to uniformly distribute mobile nodes in an unknown terrain [18, 19, 21]. In our research, we used discrete-time walk model adapted from [22]. We provide formal analysis of our GA-based topology control framework in [20, 23]. We analyze the convergence rate of our GA-based approach by using homogeneous Markov chain in [16, 24, 25] and inhomogeneous Markov chain in [23, 26]. In this dissertation, we refer to our models of homogeneous and inhomogeneous Markov chains as hMC_{FGA} and iMC_{FGA}, respectively. hMC_{FGA} and iMC_{FGA} models help us to explicate the convergence behavior and speed of our GA-based topology control approach with respect to different environmental conditions (e.g., number of neighbors and position of obstacles), number of mobile nodes, and communication range of R_{com}.

2.1 Topology Control Algorithms for Mobile Ad hoc Networks

There have been a number of proposals for self-organization of mobile nodes. In [6], mobile wireless sensors organize themselves in an adaptive manner over a geographical area by utilizing a self deployment algorithm based on cooperative mobile robotics. The percentage of the region covered, uniformity of network topology, time elapsed until all the nodes reach their final locations, and the average distance traveled by each node are considered as performance metrics. Another approach rely on computational geometry techniques such as Voronoi diagrams (VA) where the area of interest is partitioned into many subareas, one for each node, so that the nodes can move to maximize coverage in its own sub-area. [27] applies VAs to discover areas not covered by any node and provide general rules based on the principle of moving nodes from densely deployed areas to sparsely deployed areas. A modification of this approach [28] analyzes the problem of sensor deployment in a hybrid scenario, with both mobile and fixed sensors in the same environment. [29] uses Delaunay triangulation techniques to maximize coverage area while minimizing coverage gaps and overlaps by adjusting the deployment layout of nodes close to equilateral triangulation. [30] presents algorithms designed to minimize moving distance of nodes with particular emphasis on operative settings where coverage does not imply connectivity. In [31], evolutionary algorithms are used to calculate off-line path planning for unmanned aerial vehicles in a 3-D terrain. [32] proposes a reduced complexity GA for optimizing sensor networks to

create a maximum number of sensor clusters with cluster-heads using a GA.

Another common approach uses potential-field-based algorithms that assume that the movement of each node can be affected by virtual forces (VF) from other nodes and obstacles [33–35]. This approach imitates the behavior of electro-magnetic particles: when two electro-magnetic particles are too close to each other, a repulsive force pushes them apart. There are other approaches inspired by the physics as well. Based on fluid dynamics, [36] models the mobile node network as as a fluid body and the individual nodes as fluid elements that can penetrate and diffuse into highly unknown and unstructured terrain. In [37], the theory of gases is used to give a unified solution to the problem of deployment and dynamic relocation of mobile sensors in an open environment. Two physics-based approaches based on artificial forces and the kinetic theory of gases are used and compared to model sensor movements in the presence of obstacles in [38]. [39] introduces the concept of very large scale robotic (VLSR) systems and consider a distributed control approach based on artificial force laws between individual robots and robot groups. Another work on distributed formation control of robots is presented in [40], where the authors consider asynchronous distributed control and geometric pattern formation of multiple anonymous robots (the robots are anonymous in the sense that they all execute the same algorithm and they cannot be distinguished by their appearances).

Although the cited approaches assume unknown environments, all of them require the manual tuning of constants and thresholds whose appropriate values are closely dependent on the particular operative scenario. Our topology control algorithm, FGA,

uses a genetic algorithm in the decision-making process of adaptive and autonomic systems to dynamically evolve reconfiguration strategies that balance competing objectives at run time. Our GA-based topology control framework uses only locally available information and does not require manual tuning of key parameters or a priori knowledge of the operative scenario. Furthermore, most of the cited proposals may utilize algorithms that do not converge (most lack stability or convergence proofs).

2.2 Mobility Model

In [41], different mobility models (e.g., random waypoint, random way group mobility, manhattan mobility, and freeway mobility) have been studied for mobile node movements. [42] proposes a technique, based on renewal theory, for analyzing mobility models in ad hoc networks and apply their method to random waypoint mobility. [43] provides relation between various mobility models and their effects on performance of mobile ad hoc network. [44] proposes a simplified random walk model capturing the movement of mobile users in Personal Communications Services (PCS) networks. In this research, areas are covered by radio base stations whose radio coverage is called a cell, where they configure cells as hexagonal or mesh networks. In [44] only mobile station can move in one unit time, and link availability from a fixed base station to this mobile station is calculated. In [45], new analytic models are used to study the node link stability for realistic wireless mobile network applications by calculating

the exact link failure and creation probabilities. The mobility model used in this dissertation is adapted from [22, 45].

2.3 Statistical Analysis

In [46], the classes of symmetric functions (skew-normal, uniform, t, Cauchy, Laplace, and logistic distributions) depending on a parameter have been studied and some of their properties are provided. If data fitting is considered, there is a useful four-parameter probability distribution: location, scale, symmetry, and tailweight. [47] suggests using the method of moments for finding these four parameter values. In [48], detailed information about the skew-normal can be found. When skewness is absent, there is a singularity of the Fisher information matrix. [49] shows an alternative parameterization method that overcomes problems related to this singularity. [50] is the first publication that introduced skew-normal distribution. It provides a class of density function depending on the shape parameter (when the shape parameter is zero, the density function is the normal). The properties of this class of densities as a mean of the shape parameter is given in [50]. [51] posses important questions about statistical models and their parameters. [52] examines how well the different approximation methods capture the tail behavior of a function of random variables and provides a detailed explanations of the tail behaviors. Prediction and model identification of the stationary time series problems are discussed in [53]. It also provides analysis on providing realistic models for the processes generating observed

time series. [54] introduces data distortion by probability distortion in three steps. It focuses on the idea that the probability distorted series has asymptotically the same statistical behaviors of the original series. All of the cited references above provide more detail and practical information about parameterization and analysis of random variables using probability distribution functions.

2.4 Convergence Analysis

Schema theory [11] and *Markov chain* are widely used to provide a formal structure for analyzing GAs. For example, a modified elitist strategy is used to generate the current population from the reserved highest fitness valued individual and the rest are from the previous generation in [55]. Similarly, [56] extends the previous work of [55] and finds that the convergence rate of a GA is determined by the second largest eigenvalue of the transition matrix. In another study in [57], GA convergence is modeled by using Markov chain showing that GAs applied to large scale problems should avoid convergence towards an unwanted solution or a local optima. The local convergence of a GA with Cauchy mutation operator is analyzed in [58]. In [59], the eigenvalues of the transition state matrix from Markovian analysis are used to estimate the convergence rate. In [60], GA convergence time behavior is modeled with a Markov chain to show the effects of binary and higher cardinality representation of a search space. One of the first models for a simple GA as a Markov chain is given in [61]. In this model, selection, mutation, and crossover are incorporated with the transition matrix. [62]

investigates the convergence properties of GAs applied to fitness functions perturbed by multiple sources of additive noise that each take a finite number of values. [63] proposes an algorithm for the control of autonomous swarms using the Gibb's sampler simulated annealing process. [64] discusses modeling of MANETs using Markovian Model representation. [65] discusses fundamental properties of finite Markov chains, including graph theoretic considerations for transient and non-transient, recurrent and non-recurrent cases. [66] considers the convergence of a GA in which only the crossover operator is used with an elitist selection scheme. The local convergence of a GA with Cauchy mutation operator is analyzed in [67]. The convergence of a GA for a set of convex objective functions is discussed in [68]. [69] discusses the convergence analysis of canonical genetic algorithms. [70] analyzes GAs that make use of a property called niching where multiple high fitness peaks are preserved by a *niche operator*. [71] proposes a hybrid GA that incorporates the process of simulated annealing, named the *Global annealing genetic algorithm*. [72] discusses Markov chain analysis of a GA utilizing various selection schemes including maximum similar mating, proportional similar mating, proportional dissimilar matting and maximum dissimilar mating. [73] analyzes the evolution of GAs in noisy environments and proposes bounds on fitness functions that incorporate noise conditions. [74] explains various Markov chain Monte Carlo methods, including the Gibb's sampler and the Metropolis sampler and also discusses various metrics used to characterize digital images and lays out the key problems in analyzing noisy images. It then provides a basis for describing digital images in terms of random fields. Further [74] provides fundamental convergence analysis techniques laid out by R. Droburshin [75]. This analysis is further characterized using

Markov chain analysis, including the method for constructing the transition matrix of the Markov chain model. It shows that the the GA will converge to is optimum state if the noise does not bring the fitness of the highest state lower to any suboptimal state.

In this dissertation, we use Droburshin's contraction coefficients to analyze the convergence characteristics of hMC_{FGA} and iMC_{FGA} models under different circumstances in terms of different communication range of R_{com}, different environmental conditions, and different number of nodes. The results do not depend on the initial condition of mobile nodes (e.g., initial location, speed, and movement direction). Furthermore, different from the cited approaches, our hMC_{FGA} and iMC_{FGA} models shows the convergence characteristics as a system (i.e., all mobile node in a MANET) rather than individual algorithm (i.e., a mobile node).

2.5 Distributed Mobile Robotic Systems

Since the beginning of 2000's, the emphasis of robotics research has shifted from the control of individual robot to distributed robotics, swarm robotics, and mobile sensor networks. Instead of an individual robot, the vision is changed to a large number of well-coordinated miniature robots which can gather sensory information from multiple viewpoints simultaneously. This approach allow multi-robot systems to understand the environment more quickly and comprehensively. Furthermore, miniature robots have many advantages such as:

- much less expensive than a large individual robot,
- easily operate in difficult to-access areas,
- easily hide to avoid detection (esp. in military missions),
- low-power,
- easy to build a large scale team for complex tasks.

Bio-inspired approaches, especially GA, has also proven as an efficient approach in various distributed robotic applications.

In [76], a GA-based approach is used to satisfy a distance-safety criteria for a mobile robot motion. [77] proposes an algorithm to guide autonomous robots in a highway to reach to their destination without any collision. In [78], an adaptive GA is used to identify targets while avoiding obstacles. The mobile robots collect information from the environment with their video camera and light sensors and run their own GA to stay away from static and unknown blockages and arrive to a given target. [79] describes how a parallel computing GA can be applied for allocating target points to multiple mobile nodes, such as robots, appropriately so that the overall area exploration time is minimized. [80] proposes a GA for path planning of a mobile robot, which incorporates the domain knowledge into its specialized operators and is capable of finding an optimal or near-optimal robot path in both complex static and dynamic environments. [81] discusses GAs to explore the space of path finding algorithms in training and three dimensional naval real-time strategy games. Autonomous

dispersion of mobile nodes using a random diffusion method are considered in [82]. A potential-field approach is used to deploy mobile sensors in [83]. The fields are constructed such that each sensor is repelled by both obstacles and by other sensors, thereby forcing the network to spread itself throughout the environment.

Swarm robotics is another approach of using multi-robot systems collaboratively instead of single complex robot. Several promising results in swarm robotics are recently reported. Multi-agent collaboration for swarm robots for distributed missions to fetch and retrieve objects is presented in [84]. In [85], cooperative exploration strategy for mobile robots are presented based on the sensor-based random trees. [86] provides a basis for a distributed robotic system capable of constructing any given planar structure. [87] discuss various applications and a prototype for different scenarios including self powered and self-governing swarm robotic platforms. Abstract models for rescue operations using swarm robots are studied in [88]. In [89], swarm optimization is used for route planning for unmanned aerial vehicles using a fitness function based on both flight time and safety. Similarly, a particle swarm optimization algorithm for path optimization of soccer playing robots is proposed in [90]. [91] illustrates a complex transporting problem requiring collaboration for small robots. Quantum probability in the chromosome coding strategy to adapt coalition formation into multi-robot systems is used in [92]. A hierarchical behavior-based model in which several parameters are adjusted with a GA for tuning the parameters of a swarm to surround a target is proposed in [93]. [94] outlines the interactive use of autonomous robots and human beings in fire emergency settings. This study shows that a swarm of robots which are

capable of working in fire fighting operations. Dependability, robustness, and reliability of the swarm-based systems for distributed safety critical systems are discussed in [95]. An algorithm for distributing a swarm of primitive robots in an unknown geographical area is proposed in [96].

There are fundamental differences between our approach and the existing research cited above. In our bio-inspired algorithm, there is no difference in mobile node's privilege (i.e., no leader or follower). FGA only utilizes information from neighboring nodes and local terrain to make movement and speed decisions to converge towards a uniform distribution of mobile nodes; there is no central controller unit or global knowledge of the entire network. In fact, as will be shown in Chapters 5 and 7, FGA adapts to its immediate environment rapidly and does not require global network knowledge, and hence it can be used as a real-time topology controller for realistic military and civilian applications. Our GA-based topology control approach has the basic self-* properties of autonomic computing, i.e. self-healing, self-configuration, and self-adaptation, giving rise to a fully decentralized algorithm. Furthermore, the decentralized characteristic of FGA makes it resilient to mobile node losses. Another significant difference is that no a priori knowledge of the geographical area is needed for FGA. It is also important to note that the self-* is more challenging in MANETs compared to sensor networks since, unlike sensor networks, there are no stationary nodes are present in MANETs.

Chapter 3

Genetic Algorithms and Mobility Model

In this chapter, we briefly introduce GAs with examples from a traditional GA. We also discuss our mobility model used in our analysis.

3.1 Genetic Algorithms

Genetic algorithms (GAs) are a class of stochastic search algorithms forming a subset of bio-inspired computation algorithms. Biologically inspired computation algorithms are a subset of evolutionary algorithms (EAs). They mimic nature such that the hereditary transfer of biological trait information is used as a role model for stepwise improvement and development of a population of candidate solutions in EAs. GAs were

developed by Holland [11] in 1975 as a tool to find solutions to optimization problems in poorly understood large spaces. GAs are based on the *survival of the fittest* principal utilizing the *genetic process of a biological organism*. In other words, GAs mimic the way biological trait information is transferred and improved under selection pressure in nature. The desired phenotype traits (that is, those of individuals) are selected by the evaluation of a specified *fitness* (i.e., objective) function. *Individuals* with a

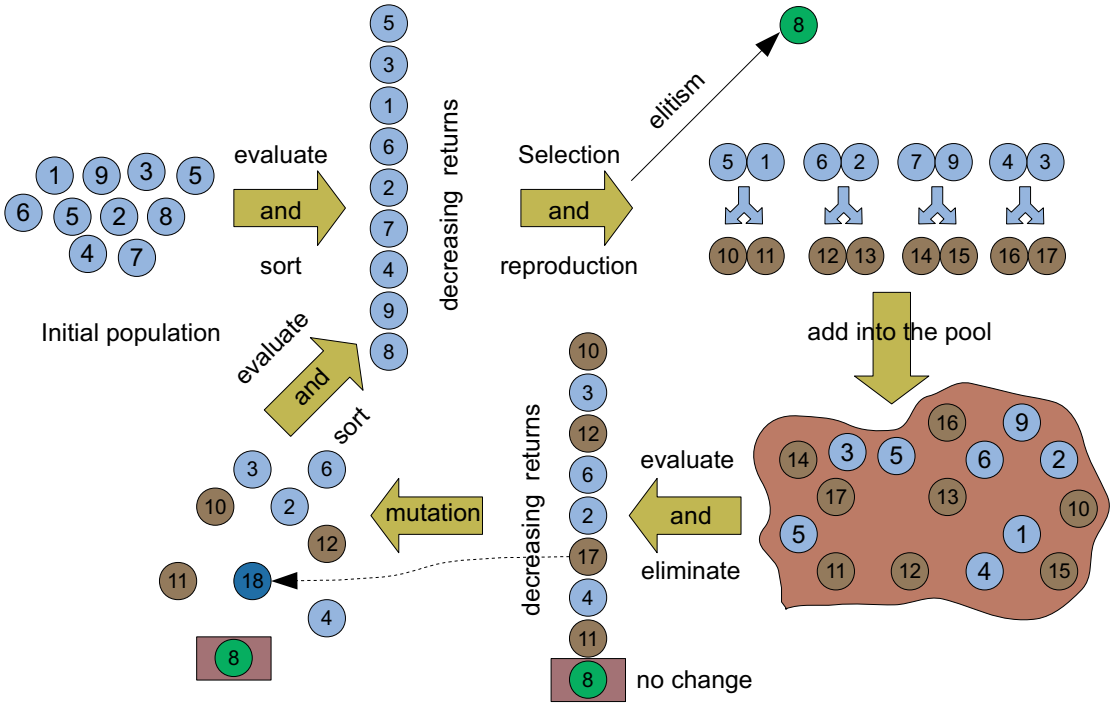


Figure 3.1: An example of operational flow in a traditional genetic algorithm

higher objective function score are more likely to be selected for breeding process in a GA. According to the theory of evolution, only those individuals in a population who are better suited to the environment are likely to survive and generate offspring, thereby transmitting their superior genetic information to new generations [8, 11].

The GAs, utilizing this principal of survival of the fittest, are typically applied to problems where deterministic methods are not present or cannot provide satisfactory results. The GAs are essentially composed of a set of individual *chromosomes* (called a *population*) and biologically inspired operators that create a new (and potentially better) population from an old one in a generation.

GAs proved to be quite successful in finding good solutions to such complex problems as the traveling salesman problem, large scheduling problems, the knapsack problem, graph partitioning, but also engineering and science problems like aircraft design, production line optimizations, DNA sequence prediction, robotic applications, and financial market predictions.

3.1.1 The Structure of Genetic Algorithms

Typically, a GA works on a population of binary strings which correspond to the chromosomes in biological systems. Algorithm 3.1 presents pseudo-code for a traditional GA where all chromosome have the same length. In a traditional GA, if there is not enough information about problem space, the initial population of $P(0)$ is randomly generated with n individuals. As a part of the goal of Algorithm 3.1, to optimize the fitness function, at each generation, the fitness value of all chromosomes in a given population $P(g)$ is computed. The transition from the population $P(g)$ at generation g to $P(g+1)$ is performed by several bio-inspired genetic operators, namely crossover, mutation, and evaluation. These operators are briefly described in

Section 3.1.2. This process continues until satisfying a certain stop criteria such as reaching a certain number of generations.

Algorithm 3.1 Pseudocode of a traditional genetic algorithm

```
g ← 0 [generation counter]
Initialize population P(g) [initial population P(0)]
Evaluate population P(g) [compute fitness value of P(0)]
while not done do
  g ← g + 1
  Select P(g) from P(g-1)
  Crossover P(g)
  Mutate P(g)
  Evaluate P(g)
end while
```

To gain more insight about how traditional GA works, let us consider the steps in a typical GA as shown in Fig. 3.1. First a population of n individuals is randomly generated and evaluated using a fitness function score. For example, in Fig. 3.1, numbers from 1 to 18 represent chromosome ID (designated IDs are only for exemplifying). The population is then sorted based on the individuals fitness values. Let us assume that the typical GA is a minimization problem and, hence, the population is sorted in a decreasing order of the fitness scores in Fig. 3.1. In crossover, individuals are selected for breeding with probabilities proportional to their fitness scores (this selection method is called a *roulette wheel* selection [8, 11]). For example, in Fig. 3.1 individuals 5 and 1 produce offspring 10 and 11. The offspring are then added to a pool consisting of candidate solutions for a new population. The offspring in the pool are then evaluated based on their fitnesses. Only the better performing individuals are accepted into the newly created population. Mutation occurs on randomly selected individuals. Mutation protects against GAs staying at local optimum points. In

Fig. 3.1, individual 17 is the only one that has been selected for mutation. In the new population, it is mutated to individual 18 (indicated by an arrow in Fig. 3.1). The elite individual has the best fitness value in the previous population and is typically chosen for the newly created population without applying any genetic operators (i.e., elite individual is a member of the new population without being changed/updated by any genetic operator). Therefore, the best fitness score in the new population is better than, or at least the same as, the previous one. For example, in Fig. 3.1 individual 8 has the highest fitness score and, therefore, is placed into the new population without being exposed to any genetic operators. The population undergoes this process (without the initialization step where the chromosomes are chosen randomly) for many generations until a termination criterion is satisfied (e.g., convergence tolerance of the best individuals reaching a certain preset limit, satisfying a predefined fitness value, or reaching a limit on the number of generations).

3.1.2 Genetic Operators

3.1.2.1 Chromosome

In a GA framework, a possible solution for a given problem is represented by a chromosome that should be constructed carefully. Ideally a chromosome should include all important parameters of the optimization problem space and be simple enough not to increase the computational time to infeasible proportions. In general, suppose a binary chromosome is defined in a discrete search space Ω. A GA minimizes

a given fitness function f_i which maps from the search space to fitness landscape as $f_i : \Omega \rightarrow \Psi$. Then, each potential solution in the search space can be encoded into l-bit binary chromosome that is an element of $\Phi = \{0, 1\}$. This is a one-to-one mapping from the search space to the chromosome space as $\Theta : \Omega \rightarrow \Phi$.

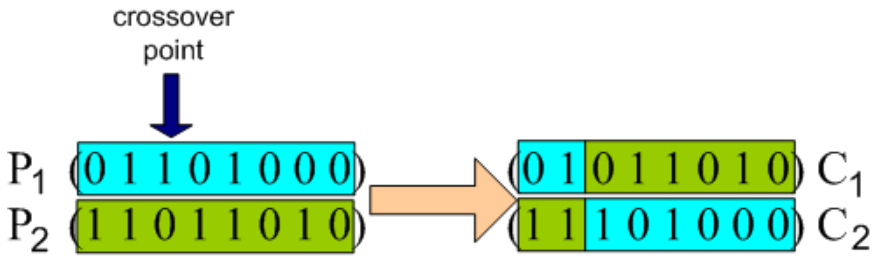


Figure 3.2: An example of 1-point crossover with crossover point two

3.1.2.2 Selection

The intent of the selection operator is to imitate the process of survival of the fittest. It determines which individuals in the current generation is allowed to inherit their genetic material to the next generation. Roulette wheel selection (also called proportional selection) and tournament selection are two of the most common selection methodologies. In roulette wheel selection, the probability that a given member j in the current population is chosen is proportional to the fitness of individual j [97]. For example, if there is a chromosome j with fitness value is twice as high as the fitness of some other chromosome i, the chromosome j will expectedly have twice as many copies of chromosome i in the next generation. Tournament selection involves running several tournaments among fitter individuals chosen at random from the current population. The winner of each tournament (i.e., the individual with the highest

fitness) is selected for crossover.

3.1.2.3 Crossover

The crossover operator is used to generate new individuals from the old ones. The intend is to create offspring from the individuals with high fitness to produce even fitter individuals [97]. The most common crossover method is called *1-point crossover*. In 1-point crossover, a point is randomly chosen between 1 to l where l is the length of the chromosome, then, the values of the bits are swapped as shown in Fig. 3.2. Suppose we have two chromosomes of $P_1 =< 01101000 >$ and $P_2 =< 11011010 >$ and the crossover point is 2. As seen in Fig. 3.2, after crossover operator applied to P_1 and P_2, the offspring of $C_1 =< 01011010 >$ and $C_2 =< 11101000 >$ are generated by bit swapping.

3.1.2.4 Mutation

The mutation operator lets GAs to introduce new genetic information into a population. It also protects GAs from stacking at a local optimum point. The most common mutation operation is 1-bit mutation at which randomly selected one bit in a chromosome is flipped from 1 to 0, or vice versa. For example, as shown in Fig. 3.3, a chromosome $< 11101000 >$ is replaced by its mutation of $< 11100000 >$ in a next generation (i.e., the fifth bit in the chromosome is changed from 1 to 0).

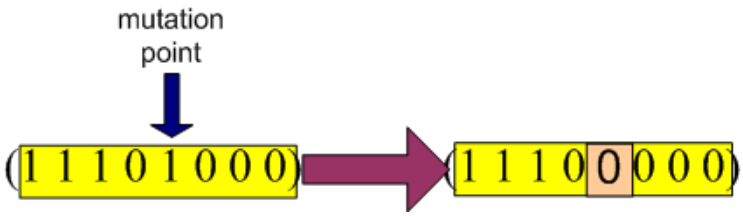


Figure 3.3: An example of 1-bit mutation with mutation point five

3.1.2.5 An Example for Traditional GA

To show the working principal of a traditional GA given in Fig. 3.1 and Algorithm 3.1 let us explicitly calculate one iteration of the function given below (shown by Fig. 3.4):

$$f(x) = \begin{cases} 1 & 0 < x \leq 6 \\ 3 & 6 < x \leq 10 \\ 7 & 10 < x \leq 14 \\ 5 & 14 < x \leq 20 \\ 8 & 20 < x \leq 27 \\ 10 & 28 < x \leq 32 \end{cases}$$

Suppose a population is composed of four binary strings of length of 5. Each chromosome encodes an integer between 1 and 32. We use roulette wheel selection algorithm (see Section 3.1.2.2) with one-point crossover with a probability of $\mu_c = 1.0$ (see Section 3.1.2.3) , and a mutation rate of $\mu_m = 0.001$ (see Section 3.1.2.4).

For this example, the initial population is $P_0 = \{00001, 01010, 11001, 11000\}$. The fitness values of the chromosomes in the initial population P_0 can be calculated as $f_{(00001)} = 1$, $f_{(01010)} = 7$, $f_{(11001)} = 8$, and $f_{(11000)} = 8$. The probabilities to be

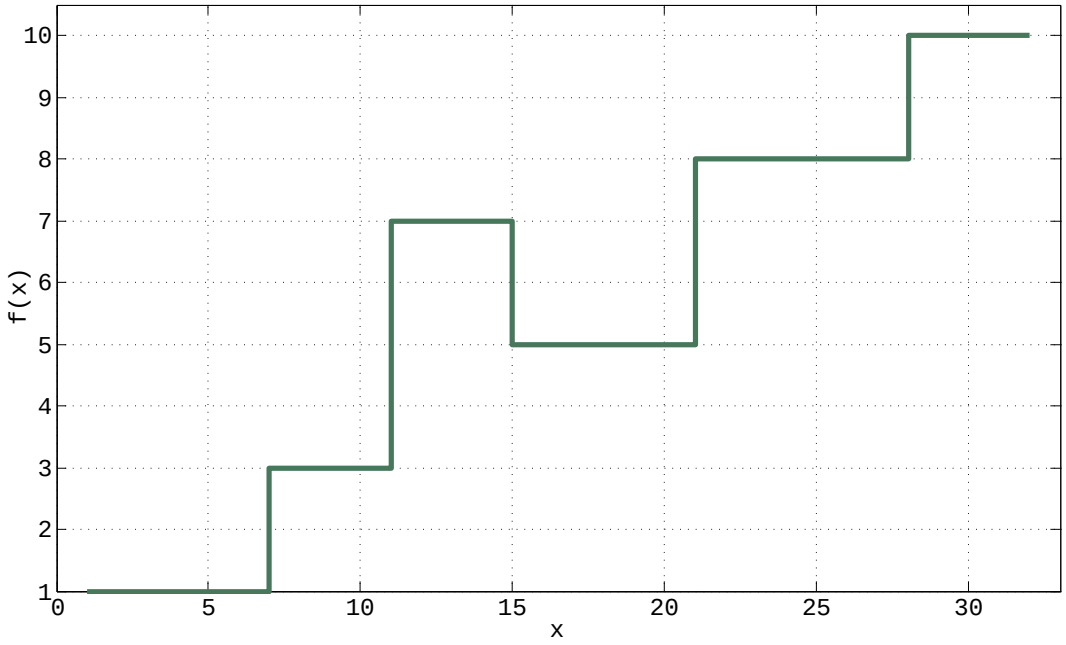


Figure 3.4: The fitness function f(x) to be maximized by the traditional genetic algorithm

selected in an arbitrary draw of roulette wheel selection (see Section 3.1.2.2) for each chromosome in the initial population are $p_{(00001)} = \frac{1}{24}$, $p_{(01010)} = \frac{7}{24}$, $p_{(11001)} = \frac{8}{24}$, and $p_{(11000)} = \frac{8}{24}$. To create the mating pool, we need 4 offspring which may include $\frac{1}{6}$ copies of $p_{(00001)}$, $\frac{28}{24}$ copies of $p_{(01010)}$, and $\frac{32}{24}$ copies of $p_{(11001)}$ and $p_{(11000)}$. Then let us assume that the mating pool is obtained as $\{11010, 11010, 01001, 01001\}$ after the crossover operation is applied. Suppose that offspring $P_{(11010)}$ in the mating pool is mutated to $P_{(11110)}$. The fitness values of the offspring are $f_{(11110)} = 10$, $f_{(11010)} = 8$, $f_{(01001)} = 3$, and $f_{(11001)} = 3$. As a result, the new population will be composed of $P_1 = \{111110, 11010, 11001, 11000\}$.

3.2 Mobile Ad hoc Networks

Wireless ad hoc networks have inspired tremendous research interest in since the beginning of 2000's. Recent achievements in wireless communication technologies, microeletromechnanical systems (called MEMs), and electronic technologies have enabled the development of efficient, low-cost, low power, and multi-functional mobile nodes that can sense the environment, communicate with local neighboring nodes, and perform data processing to make certain decisions such as next movement direction, speed etc. A mobile ad hoc network (MANET), which is a subset of wireless ad hoc networks, consists of an autonomous system of mobile nodes which dynamically form a network without any pre-existing structure. A typical MANET may consist of thousands of mobile nodes, deployed either randomly or according to some predefined statistical distribution, over a geographic region of interest. A mobile node by itself has severe resource constraints,such as low battery power, limited signal processing, limited computation and communication capabilities, and a small amount of memory; hence it can sense only a limited portion of the environment. However, when a group of mobile nodes collaborate with each other, they can accomplish a much bigger task efficiently. The primary advantages of deploying a MANET include:

- low deployment cost,
- freedom from requiring a wired communication backbone,
- low interference because of mobile nodes' limited communication ranges,

- resilience to mobile node losses,
- robustness in the sense of non-hierarchical distribution and autonomous management.

Today's technology has given many advantages for MANET nodes in terms of performance, flexibility, robustness, mobility, and functionality for sensor-oriented tasks such as environment monitoring, fire detection, transportation systems, and other complex tasks. MANET nodes has stimulated a wide range of applications in even hostile environments for military missions. An autonomous mobile node in a MANET is generally equipped with sensing, computation, communication, and locomotion capabilities. The locomotion capability is the key distinguisher for MANET nodes since it is the particular difference between a MANET node and a static network. Movement capability facilities a number of useful network features, including (but not limited to) the ability to self-deploy. Self-deployment is one of the key features in our approach since it provides capability to start from a compact initial configuration, then, the nodes in a MANET can propagate by spreading out throughout time.

These mobile entities are geographically dispersed and equipped with wireless transmitters and receivers to communicate with each other within a MANET. The communications among the mobile nodes are generally established through multi-hop routing due to the limited range of transmission capabilities of each individual node. Since the mobile nodes move arbitrarily in a MANET, the network topology may change dynamically and unpredictably. One way of maintaining a uniform distribution of

mobile nodes over any terrain is to provide the nodes with the ability to adapt their speeds and movement directions based on their local neighbor nodes and surroundings (e.g., number of neighbors, neighbors locations, obstacles within node sensing range, etc.). To handle frequent topology changes that may happen due to the locomotion of mobile nodes and obtain/maintain uniform separation of nodes, we use a bio-inspired topology control approach as explained in Section 4.

3.3 Mobility Model

Mobility is an inherent character of MANETs. In fact, to study uniform distribution of autonomous mobile nodes in a MANET, it is important to have an accurate mobility model of mobile nodes for simulation and evaluation purposes. In other words, a mobility model attempts to mimic the movements of real mobile nodes. In general, mobility models are categorized as synthetic models and actual models [43]. Synthetic mobility models for ad hoc networks are divided into two subgroups; random models (e.g., the random walk mobility model called Brownian Motion, the gauss-markov mobility model, and the column mobility model) and controlled models [43]. Actual mobility models are extracted from real-life mobile nodes. Different mobility models have different focuses and are applicable to different applications. For a uniform distribution in an unknown geographical terrain, a controlled mobility model is suitable since self-organizing autonomous mobile nodes do not have a global administrator or global knowledge, and each mobile node decides its own movement, direction, and

speed [98]. Therefore, we design our controlled mobility model such that it not only aims at describing an individual node's motion behavior, but also considers the collective motion of all the mobiles nodes over time. In our approach, adopted from the mobility model introduced by [22,45], each autonomous mobile node runs FGA as an independent software agent such that our GA-based topology control framework assigns the speed and movement direction of a mobile node based on its local neighborhood information (see Section 4). Our mobility model reflects the behavior of an individual mobile node with respect to its neighboring nodes and surroundings (e.g., obstacles, area borders, etc.). To reflect the autonomous nature of individual nodes, in our model, there is no notion of collective synchronized movement by all of the mobile nodes with reference to any particular point.

In our mobility model, we have a hexagonal terrain as a two-dimensional geographical area of (d_{max} x d_{max}) composed of logical hexagonal cells, where a unique Cartesian coordinate pair $< x, y >$ is assigned to each one of the cells. Fig. 3.5 shows an example terrain with seven mobile nodes. The area is partitioned into logical hexagonal cells. Each mobile node can move into six different directions (i.e., D_0 through D_5), if the mobile node is not on the boundary, from a neighboring cell within one time unit. The direction and speed of movement for all of the mobile nodes is determined by the FGA running on each node.

A wireless link between two mobile nodes is represented by a vector whose dimensions are in terms of layers. We assume that the wireless link between two mobile nodes is a two-way and undirected link. There is a link between two nodes if a signal

transmitted from one node is received at the other node above a minimum required power threshold [99]. One layer is equal to the center-to-center distance between two neighboring cells. Hexagonal cells are grouped into layers such that cell $< 0, 0 >$ is at layer 0, the six cells neighboring cell $< 0, 1 >$, $< 1, 0 >$, $< 1, -1 >$, $< 0, -1 >$, $< -1, 0 >$, and $< -1, 1 >$, are at layer 1. In general, for a mobile node in location $< 0, 0 >$ and another mobile node in location $< x, y >$, the wireless link state between these mobile nodes is $< x-0, y-0 >=< x, y >$. For example, in Fig. 3.5, for a mobile node N_1 in location $< 2, 0 >$ and another node N_4 in location $< -1, 2 >$, the vector representing wireless link between these nodes is $< -1-2, 2-0 >=< -3, 2 >$. Note that using the state reduction technique presented in [100], $< x, y >$ has an equivalent state $< x_a, y_a >$ such that x_a and y_a are non-negative integers and the link state between two mobile nodes are called an *available* if and only if $x_a + y_a \leq R_{com}$; otherwise it is called an *unavailable*.

Let d be the center-to-center distance between two neighboring cells, while x_a and y_a are non-negative integers that are equivalent to the state of $< x, y >$ given in previous paragraph. Two nodes are communicating with each other if and only if $d \leq R_{com}$, where $d = x_a + y_a$ and R_{com} is a positive integer representing the communication range of a corresponding mobile node. For a wireless link $< x, y >$ connecting two mobile nodes, after one time unit, each mobile node moves one of its six neighbor cells with a probability of $\frac{1}{6}$ for each direction (i.e., D_0 through D_5). There are 36 possible next link states, and, as some of the will result in the same vector, only 19 possible combinations as shown in Table 3.3 [100]. If R_{com} is a positive integer representing

the communication range of a node, and R is the center-to-center distance between two neighboring cells, a wireless link can be available only if $R \leq R_{com}$ (The wireless link is unavailable if there is an obstacle between two mobile nodes).

For simplicity, we accept all mobile nodes have the same communication range in this paper (however, it is easy to generalize the solution to varying communication ranges).

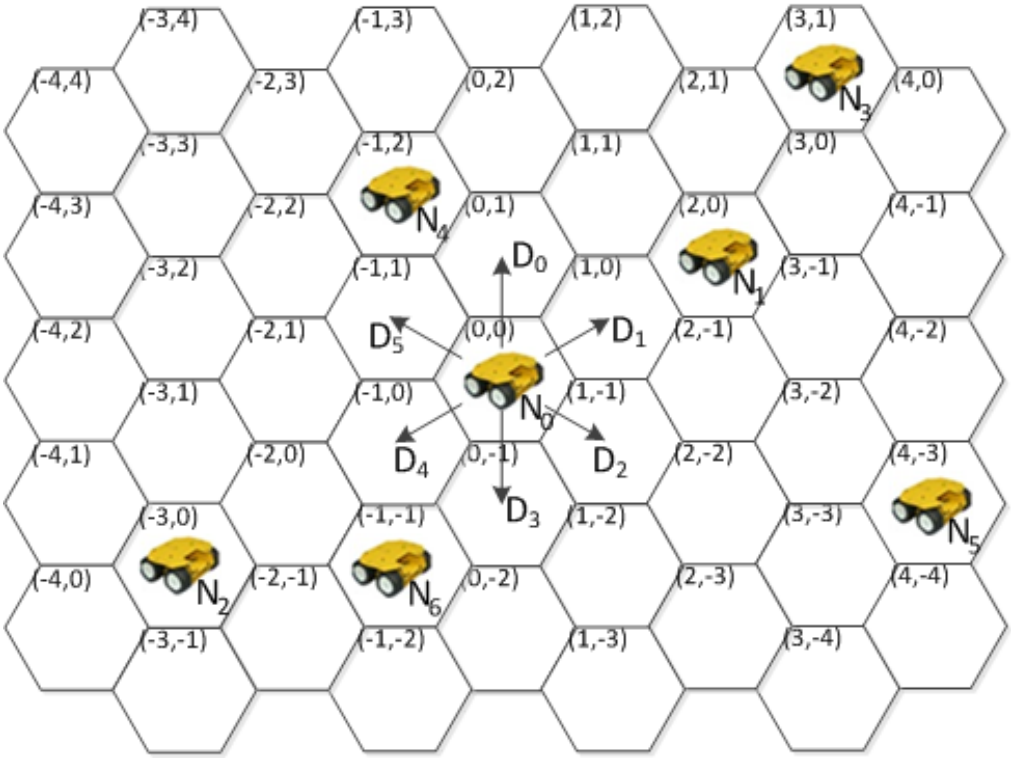


Figure 3.5: Seven mobile nodes distributed within an 5x9 hexagonal area partitioned into logical hexagonal cells ($R_{com} = 3$)

x',y'	x,y	x-1,y	x-1,y-1	x,y-2	x+1,y-2	x+1,y-1	x+1,y	x,y-1	x+2,y-2	x+2,y-1
Probability	6/36	2/36	2/36	1/36	2/36	2/36	2/36	2/36	1/36	2/36

x',y'	x+1,y+1	x,y+1	x+1,y	x,y+2	x-1,y+2	x-1,y+1	x-1,y+2	x-2,y+1	x-2,y
Probability	2/36	2/36	1/36	1/36	2/36	2/36	1/36	2/36	1/36

Table 3.1: The probability distribution for a wireless link to switch from state $< x, y >$ to state $< x', y' >$

The number of available links of a node is called its degree. In Fig. 3.5, N_0 com-

N	60	80	100	120	150	60	80	100	120	150
R_{com}	10	10	10	10	10	12	12	12	12	12
$\overline{N}$	2.59	3.48	4.36	5.24	6.56	3.68	4.93	6.18	7.42	9.30

Table 3.2: Numerical results of $\overline{N}$ for different sets of N and R_{com} (100x100)

municates with N_1, N_2, N_4, and N_6 if $R_{com} = 3$; hence, the degree of N_3 is 4 for R_{com}= 3. After one time unit each mobile node moves into one of its six directions with a certain speed. Speed and direction information are assigned by FGA based on local neighborhood information. The mobile node stays as immobile if our GA-base topology control framework assigns the speed of a mobile node as zero. In our mobility model, without loss of generality, a mobile node is not allowed to move beyond the area boundaries. For example, as shown in Fig. 3.5, N_5 can only move directions D_0, D_3, D_4. and D_5.

3.3.1 Mean Node Degree

The mean node degree (i.e., the desired number of neighbors) is the expected number of node degree to obtain the highest the coverage. The presentation of analytic formulation to obtain the mean node degree of a node located at the central cell if all mobile nodes move according to the discretetime random walk is shown in [100]. As an approximation of the optimum number of neighbors yielding the uniform node distribution, the fitness function of our GA-based topology control framework is constructed around this mean node degree [22, 100]. For example, Table 3.3 shows some numerical results for the calculated mean node degree (shown as $\overline{N}$) in the area of 100x100.

Chapter 4

Force-based Genetic Algorithm

We address a challenging problem which is typical in many civilian and military applications defined as follows:

> *Given N mobile nodes with communication ranges of R_{com}, how should they autonomously deploy themselves in an unknown region of interest (A_{RoI}) using only local neighborhood information so that resulting configuration maximizes the area coverage of the network while reducing the battery usage and staying as a fully-connected network?*

This problem becomes even more challenging due to typical characteristic of MANETs:

- the deployment cannot be determined *a priori* when A_{RoI} is unknown or hostile (for military missions)
- the geographical area may change dramatically in a short time-span during an

operation

- the number of autonomous mobile nodes in a MANET may change (increase or decrease) dynamically
- mobile nodes do not have access to navigation maps nor to GPS devices but only have limited information from local neighbors
- mobile nodes are typically deployed into the terrain from a single entry point (more difficult to analyze than random or other types of initial distributions often seen in existing research).

Since there is no deterministic solution for this problem, we will use our GA-based topology control framework to find acceptable solutions as explained in this dissertation.

We introduced a force-based genetic algorithm, called FGA [21, 101], inspired by the molecular repulsive force-based distribution in physics [6]. FGA is run by each mobile node as a stand alone software agent. A virtual force is assumed to be applied by the neighboring nodes to a corresponding node. At the equilibrium, the aggregate virtual force applied to a node by its neighbors should sum to zero. If the virtual force is not zero, our GA-based agent uses this non-zero virtual force value in its fitness calculation to adjust the nodes speed and direction of movement such that the total virtual force on the mobile node will be minimized. The value of this virtual force depends on the number of neighboring nodes within its communication range of R_{com} and the distance among them. In FGA, a smaller fitness value indicates a better position for

the corresponding node.

In FGA framework, each node maintains a *neighborhood table* to keep records for its neighboring mobile nodes. Every ΔT time units, a mobile node N_i runs its GA-based software agent to find a better location to move (if exists) based on the information from its neighborhood table; if it cannot find a location to improve its fitness, the node stops moving momentarily (the details of FGA fitness function are presented in Section 4.2).

Algorithm 4.1 presents the pseudo code of our FGA. First the neighborhood table is updated by the information received from the nodes in its communication range. Then a population of N individuals, called the initial population, is randomly generated where each individual represents a speed and movement direction for the node. Each individual (i.e., chromosome) is then evaluated using a fitness score and sorted based on its fitness value. Since our FGA is posed as a minimization problem, individuals are sorted in a decreasing order of fitness scores, representing virtual forces applied to them. In selection and crossover operators in Algorithm 4.1, individuals are paired for breeding purposes. The mating probability is proportional to their fitness scores (see Section 3.1.2.2). At generaton g, the offspring are added to a pool as candidate solutions for a new population $P(g)$ based on the old one $P(g-1)$. After the offspring in the pool are evaluated, only the better performing individuals are accepted into the newly created population of $P(g)$. Mutation occurs on randomly selected individuals of a new population to protect the populations against local optimum points. The population evolves using this process for many generations until

Algorithm 4.1 Pseudo-code of our FGA

```
while ! (stopCrit) do
  for all all neighbors do
    if Neighbor is in the neighborhood table then
      Update the neighbor's information
    else
      Add the neighbor's information into table
    end if
  end for
  g ← 0 (generation counter)
  Initialize population P(0)
  Evaluate population P(0)
  while ! (evolveDone) do
    g ← g + 1
    Select P(g) from P(g-1)
    Crossover P(g)
    Mutate P(g)
    Evaluate P(g)
    if localPositionFound then
      evolveDone := true
    end if
  end while
  if betterLocationFound then
    Move to new location
  else
    Wait (Δ T)
    Update neighborhood table
  end if
  Update stopCrit
end while
```

a termination criterion (i.e., the variable betterLocationFound in Algorithm 4.1) is satisfied (e.g., convergence tolerance of the best individuals reaches a certain limit, fitness value becomes below a predefined value, or the number of generations exceeds its limit). If FGA evolves to a better speed and movement direction to minimize the total virtual force on the corresponding node, mobile node adapts this new speed and direction; otherwise, it stops. A node repeats running FGA in this manner until the condition called *stopCrit* is satisfied when the node obtains an acceptable level of

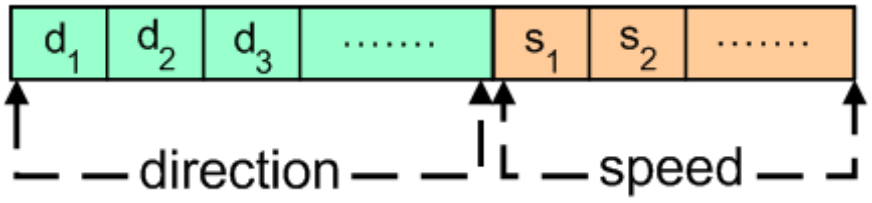


Figure 4.1: Chromosome in FGA

uniformity in its vicinity.

4.1 Chromosome in our FGA

In our GA-based topology control framework, a chromosome includes speed and movement direction as shown in Fig. 4.1. In our mobility model [26,100,101], each mobile node can move into one of six hexagonal directions in a given area of interest called A_{RoI}. As an example, let us assume that mobile nodes can move at four different speeds. Six different directions with four speeds can be coded into 5-bit chromosome ($< d_1d_2d_3s_1s_2 >$). The first three bits ($< d_1d_2d_3XX >$) represent hexagonal movement directions ($< 000 >$ representing north, $< 001 >$ northeast, $< 010 >$ southeast, $< 011 >$ south, $< 100 >$ southwest, and $< 101 >$ northwest). The last two bits of the chromosome ($< XXXs_1s_2 >$) are used for defining different speed values ($< 00 >$ for immobile, $< 01 >$ slower speed, $< 10 >$ normal speed, and $< 11 >$ faster speed). The speed implies the number of hexagonal cells that a node can move in a time unit. For example, if our FGA evolve to a chromosome $< 01110 >$, it means that the corresponding mobile node should move three positions (normal speed) heading south.

4.2 Fitness Function in our FGA

By using a fitness (i.e., objective) function, the quality of each chromosome within a solution space is measured so that each chromosome in a population may be ranked against the others. Our GA-based topology control algorithm for self-spreading of mobile nodes use a fitness function that is based on the virtual forces applied to a mobile node by its neighboring nodes [21]. The virtual force between two neighboring nodes (N_i and N_j) depends on the distance between them and the number of other nodes within their communication ranges. The virtual force applied on a mobile node N_i by its neighboring node N_j is calculated as:

$$F_{ij} = \begin{cases} F_{max} & \text{if } d_{ij} = 0 \\ \sigma_i \left(d_{th} - d_{ij}\right) & \text{if } 0 < d_{ij} < d_{th} \\ 0 & \text{if } d_{th} \leq d_{ij} \leq R_{com} \end{cases} \tag{4.1}$$

where d_{ij} is the Euclidean distance between mobile nodes N_i and N_j, d_{th} is the threshold value to define the local neighborhood, and σ_i is the expected node degree (i.e., it is a function of mean node degree [100] (see Section 3.3.1) and total number of neighbors of N_i) to obtain the highest area coverage in A_{RoI}. If the corresponding node N_i has k number of nodes within its communication range, our FGA calculates the fitness value of the autonomous node N_i as:

$$\begin{aligned} \textit{minimize} \quad &: \quad \sum_{j=1}^{k} F_{ij} = \sum_{j=1}^{k} \sigma_i \left(d_{th} - d_{ij}\right) \text{ for } 0 < d_{ij} \leq d_{th} \\ \textit{subject to} \quad &: \quad d_{mov} \leq d_{max} \end{aligned} \tag{4.2}$$

where d_{mov} is a result that encoded in each chromosome whereas d_{max} is the maximum allowable distance based on N_i's neighbors that N_i can move in one time unit. Note that $d_{mov} \leq d_{max}$ since d_{max} depends on the positions of the N_i's neighbors at a given time. For example, in a perfect equilibrium (i.e., all neighboring nodes are at distance of R_{com} from N_i), since N_i does not have to move for a better fitness, d_{max} is zero.

Recall that the goal of our GA-based approach is to evolve a chromosome composed of a speed and movement direction that minimize the fitness function (i.e., the total virtual force on the corresponding mobile node) given in Eqn. 4.2. Let us now introduce the following lemma to show the relation between d_{mov} and d_{max}.

***Lemma* 1:** *Our* FGA *evolves to a distance per time unit* d_{mov} *such that* $0 < d_{mov} < R_{com}$.

Proof : Based on Eqns. 4.1 and 4.2, our FGA generates solutions which cannot result in losing connections between a mobile node N_i and its neighboring nodes. The upper bound of d_{mov} (i.e., the value of d_{max}) is the distance between N_i and its farthest neighbor. Since the maximum distance without losing direct communication

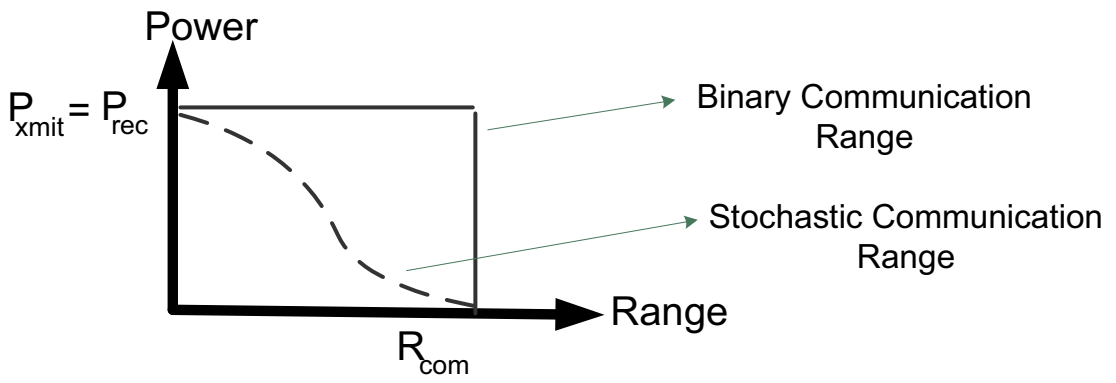


Figure 4.2: Binary and stochastic communication range models

is R_{com} we have $0 < d_{mov} < R_{com}$. ∎

The stability of an autonomous system can be analyzed with conventional Lyapunov theory [102, 103]. Lyapunov's second method provides tools for studying asymptotic stability properties of an equilibrium point of a dynamical system. A scalar output-function, often thought of as a generalized energy, is bounded and decreasing along solutions. If this function has only a single local minimum, and it is strictly decreasing along all non-equilibrium solutions, then one expects that all solutions tend to be close to equilibrium where the output function has a minimum [102]. In other words, Lyapunov theory is used to make predictions about trajectories of an autonomous systems without finding the trajectories. Firstly, let us define Lyapunov stability of an autonomous system.

***Definition* 1:** *(Lyapunov stability of autonomous system [103]) Let x be an equilibrium point for an autonomous system described by*

$$\dot{x} = f(x(t)),\ x(0) = x_0 \tag{4.3}$$

where $f : D \to R^n$ is Lipschitz continuous and $D \subset R^n$ a domain that contains the origin. Let us define $V : D \to R$ a continuously differentiable, positive definite function in D.

1. *if $\dot{V}(x) = [\frac{\partial V}{\partial x}]\ f$ is negative semidefinite, then $x = 0$ is a stable equilibrium point.*

2. *if $\dot{V}(x) = [\frac{\partial V}{\partial x}]\ f$ is negative, then $x = 0$ is an asymptotically stable equilibrium point.*

Let us build a second order time invariant linear dynamic system model for a mobile node N_i with its neighboring nodes using a potential energy $V(d_{ij})$ as Lyapunov function

$$\ddot{d}_i = \vec{f}_i = [f_i^x f_i^y]^T \tag{4.4}$$

where $\vec{f}_i$ is environmental input to N_i as the total virtual force exerted on its by its neighbors [104–106].

Let us construct a potential energy $V(d_{ij})$. The virtual force can be formulated as the negative gradient of the potential energy by $\vec{f}_i = \sum_{j=1}^{k} \vec{f}_{ij} = -\sum_{j=1}^{k} \Delta V(d_{ij})$. Now we can define a continuously differentiable Lyapunov function which is the combination of potential and kinetic energies as:

$$\phi = \frac{1}{2}\dot{d}_i^{\,T} \cdot \dot{d}_i + \frac{1}{2}\sum_{j=1}^{k} V(d_{ij}) \tag{4.5}$$

Lemma 2: *Using our* FGA*, a mobile node N_i eventually reaches an asymptotically*

stable state.

Proof : Let us first start differentiate the Lyapunov function given in Eqn. 4.5:

$$\dot{\phi} = \dot{d_i}^T \cdot \ddot{d_i} + \sum_{j=1}^{k} \Delta V(d_{ij}) \tag{4.6}$$

For simplicity, suppose the dynamical model has unit mass. Then, Eqn. 4.6 can be rewritten as:

$$\dot{\phi} = -\dot{d_i}^T \cdot c\dot{d_i} + \sum_{j=1}^{k} \sigma_i \dot{d_{ij}} \tag{4.7}$$

From Eqn. 4.7 with $\dot{d_{ij}} = \frac{\dot{d_i}^T (d_i - d_j)}{\|d_i - d_j\|}$, we have:

$$\dot{\phi} = -2\dot{d_i}^T \cdot \sigma_i \vec{f_i} \tag{4.8}$$

σ_i is always positive since it is a function of the mean node degree (see Section 3.3.1) and the total number of neighbors of N_i, and, therefore, $\dot{\phi} \leq 0$ in Eqn. 4.8. It is shown by Barbalat that as time approaches infinity, $\dot{\phi}$ approaches zero for systems with $\dot{\phi} \leq 0$ [107]. We conclude that the dynamic model of our FGA is asymptotically stable and its fitness function defined in Eqn. 4.2 eventually reaches a minimum. ∎

4.3 Effectiveness in Area Coverage

The concept of area coverage as a function of mobile robots was introduced by Gage [108]. Three types of area coverage are:

- Blanket coverage: the objective is to maximize the total occupied area,
- Barrier coverage: the nodes try to minimize the probability of undetected target penetration thought a barrier,
- Sweep coverage: it is equivalent to a moving barrier coverage.

We can consider that the dynamic topology control of autonomous mobile nodes to obtain the highest area coverage without global localization information is a combination of blanket and sweep coverages.

Uniform distribution of mobile nodes in a geographical terrain (i.e., a region of interest, A_{RoI}) relates to the issue of how well each point in an autonomous node's communication range is covered. One of the main objectives is to deploy mobile nodes in strategic ways (e.g., uniformly) such that a maximum area coverage is achieved according to the needs of the underlying applications. Therefore, effectiveness in area coverage (called A_{eff}) is an important performance metric for our GA-based topology control framework as it measures the success of mobile nodes' distribution over A_{RoI}.

The communication range of each autonomous mobile node can be modeled as either binary or stochastic communication range as seen in Fig. 4.2. In binary communication range, the probability of communicating with another mobile node within the communication range (R_{com}) is one; otherwise it is zero. However, in the stochastic communication range, the probability of communicating with another mobile node is a decaying function of their distance [34]. In this dissertation, for simplicity, we prefer using the binary model with the same communication range for all mobile nodes. The

results, however, can easily be extended for stochastic communication range model with heterogeneous communication range of R_{com} for autonomous mobile nodes.

Definition 2: *Suppose a mobile node N_i with a communication range of R_{com} using an omni directional antenna is located at (x_i, y_i). Then a point p at (x_p, y_p) in A_{ROI} is said to be covered by N_i if $\sqrt{(x_i - x_p)^2 + (y_i - y_p)^2} \leq R_{com}$.*

Definition 3: *The effectiveness in the area coverage (called A_{eff}) is given as:*

$$A_{eff} = \frac{\bigcup_{i=1}^{n} A_i}{A_{\text{ROI}}} \tag{4.9}$$

where A_i is the area covered by node N_i and $A_{eff} \in [0, 1]$.

If a node is located close to the center of the terrain (i.e., away from borders at least R_{com} units), the full area of a circle around the node with a radius of R_{com} is counted as the covered region if there are no other nodes overlapping with this node's coverage. When the node is near the boundary, only the partial area is included in A_{eff} computation [23–25].

4.4 Uniformity

Mobile nodes in a MANET have limited energy resources. Minimizing the total energy consumption is one of the important optimization objectives for autonomous nodes since they usually carry limited energy resources, such as batteries. If mobile nodes

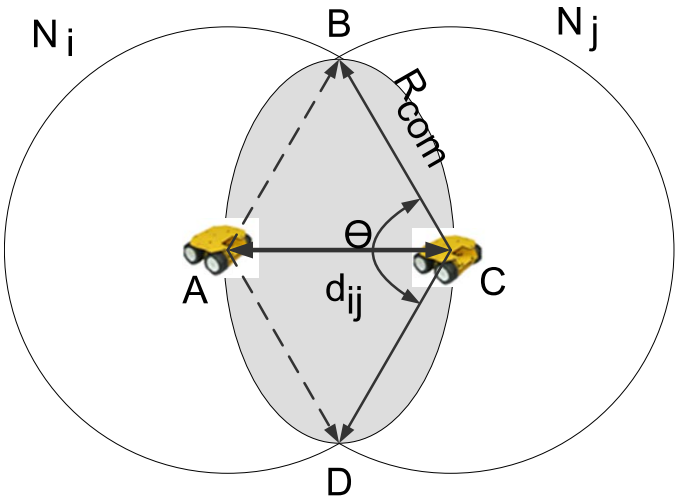


Figure 4.3: Overlap area between mobile nodes N_i and N_j when $d_{ij} = R_{com}$

know their start and end points before mission, there are existing studies focusing on energy-efficient motion planning for mobile nodes [31, 80]. However, in our problem, mobile nodes need to self-spread in A_{RoI} without a priori or global information. In general, uniformly distributed mobile nodes in A_{RoI} use their limited resources more evenly than the non-uniformly distributed nodes for movement and communication.

Definition 4: *Uniformity for a mobile node N_i, called u_i, is defined as the average of the standard deviation of the overlap area of N_i and its neighbors (Fig. 4.3):*

$$u_i = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\Lambda_{ij} - \Lambda_{\Theta})^2} \tag{4.10}$$

where u_i is the local uniformity value for the mobile node of N_i, m is the number of neighbors, Λ_{ij} is the overlap area between N_i and its neighboring node of N_j, and Λ_{Θ} is the overlap area between N_i and N_j when $d_{ij} = R_{com}$ (if N_i and N_j move any further they will lose communication).

A smaller value of u_i means that neighboring nodes are separated more uniformly. Note that the overlap area between mobile nodes N_i and N_j (see Fig. 4.4) is $\Lambda_\Theta = R_{com}^2(\Theta - sin(\Theta))$, where $\Theta = 2cos^{-1}(\frac{d_{ij}}{R_{com}})$. The following lemma shows the minimum overlap area between two neighboring nodes.

Lemma 3: *The overlap area between two neighboring nodes N_i and N_j ($1 \leq i, j \leq m$) is minimized for Λ_Θ where $\Theta = \frac{2\pi}{3}$.*

Proof : If mobile nodes N_i and N_j can communicate with each other, we have $d_{ij} \leq R_{com}$.In order to minimize the overlap area of Λ_{ij}, d_{ij} must be maximized (i.e., $d_{ij} = R_{com}$). As seen in Fig. 4.4, $\widehat{ABC}$ and $\widehat{ADC}$ are equilateral triangles if $d_{ij} = R_{com}$. Hence, $\Theta = \frac{2\pi}{3}$. ∎

Let us now show that our FGA improves the local uniformity u_i for a mobile node N_i if the fitness value of N_i is greater at time t than at $t+1$.

Lemma 4: *Our FGA reduces u_i for a mobile node N_i with k neighbors if $f_i^t > f_i^{t+1}$.*

Proof : Using Eqn. 4.2, the corresponding node's fitness value at time t and $t+1$ are

$$f_i^t = \sum_{j=1}^{k}(\sigma_i(d_{th} - d_{ij}^t)) \tag{4.11}$$

$$f_i^{t+1} = \sum_{j=1}^{k}(\sigma_i(d_{th} - d_{ij}^{t+1})) \tag{4.12}$$

The corresponding autonomous mobile node moves to a new location if and only if

$$
\begin{aligned}
f_i^t > f_i^{t+1} \quad &\Rightarrow \quad k\sigma_i d_{th} - \sum_{j=1}^{k}(d_{ij}^t) > k\sigma_i d_{th} - \sum_{j=1}^{k}(d_{ij}^{t+1}) \\
&\Rightarrow \quad \sum_{j=1}^{k}(\sigma_i(d_{th} - d_{ij}^t)) > \sum_{j=1}^{k}(\sigma_i(d_{th} - d_{ij}^{t+1})) \\
&\Rightarrow \quad \sum_{j=1}^{k}(d_{ij}^t) < \sum_{j=1}^{k}(d_{ij}^{t+1}) \qquad (4.13)
\end{aligned}
$$

From Eqn. 4.10 , we have

$$
u_i^t = (\frac{1}{m}\sum_{j=1}^{k})(\Lambda_{ij}^t - \Lambda_{\frac{\pi}{3}})^{\frac{1}{2}} = (\frac{1}{m}\sum_{j=1}^{k}(R_{com}^2(\Theta^t - sin(\Theta^t)) - \Lambda_{\frac{\pi}{3}})^2)^{\frac{1}{2}}
$$

$$
u_i^{t+1} = (\frac{1}{m}\sum_{j=1}^{k})(\Lambda_{ij}^{t+1} - \Lambda_{\frac{\pi}{3}})^{\frac{1}{2}} = (\frac{1}{m}\sum_{j=1}^{k}(R_{com}^2(\Theta^{t+1} - sin(\Theta^{t+1})) - \Lambda_{\frac{\pi}{3}})^2)^{\frac{1}{2}}
$$

$$
\text{where } \Theta = 2cos^{-1}(\tfrac{d_{ij}}{R_{com}})
$$

$$
u_i^t = (\frac{1}{m}\sum_{j=1}^{k}(R_{com}^2(2cos^{-1}(\frac{d_{ij^t}}{R_{com}}) - sin(2cos^{-1}(\frac{d_{ij^t}}{R_{com}}))) - \Lambda_{\frac{\pi}{3}})^2)^{\frac{1}{2}} \qquad (4.14)
$$

$$
u_i^{t+1} = (\frac{1}{m}\sum_{j=1}^{k}(R_{com}^2(2cos^{-1}(\frac{d_{ij^{t+1}}}{R_{com}}) - sin(2cos^{-1}(\frac{d_{ij^{t+1}}}{R_{com}}))) - \Lambda_{\frac{\pi}{3}})^2)^{\frac{1}{2}} \qquad (4.15)
$$

Based on Eqn. 4.13, $d_{ij}^t > d_{ij}^{t+1}$ which means:

$$
\begin{aligned}
&\frac{d_{ij}^t}{R_{com}} > \frac{d_{ij}^{t+1}}{R_{com}} \Rightarrow cos^{-1}(\frac{d_{ij}^t}{R_{com}}) < cos^{-1}(\frac{d_{ij}^{t+1}}{R_{com}}) \\
&\Rightarrow \underbrace{R_{com}^2(2cos^{-1}(\frac{d_{ij^t}}{R_{com}}) - sin(2cos^{-1}(\frac{d_{ij^t}}{R_{com}}))}_{\eta^t} > \\
&\qquad \underbrace{R_{com}^2(2cos^{-1}(\frac{d_{ij^{t+1}}}{R_{com}}) - sin(2cos^{-1}(\frac{d_{ij^{t+1}}}{R_{com}})))}_{\eta^{t+1}} \\
&\qquad\qquad \Rightarrow \eta^t > \eta^{t+1} \qquad (4.16)
\end{aligned}
$$

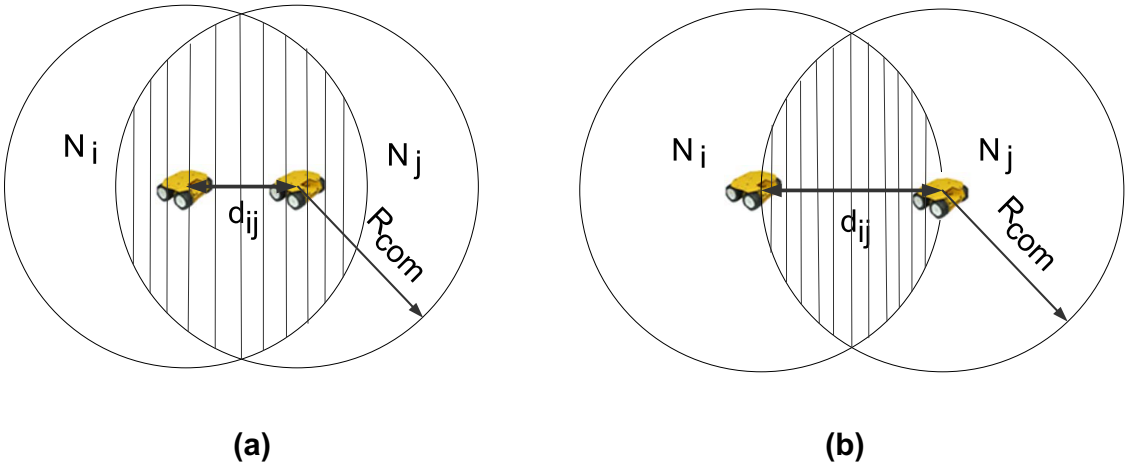


Figure 4.4: Overlap area between mobile nodes N_i and N_j when (a) $d_{ij} < R_{com}$ and (b) $d_{ij} = R_{com}$

.

Eqn. 4.16 yields $u_i^t > u_i^{t+1}$. Since, from Definition 4, a smaller value of u_i implies better uniformity, N_i has better uniformity u_i at time $t+1$. ∎

***Definition* 5:** *Average of all local uniformities of u_i ($i = 1, \cdots, N$) yields the uniformity of all mobile nodes as*

$$U = \frac{1}{N} \sum_{i=1}^{N} u_i \tag{4.17}$$

where N is the total number of mobile nodes in a MANET.

The following lemma states that if there are enough nodes to cover a given a convex terrain, FGA distributes the nodes such that the MANET remains connected.

***Lemma* 5:** *For a convex terrain of A_{RoI} completely covered by mobile nodes N_i and N_j,* FGA *guarantees that N_i and N_j remain connected.*

Proof : Let us consider two conditions contradicting the lemma. First suppose N_i and N_j cover A_{RoI} but do not communicate with each other (i.e., $d_{ij} > R_{com}$) as seen in Fig. 4.5 (a). From Eqn. 4.2, since σ_i and σ_j are ∞ when N_i and N_j are not connected, they will start moving to improve their fitness values. FGA will eventually force N_i and N_j to move towards each other such that they will be connected. Note that if N_i and N_j move away from each other more than $2R_{com}$ they no longer cover A_{RoI} and hence contradicted the condition in the lemma.

Now as the second contradictory condition, consider the case that N_i and N_j can communicate with each other (i.e., $d_{ij} < R_{com}$), however, they do not cover A_{RoI} as shown in Fig. 4.5 (b). From Eqn. 4.2, FGA will move N_i and N_j to improve their fitnesses by increasing d_{ij} between them in Fig. 4.5 (c) until the distance is R_{com} which minimizes the local uniformity given in Eqn. 4.10.

Therefore, FGA guarantees that if N_i and N_j completely cover A_{RoI}, they are connected. ∎

If u_i is zero, our FGA evolves to a chromosome (see Section 4.1) having zero speed as stated by the following lemma.

***Lemma* 6:** *For a given mobile node N_i with k neighboring nodes, if the local uniformity u_i is zero, our* FGA *guarantees that N_i is immobile.*

Proof : If the local uniformity u_i for N_i is zero, $\Lambda_{ij} = \Lambda_{\Theta}$, $j \in [1, k]$ in Definition 4. As the overlap area between mobile nodes N_i and N_j (Λ_{ij}) equals to Λ_{Θ}, the distance

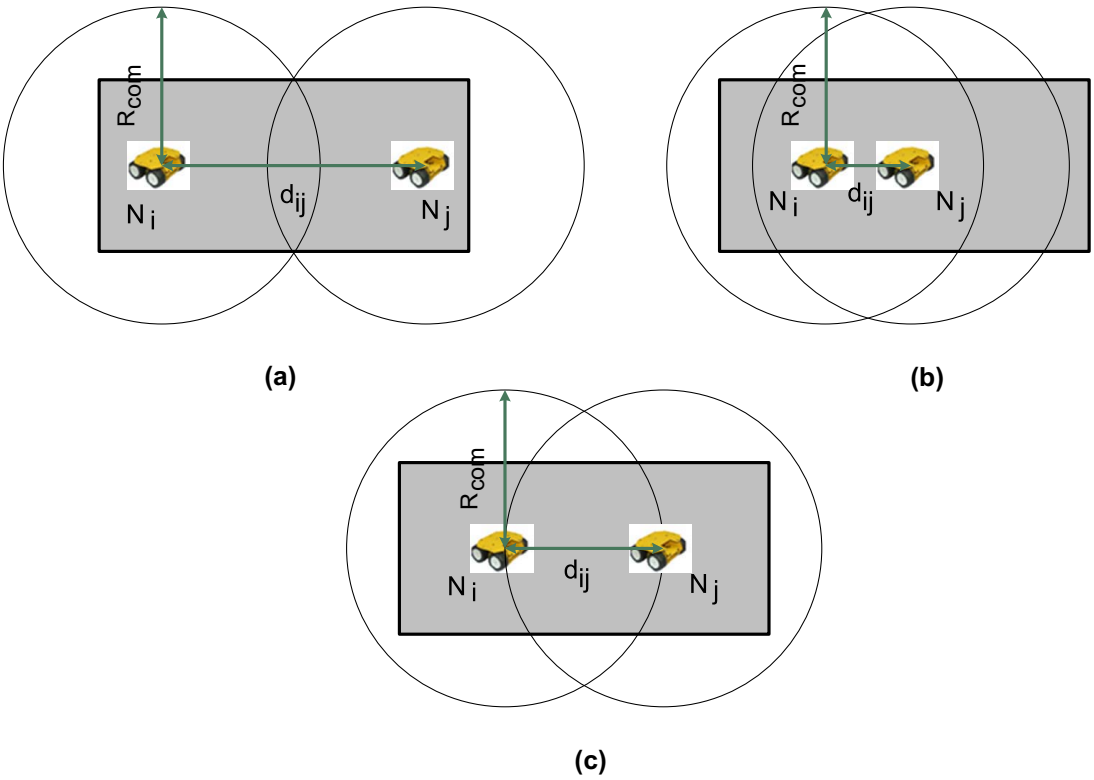


Figure 4.5: Area coverage when (a) $d_{ij} > R_{com}$, (b) $d_{ij} < R_{com}$, and (c) $d_{ij} = R_{com}$.

between the mobile nodes is $d_{ij} = R_{com}$. As seen in Eqn. 4.1, the total virtual force exerted on N_i is zero for $d_{ij} = R_{com}$. Our FGA, thus, evolves to a chromosome that has zero speed. ∎

Now we can introduce the following corollary about the connectivity of the nodes after FGA distributes the nodes using the fitness function given in Eqn. 4.1.

***Corollary* 1:** *For a convex terrain of A_{RoI} completely covered by N mobile nodes, our* FGA *guarantees that any two mobile nodes N_i and N_j $(i, j = [1, N])$ remain connected.*

Chapter 5

Experimental Analysis of FGA

5.1 Simulation Software

We implemented a simulation software system in Java to study the effectiveness of our distributed GA-based algorithms for a uniform distribution of knowledge sharing agents. Eclipse SDK version 3.2.0 was used as the development environment, and Mason, a fast discrete-event multi-agent simulation library core developed by George Mason University ECJLab, was used for the GUI interface.

The simulation software implementation has more than 4,500 lines of algorithmic Java code. To avoid possible inefficiencies, we developed our algorithms without using any existing GA libraries. Our design philosophy was to build a GA-based application to which a programmer can easily add new features (e.g., different types of crossover, or different rules for tournament, etc.) and new evolutionary computation

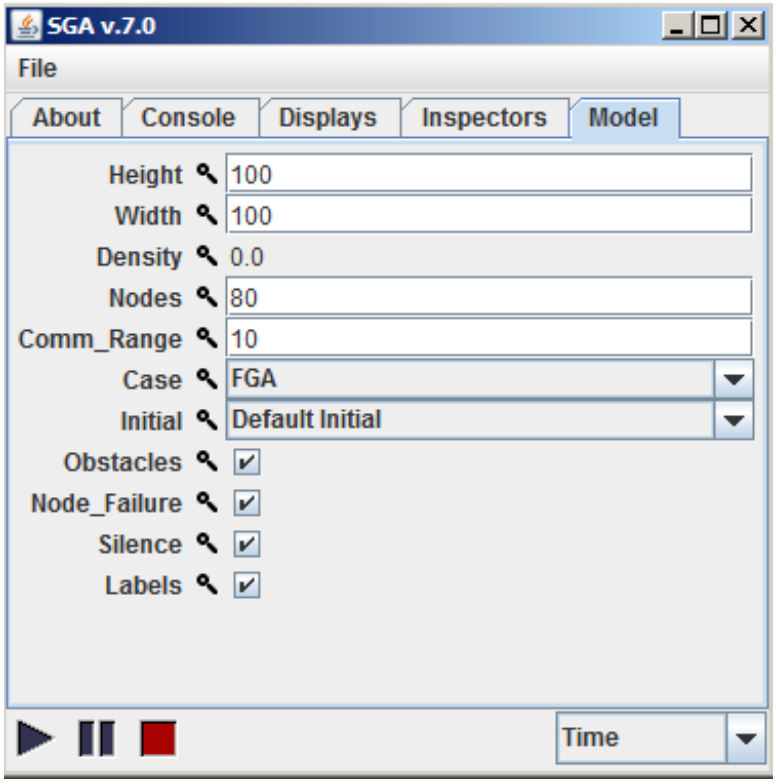


Figure 5.1: Graphical user interface for our GA software package: a screen shot of user inputs

approaches. Our simulation software was implemented such that it runs as a multi-agent application which imitates a real-time topology control scenario. As a result, the observations from our simulation software match closely to those from the real testbed experiments given in Section 5.2.

Sample screen shots of the graphical user interface of our simulation software are shown in Figs. 5.1 and 5.2. User-defined input parameters for our software include:

- Nodes(N): total number of mobile nodes,
- Comm_Range(R_{com}): communication range of a mobile node,
- Case: type of evolutionary algorithms,

- T_{max}: maximum number of iterations,
- Initial (Initial deployment type): currently there are three different initial deployment strategies for the mobile nodes:
  - (i) start from the northwest corner
  - (ii) place the nodes randomly in a given area
  - (iii) start from a given coordinate (e.g., the center of the area) in the terrain
- Height and weight: size of the geographical terrain (d_{max}),

The simulation software also allows the inclusion of obstacles on used defined locations, random node fails, and invoking silent mode (i.e., no communication among autonomous nodes in a MANET for given time periods).

Fig. 5.2 shows a sample initial deployment of autonomous mobile nodes starting from the northwest sector of a given terrain. Note that selecting a corner as the initial deployment region represents a more realistic approach of the topology control problem for the knowledge sharing mobile nodes than other deployment possibilities over an unknown terrain. For example, in an earthquake rescue, a mine clearing mission, a military mission in hostile area, or a surveillance operation, all mobile nodes may be forced to enter the operation area from the same vicinity rather than random or central node deployment.

Our simulation software also has the ability to run experiments using a previously used initial mobile node distribution and initial conditions from previous runs (i.e., the

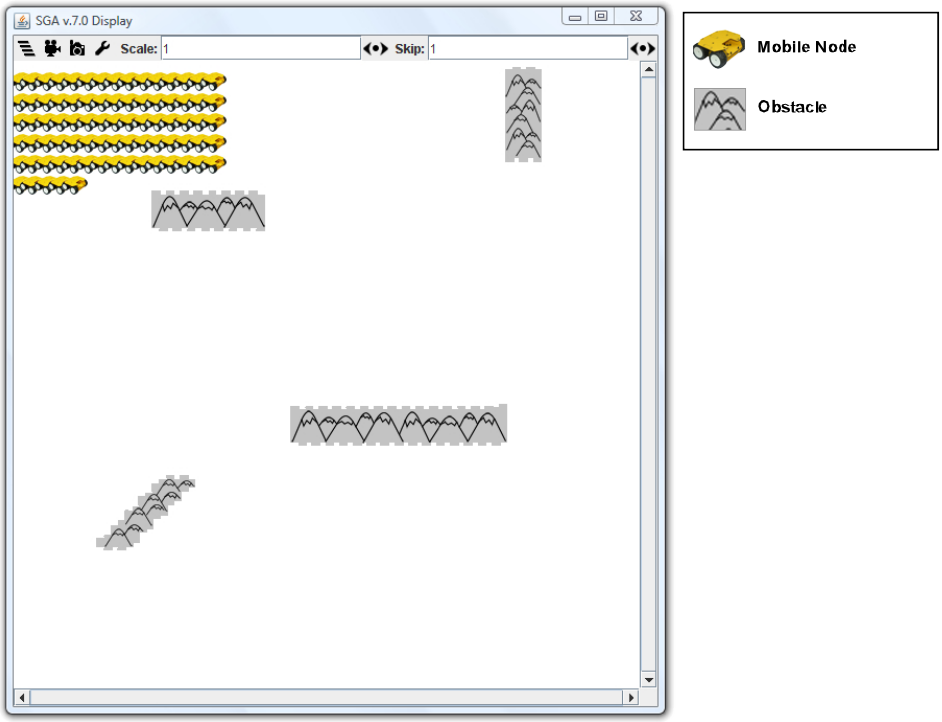


Figure 5.2: Graphical user interface for our GA software package: a screen shot from an initial mobile node distribution for FGA

initial data for each mobile node includes a starting coordinate, speed, and direction). This ability is important since each experiment is repeated many times to eliminate the noise in the collected data and provide an accurate stochastic behavior of GA-based algorithms.

5.1.1 Experiment Results

In this section, we evaluate our GA-based approach with respect to A_{eff} (see Section 4.3) and the deployment time. As explained in Section 4.3, A_{eff} is one of the most important metrics for self-spreading algorithms. It shows the portion of the geographical terrain which can be located within at least one of the mobile node's

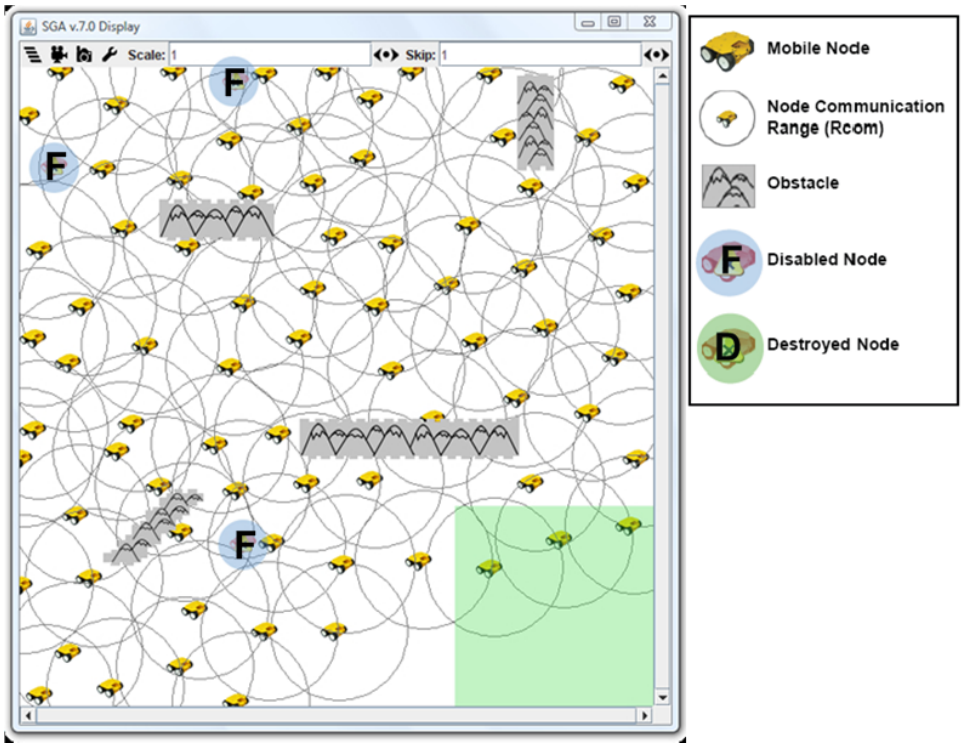


Figure 5.3: Graphical user interface for our GA software package: a screen shot of mobile nodes distribution for the first application at $T = 400$

communication range of R_{com}. One of the goals of our FGA is to obtain highest possible A_{eff} value and maintain it in the presence of unknown obstacles, hostile attacks, malfunctions, and silent mode. Deployment time shows the total time it takes for the autonomous mobile nodes in a MANET to converge towards a uniform distribution over a geographical area. Deployment time includes communication overhead, FGA processing time, and moving from one location to another. Another important performance metric for our algorithm is the network recovery time (i.e., the time it takes for FGA to compensate for the lost coverage) after any attack, silence mode, or malfunctions.

We consider 80 mobile nodes with the same initial node distribution over an unknown

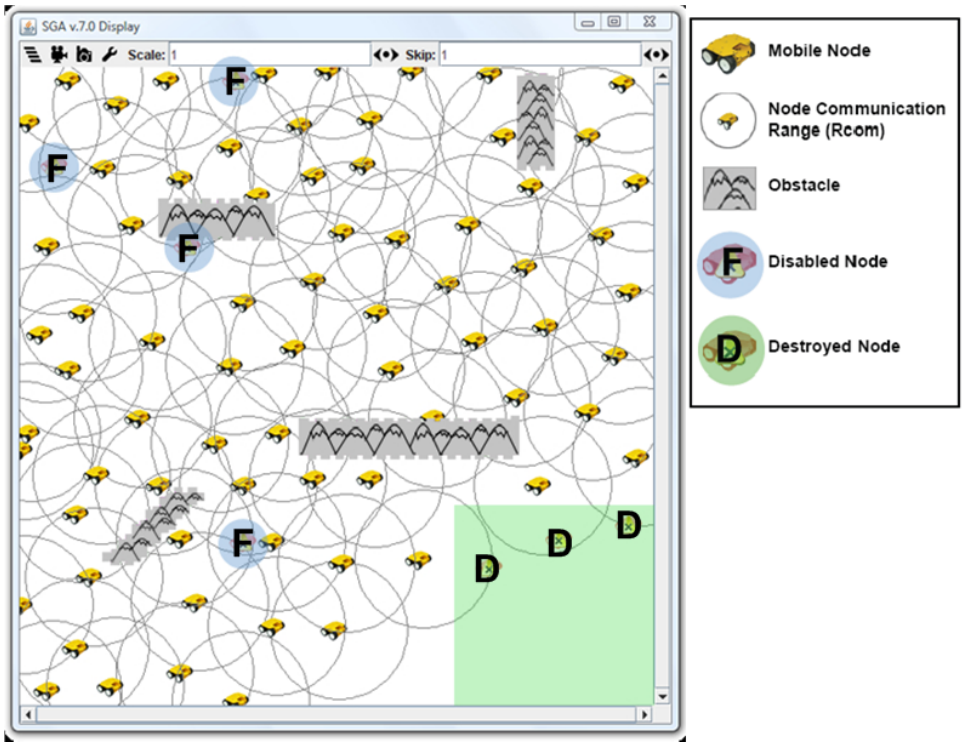


Figure 5.4: Graphical user interface for our GA software package: a screen shot of mobile nodes distribution for the first application at $T = 401$ (after the first attack)

region. Each mobile node has the same limited communication range of R_{com}, and, hence, can only be aware of its neighbors and the obstacles if they are located in the node's sensing and communication ranges. The initial mobile entity deployment and the positions of the obstacles are shown in Fig. 5.2 where mountains represent physical obstacles. As movement of each mobile node is only affected by the current status of neighboring nodes, each node is adaptive to environment changes such as node failures, various terrain shapes, and hostile attacks. To evaluate the performance and effectiveness of our FGA framework with respect to A_{eff} and the deployment time, we experiment with two types of applications. First, mobile nodes are deployed in a hostile region where some nodes can be affected during and after deployment. In this application, there are two events that affect the deployment of the mobile nodes.

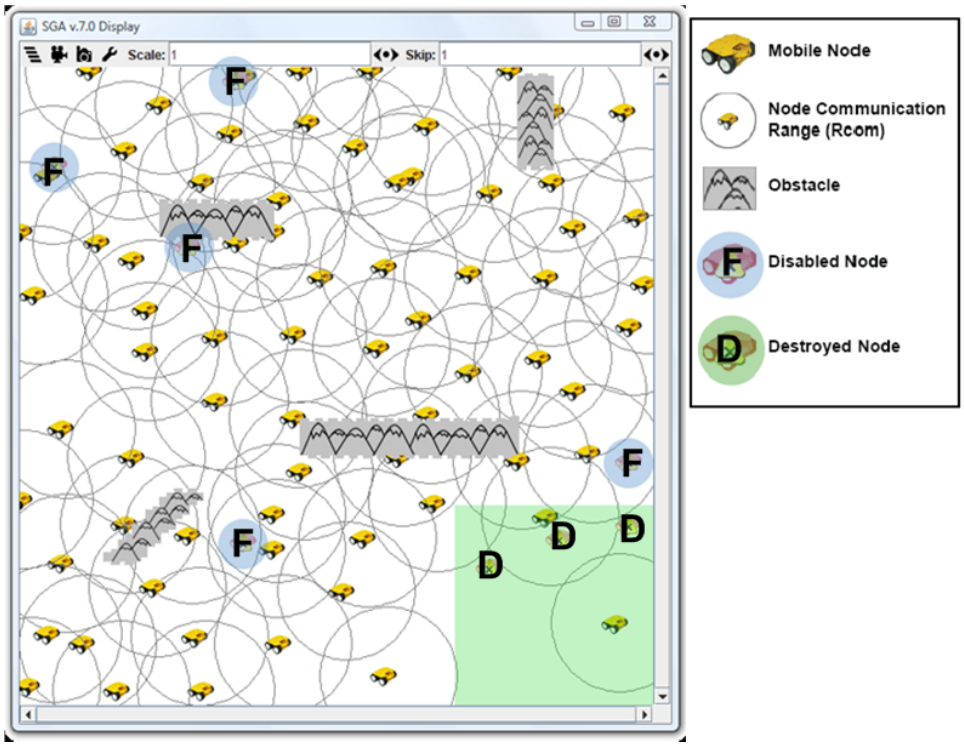


Figure 5.5: Graphical user interface for our GA software package: a screen shot of mobile nodes distribution for the first application at $T = 600$

Some nodes can lose their communication functionality due to malfunctions while others are destroyed due to enemy attacks. The nodes affected by malfunctions and hostile activity are considered disabled for the rest of the simulation experiment. As a consequence of such as events, the remaining nodes must reconfigure their positions to compensate the missing area coverage due to lost team members. In a second application, autonomous mobile nodes intentionally stop communicating with the neighboring nodes located within R_{com} during short periods of time in order to go unnoticed by adversary forces and avoid being the target of enemy attacks (*silence mode*). Afterwards, all autonomous mobile nodes resume transitions again.

In Fig. 5.3, we can observe that, in spite of the obstacles, the mobile nodes using

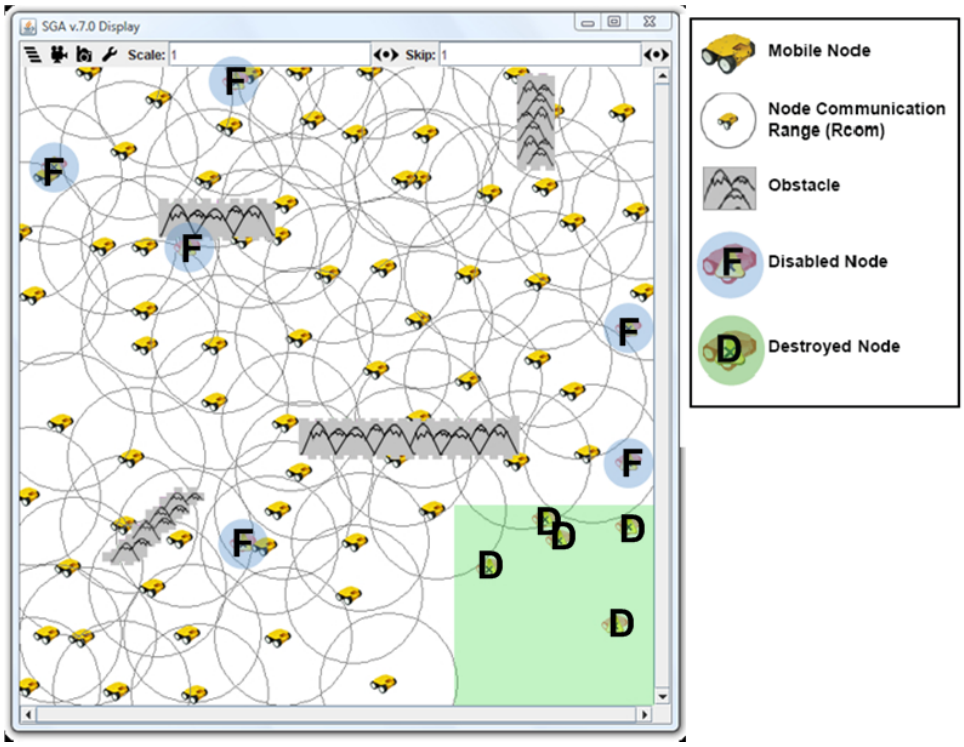


Figure 5.6: Graphical user interface for our GA software package: a screen shot of mobile nodes distribution for the first application at $T = 601$ time unit (after the second attack)

FGA obtain an almost uniform coverage of the area during the first 400 time units of the first application. Large circles represent the communication range for the mobile nodes. The circles marked with **F** indicate disabled autonomous nodes. The gray square in the lower right corner represents the region where enemy attacks take place. At $T = 401$, the first hostile attack takes place and destroys three mobile nodes, as shown in Fig. 5.4. The destroyed mobile nodes are labeled as **D** and are shown as small circles. In addition, three mobile agents have experienced a malfunction, reducing the total number of mobile nodes to 73.

Figs. 5.5 and 5.6 show the screen shots of the geographical area before and after the second enemy attack, respectively. As we can observed, the remaining mobile

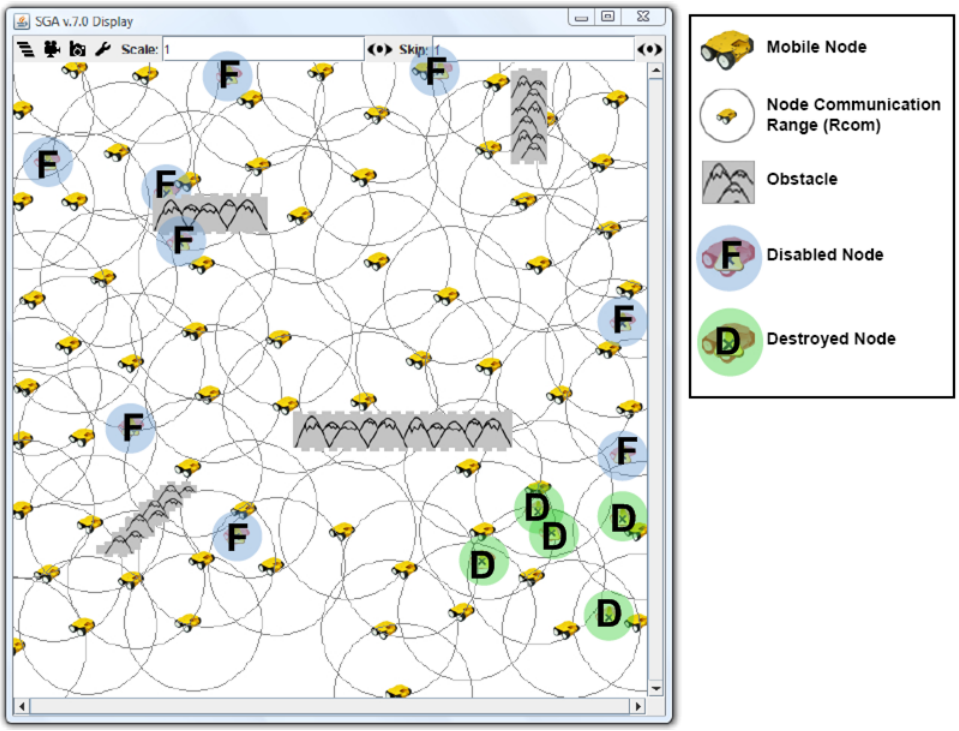


Figure 5.7: Graphical user interface for our GA software package: a screen shot where 66 remaining nodes (eight nodes are disabled throughout the region and six nodes are destroyed by hostile attacks at the south east region) are distributed over an unknown terrain at T=1000

nodes keep performing our GA-based topology control approach, and readjust their positions for a uniform coverage. After the second enemy attack, another two mobile agents are destroyed, leaving 69 nodes in the experiment.

The final mobile node distribution after running FGA for 1000 time units is presented in Fig. 5.7, where the remaining autonomous nodes reshape their deployment to cover the attack region. Also, the network is considered fully connected since all nodes in the network are reachable by other nodes through either one-hop or multi-hop communication.

Fig. 5.8 shows the improvement in A_{eff} through time for the experiment as the nodes

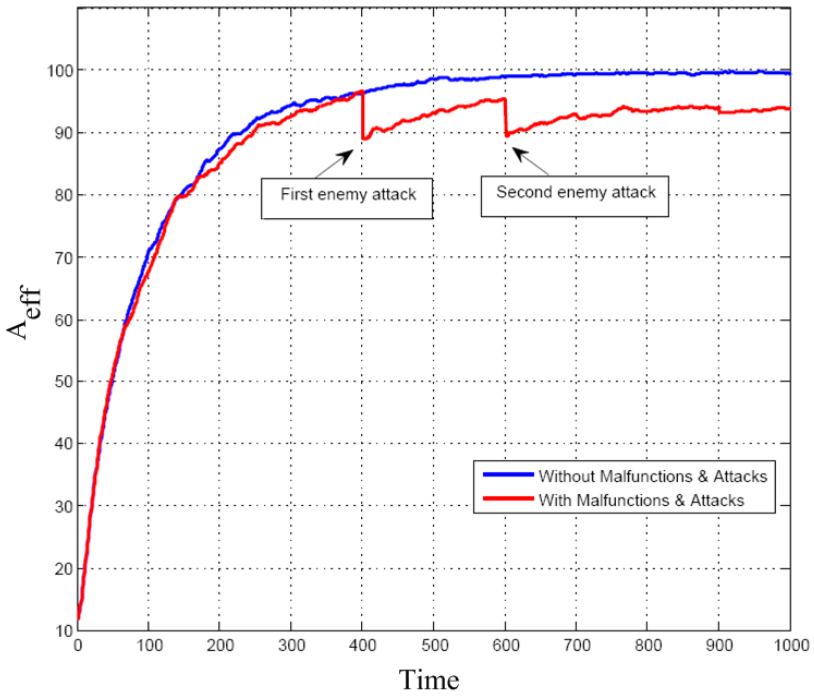


Figure 5.8: A_{eff} after 1000 time units

perform our GA-based topology control framework. As shown in Fig. 5.8 mobile nodes using FGA successfully deploy themselves around the obstacles if there were no hostile activity in the area, achieving a A_{eff} value of 99% at $T = 1000$. Meanwhile, when the nodes undergo malfunctions and hostile retaliation, we observe approximately 97% of the total area coverage at $T = 400$. After the first attack, there is a drop in A_{eff} due to the lost nodes, which recovers after 200 time units ($T = 600$) to an A_{eff} value of 95%. Similarly, after the second attack, there is a drop of A_{eff} at $T = 601$, which is then compensated by the remaining nodes after they reposition themselves at approximately 300 time units after the second attack ($T = 1000$).

Fig. 5.9 shows the A_{eff} for *silence mode* experiments in a MANET. Autonomous mobile nodes intentionally stop communicating with their neighbors during short periods of

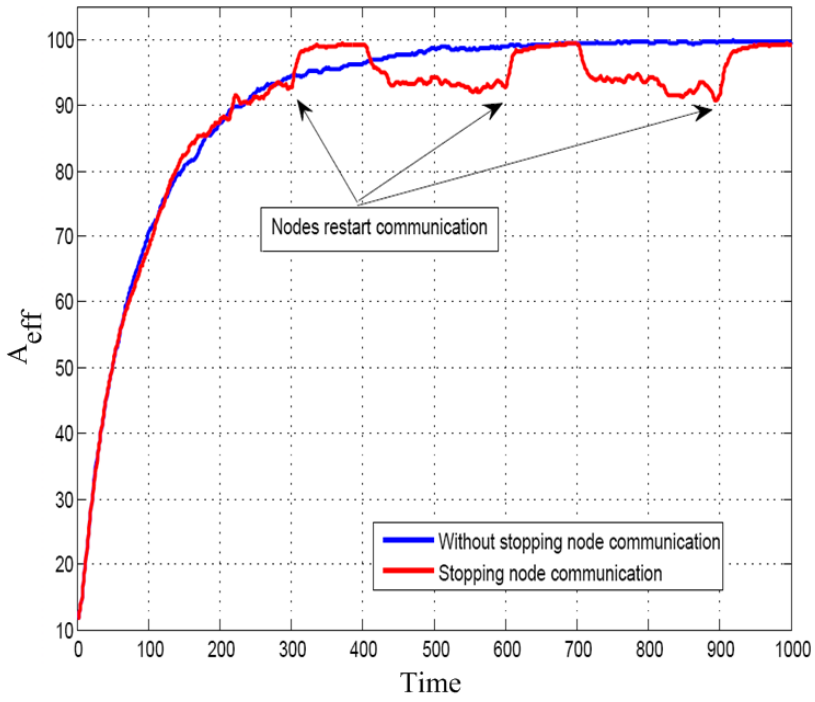


Figure 5.9: A_{eff} after 1000 time units for silence mode

time, consequently, they do not execute the FGA application in order to prevent being detected and destroyed by adversary forces. In this simulation experiment, mobile nodes perform our GA-based topology control application for 100 consecutive steps starting at $T = 0, 300, 600$, and 900 time units. We can observe that during these periods, the nodes rapidly adjust their location, speed, and directions resulting in a significant increase of A_{eff} values to almost normal numbers. As seen in Fig. 5.9, when mobile nodes stop communicating, A_{eff} value immediately drops. This result show that the mobile nodes randomly deciding their next speed and movement directions during silent mode lose area coverage. When our FGA starts running by mobile nodes after silent mode, autonomous nodes use their neighborhood information to decide their next location which results in sudden increase at the A_{eff}.

5.2 Testbed Implementations

Most of the research in wireless ad-hoc networks is based on software tools simulating network environments under strictly controlled conditions rather than implementing realistic testbeds mainly due to their extreme cost of design, operation and difficulty of adapting real-time topological changes. However, a realistic testbed is very important and useful to learn physical and radio characteristics, test unknown implementation issues, and discuss hardware and software requirements for FGA implementations in MANETs. To validate our framework and to study the effectiveness of our GA-based topology control algorithms, we have implemented four different testbeds using:

- FPGA devices (*Virtex-II Pro*™ [109]), laptops, and desktops,
- Small robotic units (*iRobots*™ [110] controlled by *gumstix*™ [111] processors with wireless capabilities),
- Virtual machine technology (VM*ware*™ [112]),
- Off-the-shelf laptops and PDAs.

Our testbeds provides a proof-of-concept for implementing GAs in a distributed robotic environments with a diverse range of mobile nodes and configurations. They also validate the experiment results from our simulation software.

5.2.1 Testbed Implementation with FPGA Devices, Laptops, and Desktops

This testbed platform consists of *Xilinx* ML310™ development boards with Virtex-II Pro FPGA™ devices, laptops and desktops as shown in Fig. 5.10 [113]. It uses off-the-shelf wireless PCI cards that complies with the PCI Local Bus Specification version 2.3 on the *Xilinx* ML310™ with Power PC Virtex-II™ Pro based Processor running *VxWorks*™. We choose wireless hardware support (*Atheros AR521x* chip set) under network devices 802.11a/b/g components and wireless mode support to include component support for ad-hoc IBSS to build the driver module into VxWorks™ real-time kernel. Our testbed architecture allows us using hybrid deployment of various hardware and software components as shown in Fig. 5.10.

This testbed allows us to implement and study different scenarios of our FGA for uniform node distribution. Each node is represented by one device (i.e., FPGA™, laptop, or desktop) as shown in Fig. 5.10. It is programmed in C++ and runs in Linux, Cygwin, and real-time embedded platforms such as VxWorks™. The experiments using this testbed represent realistic MANET conditions where each node has autonomous mobility and wireless communication capabilities. Software in all mobile nodes are configured with identical characteristics. For example, R_{com}, for simplicity, is set as the same for all mobile nodes.

By the nature of real-time testbed applications, a latency factor in network bound data interactions compounding with the wireless network characteristics must be

Figure 5.10: Wireless ad hoc network topology in testbed using FPGA devices, laptops, and desktops

considered in our GA-based topology control framework implementation. In addition, there can be late arriving acknowledgments, delayed responses, and other timing related synchronization issues. Each mobile node broadcasts a periodic heartbeat message to the maximum radial distance allowed by the configured communication ranges of R_{com}. The periodical heartbeat broadcasts includes the node name, coordinates and a time stamp. The heartbeat broadcasts are ignored at the corresponding node if the calculated distance between a source and a destination nodes is greater than R_{com} since the mobile nodes would hear each other only if their communication ranges overlap. Otherwise, the receiving node updates its neighborhood table with the heartbeat message information. If no heartbeat message is received from a node that is already in the table for longer than a certain time, that node is assumed to

move out of range and purged from the table. A new entry is created when a heartbeat message is received the first time. It is important to note that the heartbeat messages broadcast at every Δt seconds by each mobile node and are not synchronized since our testbed represents a realistic scenario and does not employ a central controller. The collected local neighborhood information is then used by each mobile node as an input for our FGA to decide the next speed and direction. NITEXPIRYTIME is a configurable attribute that indicates the longest time allowed since the last heartbeat message from a neighbor. For instance, if no heartbeat message received from a neighbor within the last NITEXPIRYTIME time units, that neighbor is purged from the neighborhood table. To cover cost/benefit ratio for mobile nodes reachability and UDP packet drops (heartbeat message loss) due to wireless characteristics, NITEXPIRYTIME can be considered as a refresh cycle to adjust the granularity for which the neighborhood table gets updated. This provides an effective and configurable mechanism to find neighbors within the communication range where the multi-cast messages for performing genetic operator requests (e.g., crossover, mutation etc.) would be responded with positive or negative acknowledgments.

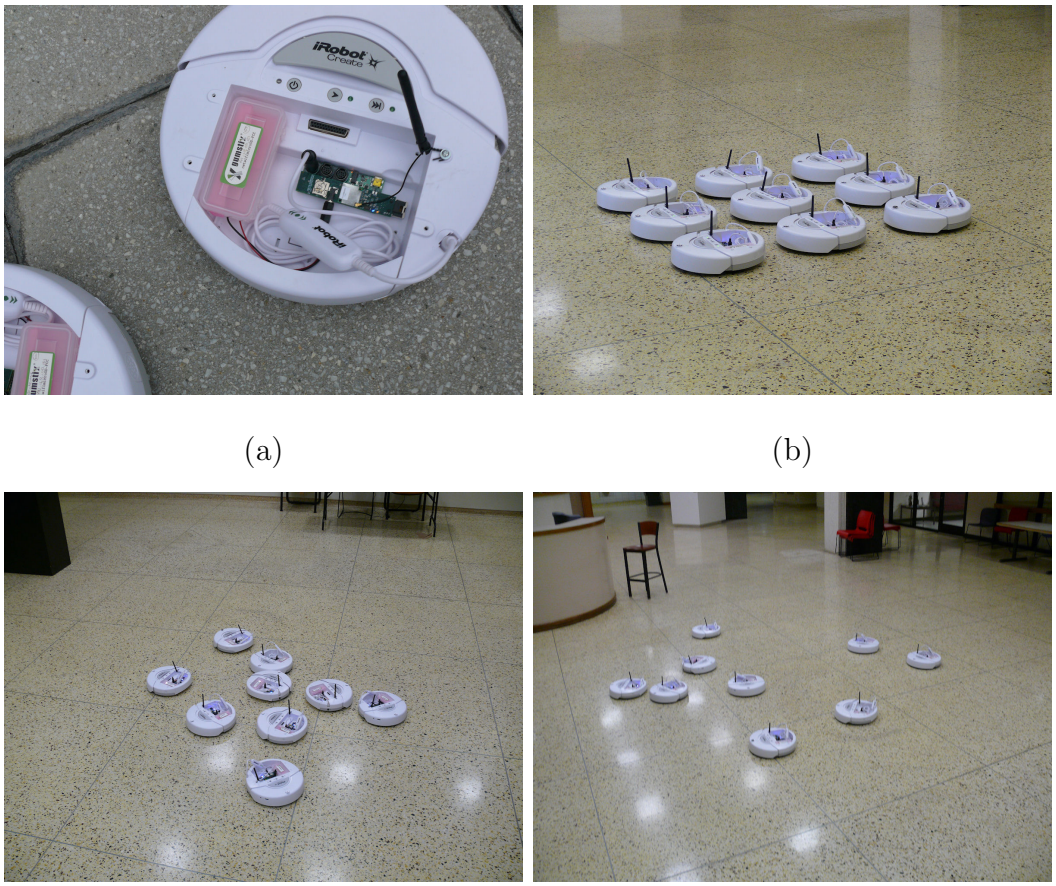
Detection and avoidance of node collision (i.e., two autonomous mobile nodes cannot be at $< x_p, y_p >$ coordinate at the same time) and out-of-boundary area coverages are considered as an add-on for effective GA-based solution evaluation. The TCP packets are exchanged among mobile nodes which are dropped intentionally when the virtual distance between a source and a destination is greater than the configured communication range of R_{com}. This implementation therefore emulates the node

movements for the FPGA, laptop, and desktop units without actually moving them. The current mobility model is developed for Cartesian coordinate system with eight directions and four speeds to move within the geographical terrain. Each mobile node moves independently from all others based on the results of its own GA-based topology control framework.

5.2.2 Testbed Implementation with Small Robotic Units

In this testbed implementation, we used iRobot Create™ robots as the hardware platform which are Roombas™ without vacuum capability. Each robot was then equipped with an onboard Gumstix™ computer. We chose a Gumstix™ because, besides its small size, it is a fully functional Linux-based computer, and consumes very little power (<120mA @5V).

The Gumstix™ computer acts as the brain of the iRobot™. All of our FGA, mobility commands including speed and direction decisions, and wireless communication code run on the Gumstix™ computer. In our testbed implementation, each Gumstix™ runs an identical software code. The connection between the Gumstix™ computers is established automatically via Wi-Fi using UDP broadcast. To study the performance of our GA-based topology control framework with different network densities, we developed this testbed using nine real integrated single-board computers (Gumstix™) and automated robots (iRobots™) as shown in Figs. 5.11 (b)-(d) [114]. The initial positions of the units are displayed in Fig. 5.11 (b). Figs. 5.11 (c) and (d) show two



(a) (b)

(c) (d)

Figure 5.11: Node spreading experiments using *iRobots*™ controlled by the *Gumstix*™ processors with wireless capabilities (a total of 30 time units elapsed)

snapshots of movements of small robotic units by using our FGA.

5.2.3 Testbed Implementation with Virtual Machines

Using VMware™ technology, we implemented a testbed to create a configurable multiplicity emulation environment to study the effectiveness of our FGA. VMware™ virtualization is an abstraction layer decouples the physical hardware from the operating system to deliver greater IT resource utilization and flexibility [112]. It is

possible to run several virtual machines (VMs) simultaneously on the same physical hardware by isolating each one from the physical environment by using VMware virtualization technique. With this capability, a number of VMs can be implemented on a single real machine to act as mobile nodes. Using this technique in our testbed enabled us to scale down the development costs for experiments with a large number of mobile nodes and overcoming the limited availability of computer resources. For simplicity, the autonomous mobile nodes in our testbed are configured with the same capabilities, emulating realistic node mobility and wireless features of MANETs including, but not limited to, autonomous mobility, wireless communication characteristics, and periodic heartbeat messages (periodically broadcast to neighboring nodes within the communication range of R_{com} distance).

This testbed is programmed in C++ and runs in Windows, Linux, and Windows Mobile operating systems. Each computer in our testbed has a configurable number of VMs interconnected by a virtual switch, simplifying and allowing our mobile nodes running GA-based framework experimentations on the single computer as though they run on a real network. Typically, a single computer can handle approximately seven nodes for FGA applications. Our testbed implementation is independent of differences between platforms or whether it is actually running on a physical or a virtual machine. This helps to facilitate a flexible deployment paradigm. All VMs are connected to the network through a virtual switch. Typically, all nodes on this network use the TCP/IP protocol suite, although other communication protocols can be used. A host virtual adapter connects the host computer to the private network used for network

address translation. Each virtual machine and the host have assigned addresses on the private network. This is done through the DHCP server included with the VMware Workstation.

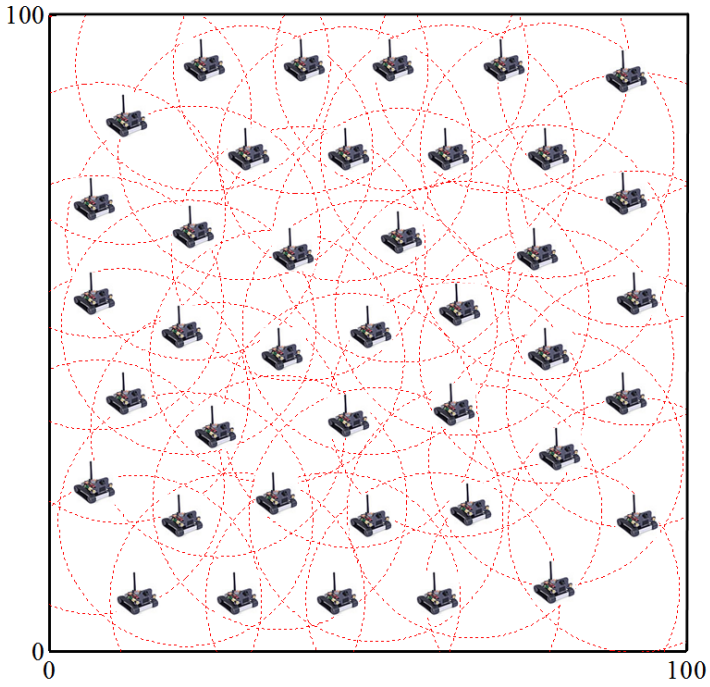


Figure 5.12: A screen shot of final mobile node distribution after 300 time units for mobile nodes spreading experiments using virtual machines

In our testbed, we ran experiments for a MANET with $N = 40$ mobile nodes and $R_{com} = 20$ for a total of $T = 300$ time units in a terrain of 100 x 100 units. At the beginning of each experiment, all mobile nodes are located at the northwest corner of the given area (this is similar to the simulation software experiment given in Fig. 5.2). For simplicity, the mobile nodes in our testbed are configured with the same capabilities. The mobile node distribution after 300 time units is shown in Fig. 5.12. We can observe that, in spite of the lack of global knowledge and a centralized controller, the mobile nodes using our FGA obtain an almost uniform

coverage of the area in a relatively short period of time.

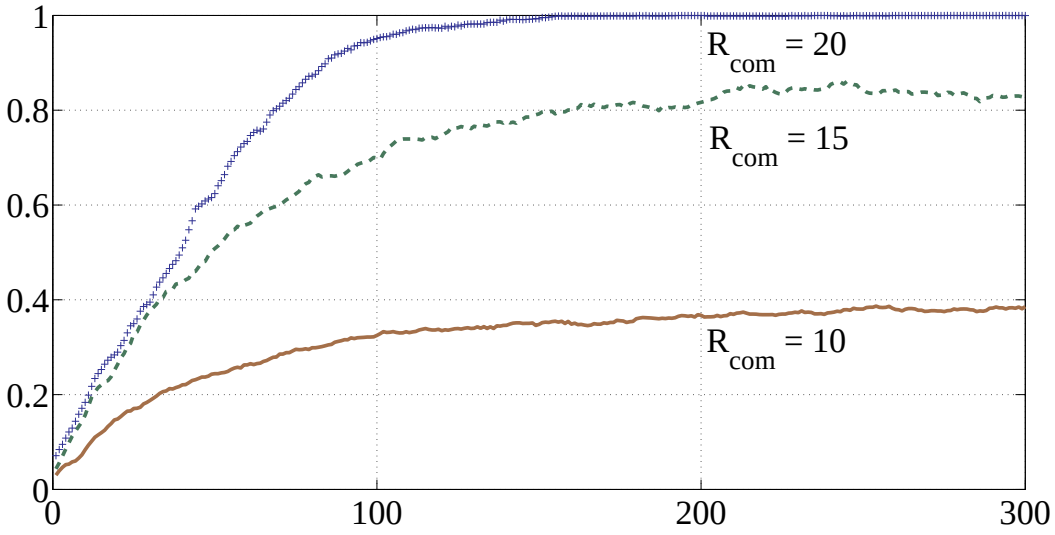
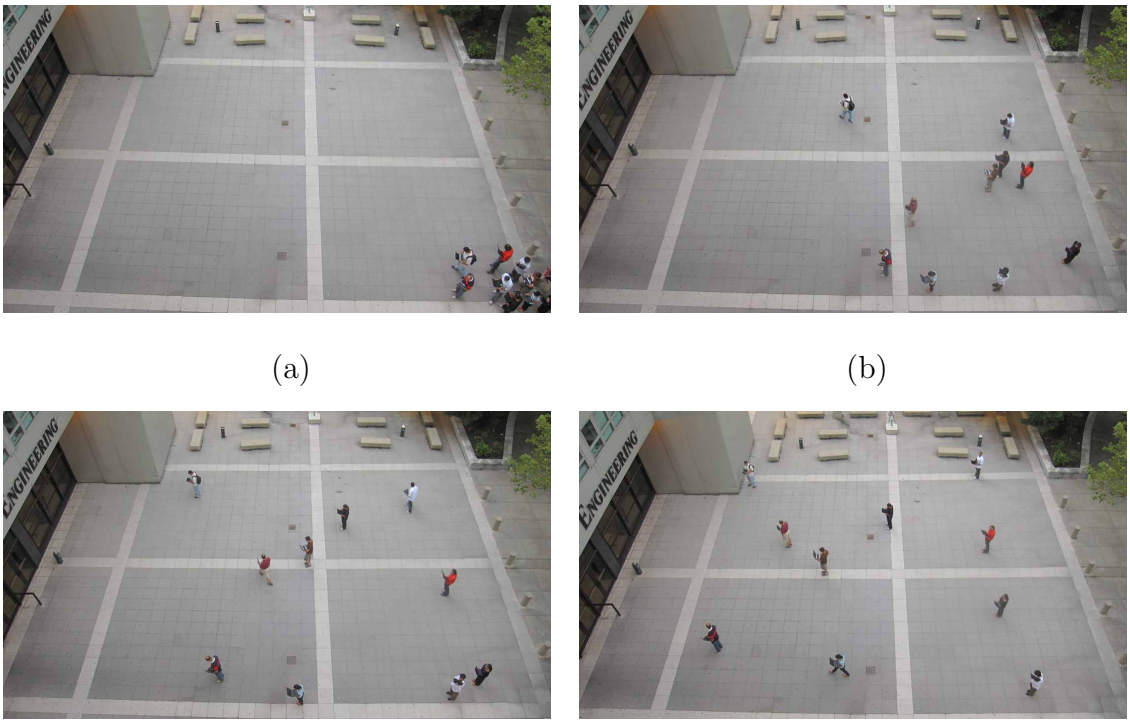


Figure 5.13: Effectiveness in area coverage (A_{eff}) for mobile nodes spreading experiments using virtual machines

In Fig. 5.13, the improvement in the effectiveness in area coverage (A_{eff}) through time is shown for three different communication ranges of R_{com} = 10, 15, and 20 as the mobile nodes perform our GA-based topology control approach in this testbed. As seen from Fig. 5.13, the mobile nodes successfully deploy themselves in an unknown area and achieve 38%, 82%, and 100% area coverage for R_{com} = 10, 15, and 20, respectively. The area coverage reaches and stays at each maximum value at $T \approx$ 110, 160, and 200 time unit for R_{com} = 10, 15, and 20, respectively.

5.2.4 Testbed Implementation with Laptops and PDAs

In this testbed implementation, each laptop and personal digital assistant (PDA) runs an identical our GA-based topology control application to obtain a uniform mobile node separation in an unknown terrain. As opposed to virtual machine testbed pre-



(a) (b)

(c) (d)

Figure 5.14: Node spreading experiments using laptops and PDAs (a total of 30 time units elapsed)

sented in Section 5.2.3, there is only one running instance in each laptop or PDA. In Figs. 5.14 (a)-(d), each student has either a laptop or PDA which provides vocal and visual commands for direction and number of steps (emulating different speeds) using local neighborhood information. Fig. 5.14 (a) shows the initial deployment of nodes for this experiment where all students are placed together at the bottom-right part of the area. The convergence towards a uniform distribution is displayed in Figs. 5.14 (b)-(d) over 30 time units.

Chapter 6

Statistical Analysis of FGA

6.1 Statistical Methods

Statistics is the science that relates data to specific questions of interest [115]. This includes devising methods to gather data based on the question, methods to process, summarize, and display the data. Except for simple examples, experimental data always contain uncertainty. This uncertainty may arise from different sources such as selection of the items to be measured and application of wrong statistical methods. In our research, this uncertainty comes from the non-deterministic nature of the FGA. *Statistical methods* also known as *descriptive statistics* can be used to describe a collection of data of random variables. To search for patterns so that the behavior of our decentralized FGA may be modeled in a way that accounts for randomness and uncertainty in the observations (i.e., collected data). Using this description one can

draw inferences about various parameters of FGA such as number of mobile nodes, communication range, speed, etc.

Statistical model is defined as a set of mathematical equations describing the behavior of an object of study in terms of random variables and their associated probability distributions (i.e., it is a set of probability distributions in a sample space, S) [51]. It is mathematically thought of as a pair (X, P) where X is the set of possible observations (i.e.,$X = \{x_1, x_2, \cdots, x_n\}$, n is the number of observations) and P the set of possible probability distributions on X.

Statistical inference is defined as a method dealing with the problem of inferring properties of an unknown probability distribution from data generated by that probability distribution. The most common type of statistical inference includes approximating the unknown probability distribution by choosing a distribution from a restricted family of distributions [116, 117]. The family of probability distributions is generally specified parametrically including:

- point estimation (involves the use of the data set to calculate a single value which is to serve as a "best guess" for an unknown fixed or random population parameter),
- interval estimation (the use of the data set to calculate an interval of probable values of an unknown population parameter, in contrast to point estimation, which is a single number),
- hypothesis testing (a method of making statistical decisions using experimental

data),

- prediction (a statement or claim that a particular event will occur in the future in more certain terms than a forecast).

Parametric model is one of the most inferential context describing how the observed data are generated. For example, if we have noisy data (x, y) that we think follows the pattern $y = \alpha + \beta \cdot x + \epsilon$, where ϵ represents error, we might want to estimate the parameters $\alpha, \beta,$ and the error ϵ.

All probability distribution functions (pdfs) for continuous random variables have a form of:

$$\frac{1}{cA_r(s)}\Theta(\frac{x-l}{c}) \tag{6.1}$$

where $L_b \leq x \leq L_c$, l is range, $\phi(s)$ is the actual shape of pdfs, $A_r(s)$ is the area under the function (i.e., l represents the location parameter which has the effect of translating the pdfs or x-axis, c is the scale parameter that expands the scale of x-axis, and s is the shape parameter governing the actual shape of the function Θ).

Location parameter, l, is also known as *measures of central tendency*. This statistical metric provides us a measure of what values lie at the center of the pdf. The *mean* and *median* are the most popular measures of location due to their simplicity and ease of estimation. The most common way to show central tendency is mean. The median is often used instead of the mean for asymmetric data because it is closer to the mode and is insensitive to extreme values in the sample [118]. The mean and median are all measures of location $\mu(x)$ of a continuous random variables $X = \{x_1, x_2, \cdots, x_n\}$

in the sense that they satisfy [119]:

$$\forall_{a>0,b}\ \mu(aX+b) = a\mu(x) + b, \mu(-X) = -\mu(X),\ \text{and}\ X \geq 0 \Rightarrow \mu(X) \geq 0 \qquad (6.2)$$

Scale parameter, called *measures of dispersion*, presents information about how "spread out" the values around the location parameter of the random variables. In [120], it is defined as functionals satisfying certain equivariance and order conditions. In other words, let us assume that we have a functional $\tau(F)$ (also denoted by $\tau(X)$ when X is a random variable with distribution F) defined over a sufficiently large class of symmetric distributions which is closed under changes of location and scale. We shall require τ to be non-negative and to satisfy:

$$\tau(aX) = |a|\,\tau(X) \text{ for a>0 and } \tau(X+b) = \tau(X) \text{ for all b} \qquad (6.3)$$

A non-negative functional τ satisfying Eq. 6.3 will be called as a measure of dispersion if and only if it satisfies in addition $\tau(F) \leq \tau(G)$ whenever G is more dispersed than F [120].

Shape parameter is also called as *measures of shape*. There are two common types of measures of shape: skewness and kurtosis. Skewness is a measure of the degree of asymmetry of a probability distribution of random variables. If the left tail (tail at small end of the distribution) is more pronounced than the right tail (tail at the large end of the distribution), the function is said to have a negative skewness. If the

reverse is true, it has a positive skewness. If the two are equal, it has zero skewness. It is calculated by $\gamma_1 = \frac{\mu_3}{\mu_2^{3/2}}$, where μ_i is the i^{th} central moment. Kurtosis is the degree of peakedness of a probability distribution, defined as a normalized form of the fourth central moment of a distribution. It is calculated as $\beta_2 = \frac{\mu_4}{\mu_2^2}$.

Definition 6: *(in [121]) A parametric family of probability distribution is a collection of pdfs on $\Re^n$ indexed by the parameter space Θ, that is a collection of densities of the form $f(x; \theta : \theta \epsilon \Theta)$. In other words, given a parametric family, for each parameter, $\theta \epsilon \Theta$ uniquely specifies a pdf $f(x; \theta)$.*

For better understanding of Def. 6, let us follow an example from [121] and assume that the family of Gaussian (i.e., normal) pdfs has parameter space of $\Theta = \Re_x(0, \infty)$. The parameter is $\theta = (\mu, \sigma^2)$, and probability distribution of Gaussian can be specified by θ in case of an independent and identically-distributed (called i.i.d.) data set $(x_1, x_2, \cdots, x_n)$;

$$f(x; \mu, \sigma) = \frac{1}{\sqrt[n]{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2} \tag{6.4}$$

6.1.1 Parameter Estimation

Parametric model is shown as $P \epsilon \mathbf{P} = \{P_\theta, \theta \epsilon \Theta\}$. The vector θ is a way of labeling the distributions in the model.

Parameterization is formally defined as an onto map from a parameter space as $\Theta \rightarrow P$

called parameterization of $\mathbf{P}$. In other words, the parameterization is known as a way of labeling the probability distributions in the model.

The parameterization changes based on the chosen parameter set and space and, hence, is not unique. The goal is to choose a parameterization in which the components of the parameterization are interpretable in terms of the phenomenon we are trying to measure.

Definition 7: *(in [51]) The parameterization of a probability distribution is identifiable if the mapping from a parameter space Θ to a parameter space P (i.e., $\Theta \to P$) is one-to-one (i.e., if $\theta_1 \neq \theta_2 \Rightarrow P_{\theta_1} \neq P_{\theta_2}$).*

If Def. 7 is not satisfied (i.e., if there exists $\theta_1 \neq \theta_2$ such that $P_{\theta_1} = P_{\theta_2}$), the parameterization is said to be *unidentifiable*.

Parameter is a feature $v(\Theta)$ of Θ (i.e., a map from Θ to another space P). For example, let us assume that we have data set $X = \{x_1, x_2, \cdots, x_n\}$ where $x_i = (y_1, \cdots, y_m)$ and X is i.i.d. with the distribution of $N(\mu, \sigma^2)$. μ (the mean of each x_i), σ^2 (the variance of each x_i), and $E(x_i^2) = \mu^2 + \sigma_2$ are parameters.

Let us consider Bayesian inference for the normal distribution [122]. Suppose we have a data set, $X = \{x_1, x_2, \cdots, x_n\}$ which is i.i.d. and $N(\theta, \sigma^2)$ where σ^2 is known and

our prior on θ is $N(\mu, b^2)$. In this case, the posterior distribution is proportional to:

$$\begin{aligned} f(x—\theta)\pi(\theta) &\propto \\ \left[\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}b} e^{\left[-\frac{1}{2\sigma^2}(x_i-\theta)^2\right]}\right] \left[\frac{1}{\sqrt{2\pi}b} e^{\left[-\frac{1}{2b^2}(\theta-\mu)^2\right]}\right] &\propto \\ e^{-\frac{1}{\sqrt{2\pi}\sigma}\left[\sum_{i=1}^{n} x_i^2 - 2n\overline{x}\theta + n(\theta)^2\right] - \frac{1}{\sqrt{2b^2}}(\theta)^2 - 2\theta\mu + \mu^2} &\propto \\ e^{\left(\theta\left[\frac{n\overline{x}}{\sigma^2}+\frac{\mu}{b^2}\right]+\theta^2\left[-\frac{n}{2\sigma^2}-\frac{1}{2b^2}\right]\right)} &\propto \\ e^{-\frac{1}{2}\left[\theta - \frac{\frac{n\overline{x}}{\sigma^2}+\frac{\mu}{b^2}}{\frac{n}{\sigma^2}+\frac{1}{b}}\right]^2\left[\frac{n}{\sigma^2}+\frac{1}{b^2}\right]} & \end{aligned} \tag{6.5}$$

Thus, the posterior distribution is

$$N\left(\frac{\frac{n\overline{x}}{\sigma^2}+\frac{\mu}{b^2}}{\frac{n}{\sigma^2}+\frac{1}{b^2}}, \frac{1}{\frac{n}{\sigma^2}+\frac{1}{b^2}}\right) \tag{6.6}$$

As a result, the data $(x_1, x_2, \cdots, x_n)$ can be modeled by a random variable:

$$x_i \cong \Theta(\frac{x-l}{c}) \tag{6.7}$$

where the two parameters are the location l and the spread c.

There are three popular methods to estimate location, shape, and scale parameters: the Method of Moments, Order Statistics, and Trimmed Estimates.

6.1.1.1 Methods of Moments

As an estimation of the location parameter l, we may use the arithmetic mean such that

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \text{ analogous to } \mu = \int_{+\infty}^{-\infty} x\Theta(x)dx \tag{6.8}$$

where "analogous to" means that the same kind of operation is performed on a pdf to get the first moment or expected value [51].

If we use the same analogy for the second moment, we get the variance as:

$$s^2 = \frac{1}{n}[\sum_{i=1}^{n}(x_i - \bar{x})^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i - \bar{x})^2] \text{ analogous to } \sigma^2 = \int_{-\infty}^{+\infty}(x-\mu)^2\Theta(x)dx \tag{6.9}$$

Eqns. 6.8 and 6.9 show the methods of moment as a procedure for estimating parameters. In fact, if we know all moments of a distribution, we basically know everything there is to know about the data set and its distribution. The main disadvantage of the methods of moments procedure is to be affected from sample size and extreme samples in the data set which may cause inaccurate results.

6.1.1.2 Order Statistics

Using the order statistics instead of the method of moments gives more accurate estimation parameters. In this method, the data set $x_{(1)}, x_{(2)}, \cdots, x_{(n)}$ is sorted into the increasing order where the use of parenthesis on the subscripts is the standard notation in statistics to indicate a sorted set of values [51]. To estimate of the location

parameter l, the sample median can be used:

$$x_{med} = \frac{1}{2}(x_{(n/2)} - x_{(n/2+1)}) \text{ analogous to } \int_{-\infty}^{x_m} \Theta(x)dx \tag{6.10}$$

To estimate the spread, the interquartile range, which is $x_{0.75n} - x_{0.25n}$, can be used.

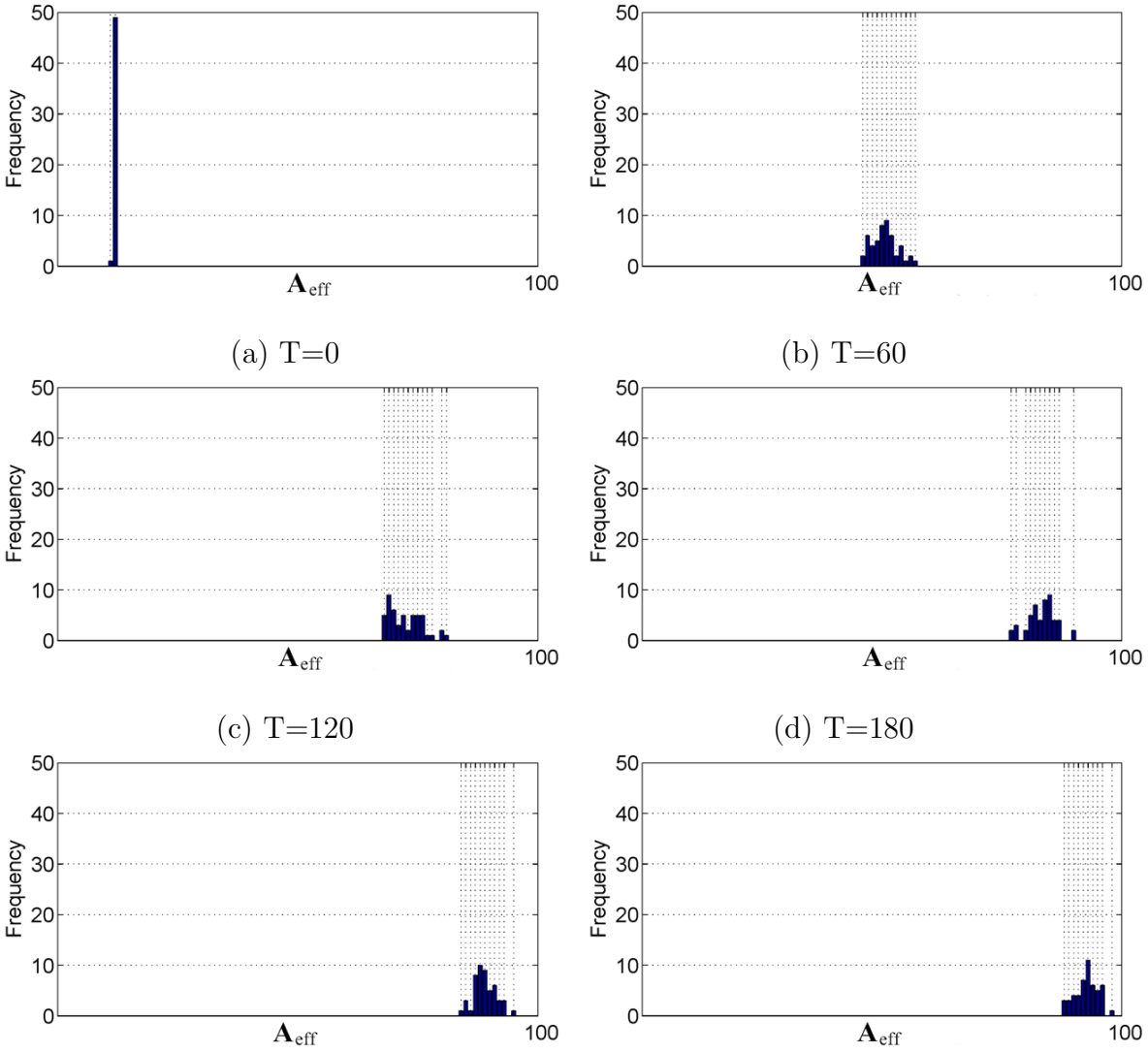
6.1.1.3 Trimmed Estimates

The extreme values may still be problematic to efficiently estimate the parameters even in the order statistics methods [51]. The trimmed estimate method promises more precise results. In this method, the ordered data set $x_{(1)}, x_{(2)}, \cdots, x_{(n)}$ is trimmed from both sides. For example, the 10% trimmed mean is calculated as $\overline{x_{10\%}} = \frac{1}{0.95n}\sum_{i=0.05n}^{0.95n} x_{(i)}$. The spread can be calculated by using the following formula:

$$s_{10\%}^2 = \frac{1.64}{0.9n}\sum_{i=0.05n}^{0.95n}(x_{(i)} - \overline{x_{10\%}})^2 \tag{6.11}$$

6.2 Simulation Experiment Results for Statistical Analysis of our FGA

Statistical analysis of our FGA provides us useful information about its non-deterministic characteristics. We can also predict FGA behavior under different network conditions in term of, for example, the number of mobile nodes and the communication range



(a) T=0 (b) T=60

(c) T=120 (d) T=180

(e) T=240 (f) T=300

Figure 6.1: Frequency of A_{eff} for $N = 100$, $R_{com} = 10$, and $d_{max} = 100$

of a mobile node.

We consider scenarios in which a team of autonomous mobile nodes enter an unknown geographical area without a priori information or a global control unit. Each autonomous node has a limited communication range of R_{com}, hence, can only be aware of its neighbors when running its own GA-based software application. Our main goal is to keep the network fully connected among the mobile nodes while uniformly cov-

ering an unknown geographical terrain. Our decentralized GA-based topology control approach aims to provide each mobile node with a near-optimal number of neighbors so that our FGA can reach its target with the least possible number of nodes.

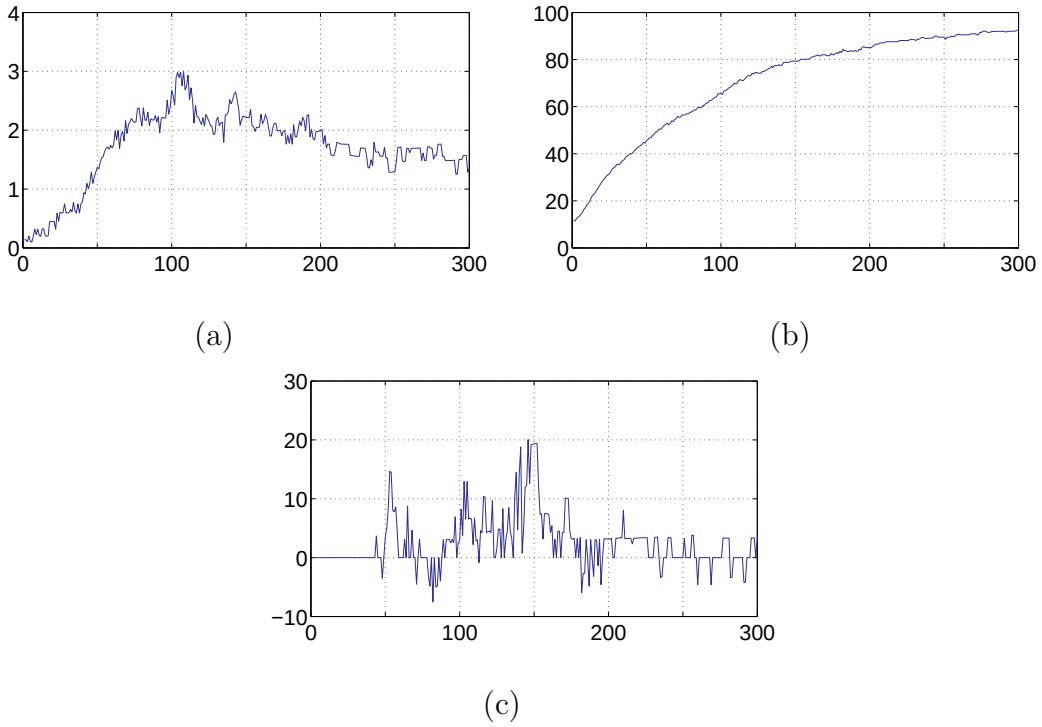


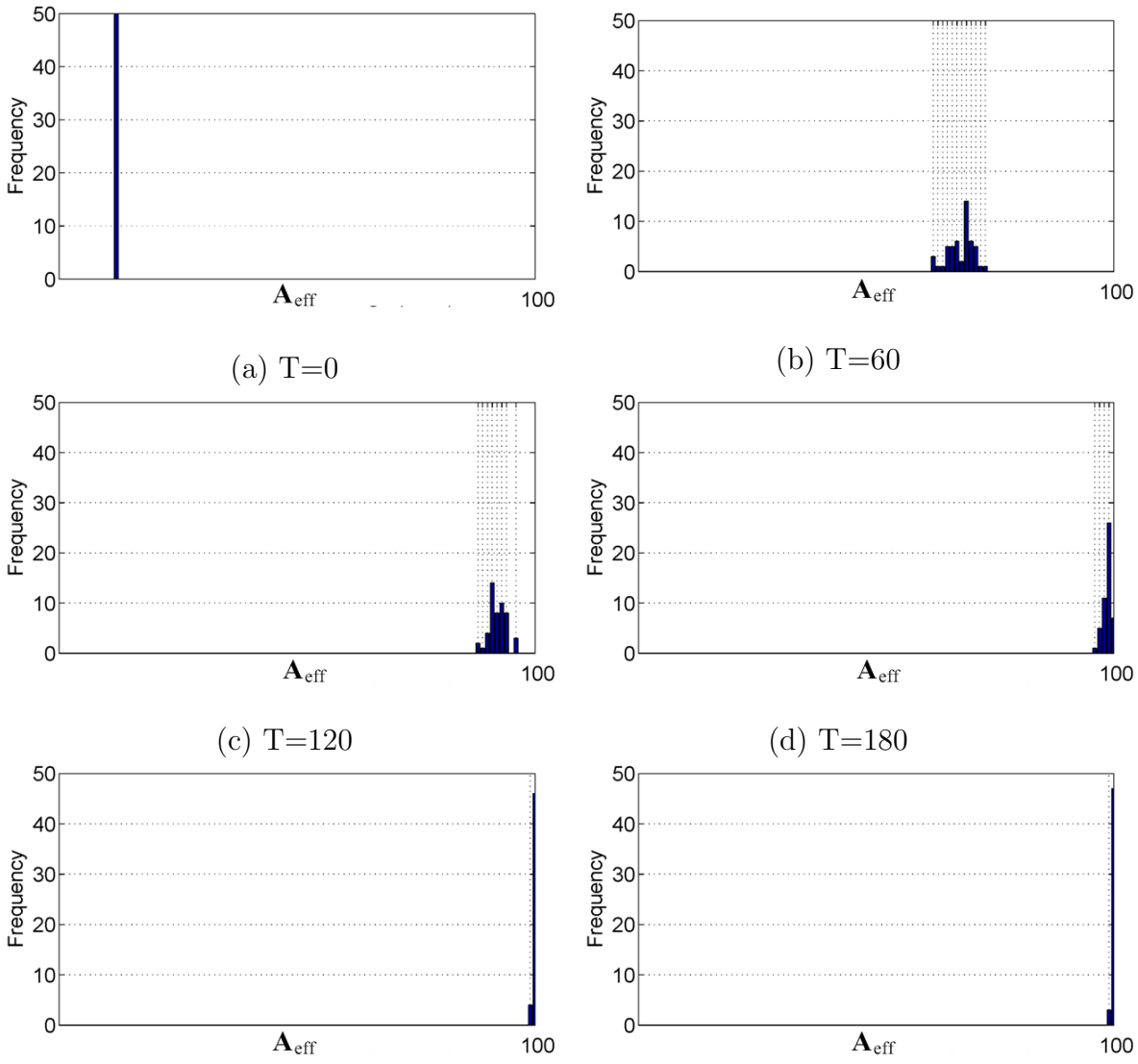
Figure 6.2: Standard deviation, mean, and skew for A_{eff} experiments in Fig. 6.1

The behavior of our FGA may be modeled statistically so that we can extract the patterns in the data collected from simulation experiments. This model then can be a useful guide to predict behavior of our FGA for similar experiments. We have two different experimental setups to statistically analyze effects of the number of mobile nodes and communication range. During the first experimental setup, the data set includes A_{eff} values (see Section 4.3) for different number of autonomous mobile nodes each with a fixed maximum communication distance. At the second experimental setup, we have a fixed number of mobile nodes while varying the communication range of R_{com}. Each simulation experiment was run for $T_{max} = 300$ time

units, and was repeated for 50 times so as to avoid transient results from the natural non-deterministic behavior of our GA-based topology control framework and to collect enough data. At the beginning of each experiment all autonomous mobile nodes were located at the north-west corner of A_{RoI} as shown in Fig. 5.2. The basic statistical model of A_{eff} is obtained using statistical inference method which deals with the problem of inferring properties of an unknown probability distribution from the data set generated by that probability distribution (as explained in Section 6.1). Eqn. 6.1 shows the probability distribution functions for the data set generated by our simulation experiments.

6.2.1 Effects of Autonomous Mobile Node Communication Range on FGA

In this experiment, our goal is to observe the effects of different R_{com} values ($R_{com} = 10, 15$, and 20) on the statistical analysis of our decentralized GA-based topology control framework. Different R_{com} values provide us the ability to analyze our FGA from the statistical inference perspective in various network densities. For larger R_{com} values, more autonomous mobile nodes can communicate with each other, emulating a dense network. Similarly, a small R_{com} implies that the network is sparse since fewer mobile nodes can find neighboring nodes to perform our FGA. In order to get a fair comparison between three different R_{com} values, the hexagonal area was fixed of $10,000$ cells (100x100).



(a) T=0 (b) T=60

(c) T=120 (d) T=180

(e) T=240 (f) T=300

Figure 6.3: Frequency of A_{eff} for $N = 100$, $R_{com} = 15$, and $d_{max} = 100$

Figs. 6.1 (a)-(f) show the frequency of A_{eff} for the total of 100 autonomous mobile nodes ($N = 100$) with maximum communication range of $R_{com} = 10$ for 300 time units. These figures display the improvement in A_{eff} as the experiment proceeds in time and the autonomous mobile nodes perform our FGA. In Fig. 6.1 (a), A_{eff} is very low since all mobile nodes are at their initial positions of north west corner of the area as shown in Fig. 5.2. The frequency is 50 for $T = 0$ meaning that all 50

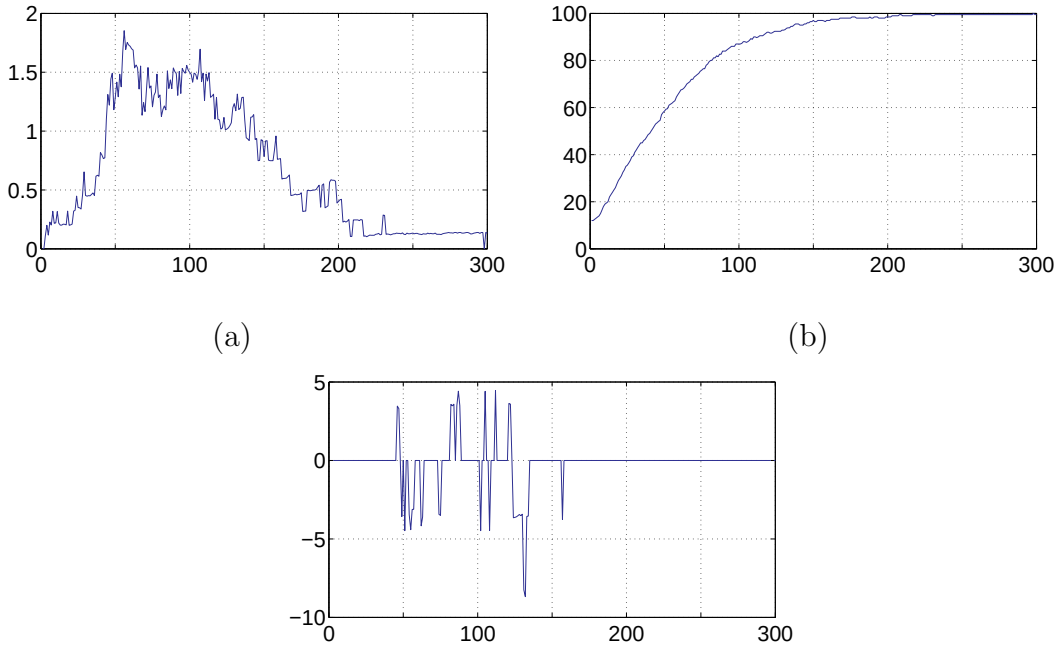
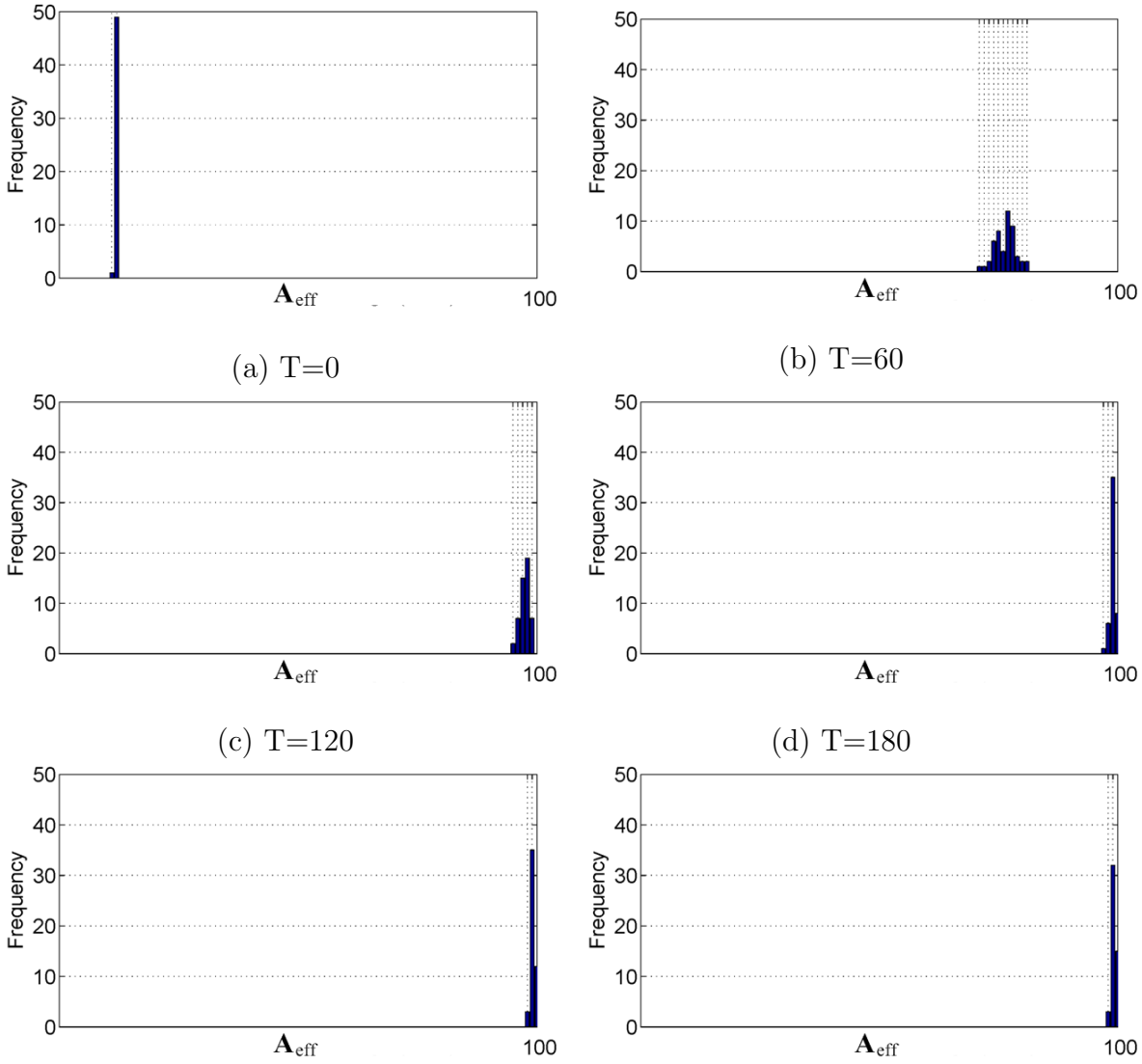


Figure 6.4: Standard deviation, mean, and skew for A_{eff} experiments in Fig. 6.3

experiments start with a low A_{eff} value. As time progresses and the area coverage improves, frequency of higher A_{eff} values are observed. For example, in Fig. 6.1 (c), we see that higher A_{eff} values are achieved by more experiments after 120 time units. Since the maximum communication range is not sufficiently large and there are not enough mobile nodes to cover the geographical area for $N = 100$, A_{eff} value and its frequency do not improve after 240 time units.

For the experiments in Figs. 6.1 (a)-(f), the parameters of scale, location, and skew are shown in Figs. 6.2 (a)-(c), respectively. The standard deviation (σ) reaches its highest value at $T \approx 100$ time units as displayed in Fig. 6.2 (a). However, σ stays high for all of 100 autonomous mobile nodes when the maximum communication is $R_{com} = 10$. Fig. 6.2 (b) illustrates the mean A_{eff} in percentage for our GA-based topology control



(a) T=0 (b) T=60

(c) T=120 (d) T=180

(e) T=240 (f) T=300

Figure 6.5: Frequency of A_{eff} for $N = 100$, $R_{com} = 20$, and $d_{max} = 100$

framework. As seen in Fig. 6.2 (c), the skew highly oscillates. In fact, the distribution shown in Figs. 6.1 (a)-(d) do not show resemblance to Gaussian distribution, we obtain large oscillations in Figs. 6.2 (a) and (c). We see that Figs. 6.1 (e) and (f) converge towards a Gaussian distribution. The effect of this convergence can be seen in Figs. 6.2 (a)-(c) from $T = 200$ to $T = 300$ time units. We can conclude that 100 mobile nodes with maximum communication range of $R_{com} = 10$ are not enough

to uniformly cover the entire terrain 100x100. However, our FGA performed well to obtain the maximum coverage even in this case.

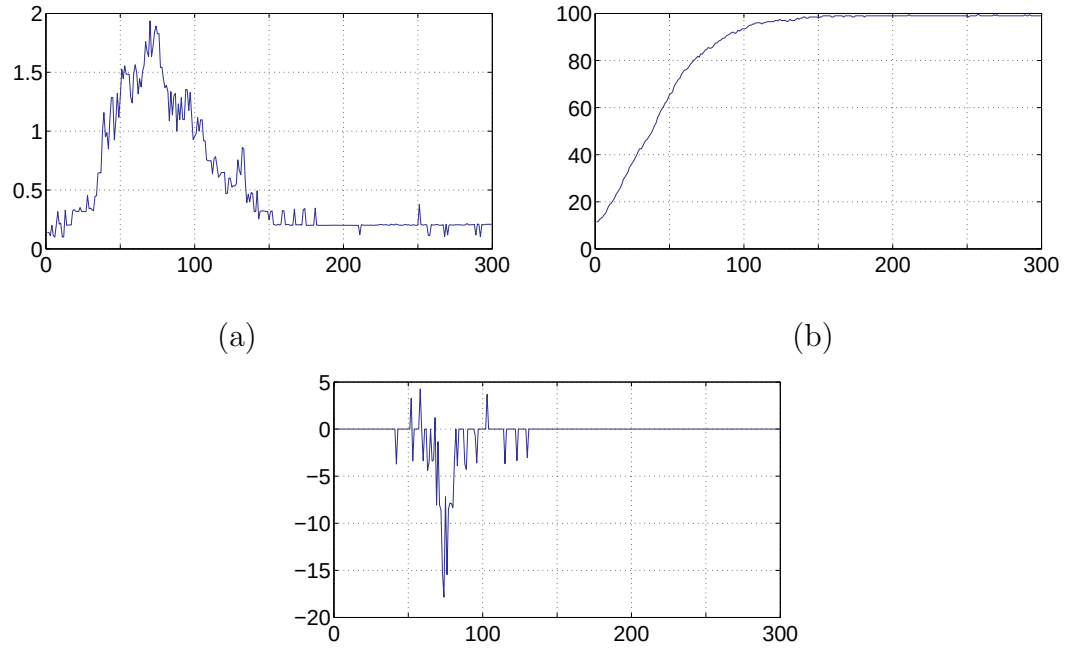


Figure 6.6: Standard deviation, mean, and skew for A_{eff} experiments in Fig. 6.5

The frequency of A_{eff} for $R_{com} = 15$ and $N = 100$ mobile nodes for 300 time units are displayed in Figs. 6.3 (a)-(f). The autonomous mobile nodes reach the maximum A_{eff} at $T = 180$ time units since the frequency of high A_{eff} values are observed in Fig. 6.3 (d). The figures show a Gaussian distribution shape after 50 runs. Figs. 6.4 (a)-(c) support the normal distribution observed in Figs. 6.3 (b)-(c) (i.e., from $T = 120$ to $T = 180$ time units). After $T = 180$ time units, the frequency values close to 100%, for which A_{eff} become dominant, results in convergence to a uniform distribution. We observe from Figs. 6.4 (a)-(c) that after $T = 180$ time units, σ is low (implying small changes in locations) with zero skew values resulting in a single value of in the frequency of A_{eff} for $R_{com} = 15$. In other words, our FGA spreads

the autonomous mobile nodes to cover the entire region. Compared to the case of $R_{com} = 10$, we see a much improved set of results for $R_{com} = 15$ since there are more mobile nodes within the increased communication range yields collecting more neighborhood information and better area coverage.

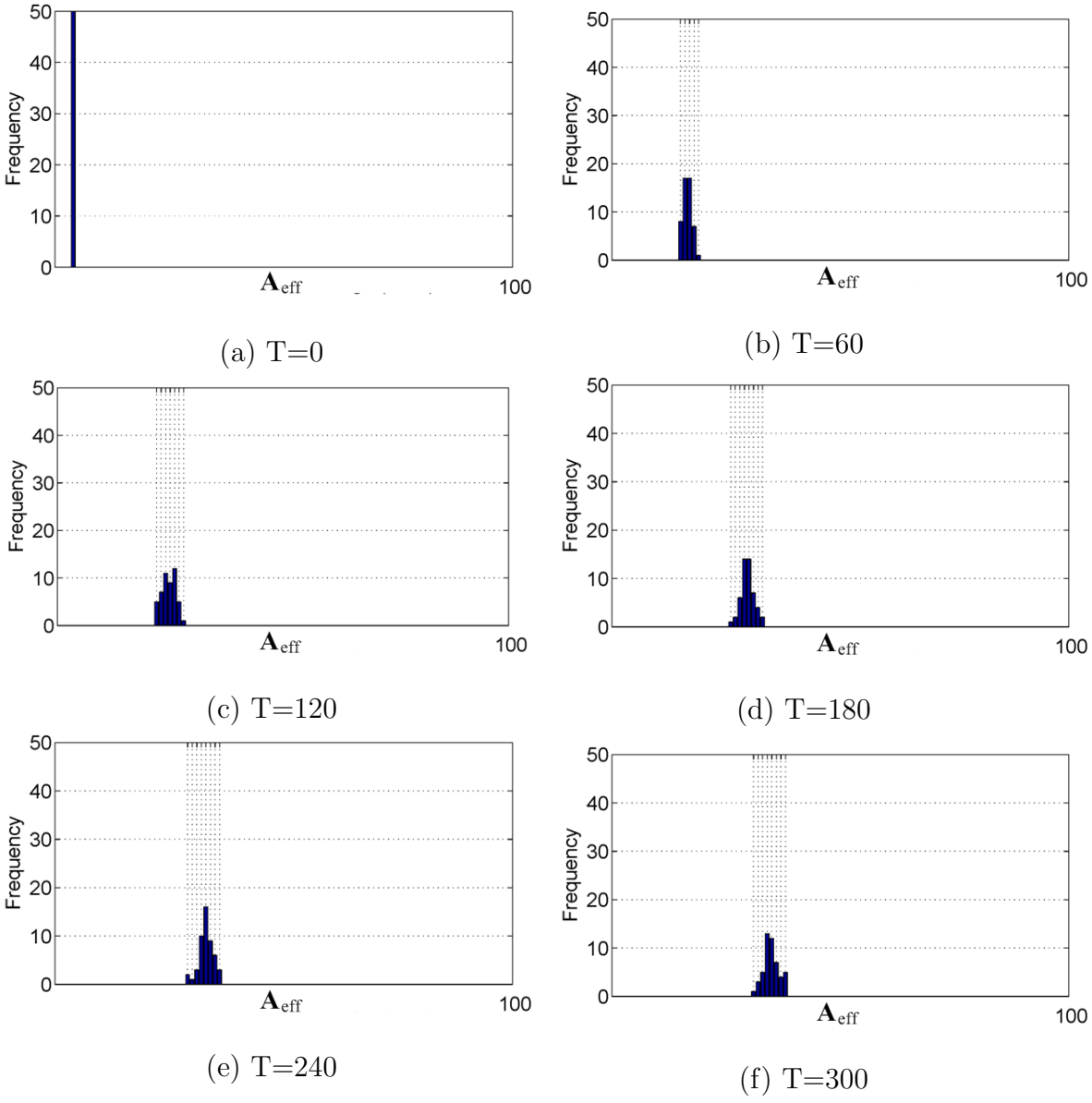


Figure 6.7: Frequency of A_{eff} for $N = 100$, $R_{com} = 10$, and $d_{max} = 200$

Figs. 6.5 (a)-(f) show $N = 100$ for $R_{com} = 20$ from $T = 0$ to $T = 300$ time units.

The autonomous mobile nodes spread over the entire A_{RoI} in less than $T = 180$ time units. The node distribution is closer to a Gaussian distribution in Figs. 6.5 (b)-(c). After $T = 180$ time units, the entire region is covered and the frequency of A_{eff} is approximately 100% which is an indication of the uniform distribution. Figs. 6.6 (a)-(c) support this uniform A_{eff} distribution after $T = 180$ time units.

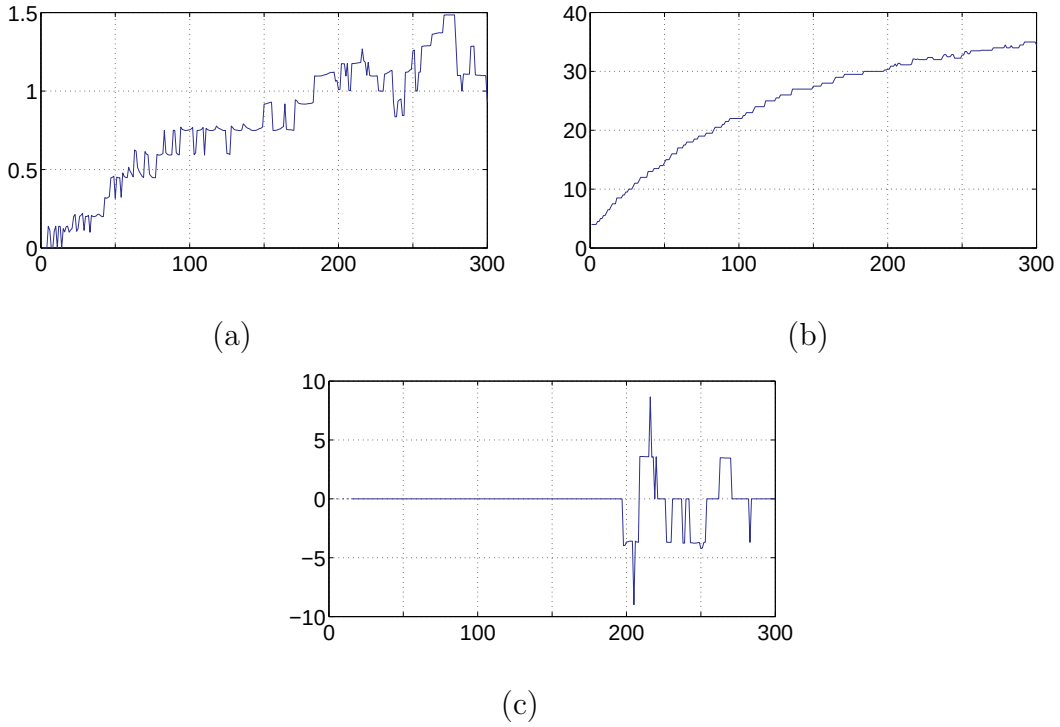


Figure 6.8: Standard deviation, mean, and skew for A_{eff} experiments in Fig. 6.7

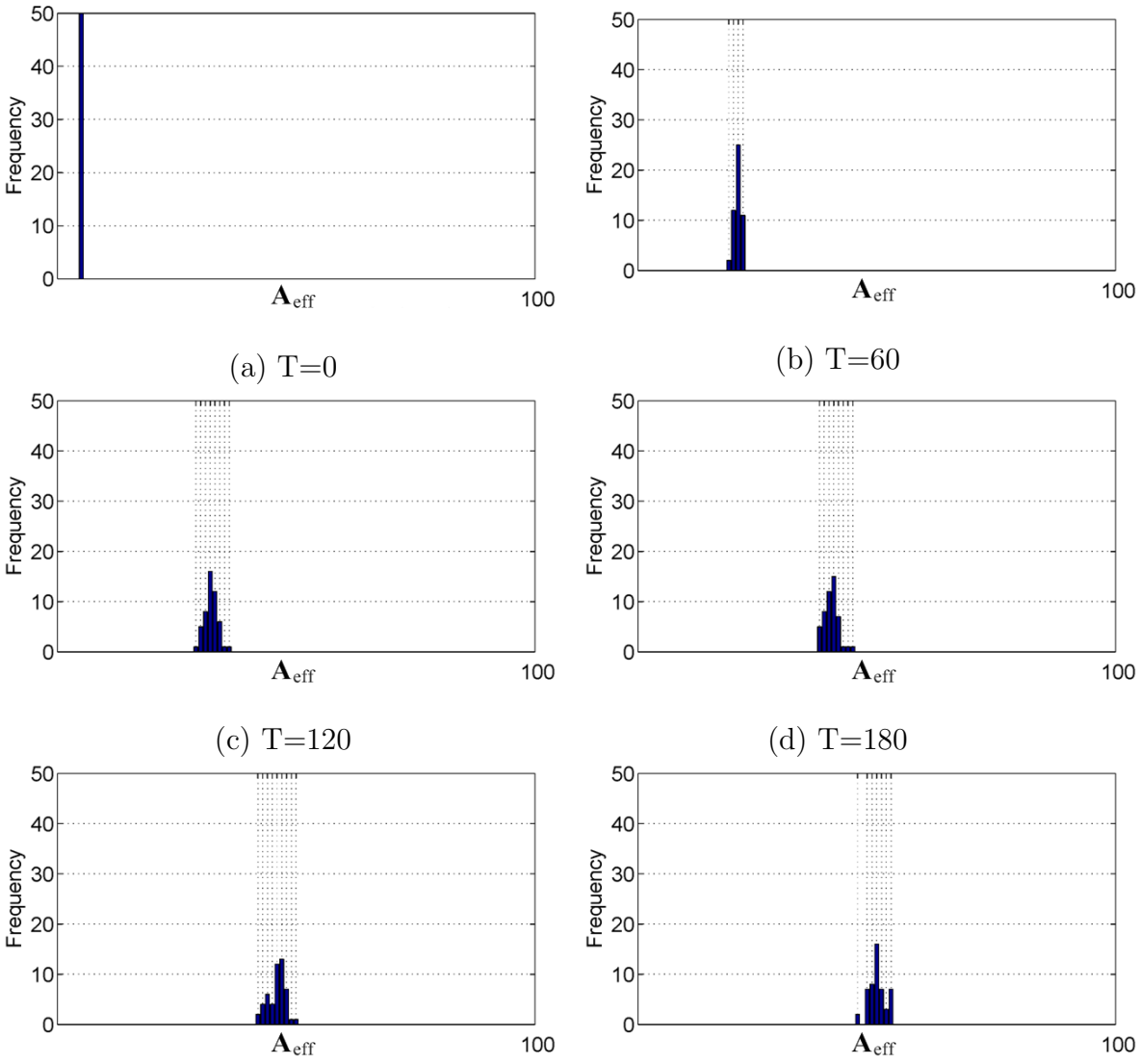
The goal of these experiments was to observe the effects of different communication ranges over the performance of our FGA. Since R_{com} has the highest values in Figs. 6.5 (a)-(f), the best results are observed for these experiments compared to the cases shown in Figs. 6.1 (a)-(f) and Fig. 6.3 (a)-(f). In other words, the mobile nodes converge towards a uniform distribution faster faster when R_{com} is higher. This result is supported by small fluctuation in scale and skew parameters when R_{com} increases.

These results are expected since increasing the communication range of R_{com} results in communicating with more neighbors, collecting more local information, and controlling wider area. However, longer range of communication capability also causes more battery consumption than smaller R_{com} values. It shows the importance of selecting the right communication range value as an input for our research and similar applications. Therefore, one has to select the correct balance between the battery consumption and convergence speed for a given application.

6.2.2 Effects of Network Size on FGA

The aim of this experimental setup is to observe the effects of number of autonomous mobile nodes ($N = 100, 200,$ and 300) over convergence speed and area coverage and to extract the statistical behavior of FGA while each mobile node has a fixed communication range. The chosen hexagonal area consists of 40,000 cells (200x200) whereas R_{com} is set to 10 for all cases. The geographical area is wider than the experiments described in Section 6.2.1 in order to make better observation using a larger number of mobile nodes. Each experiment is repeated 50 times with the same initial condition as shown in Fig. 5.2. The autonomous mobile node speed, movement direction, and locations are the same at the beginning of each experiment. The stochastic nature of GA-based framework however generates different results for each run.

Figs. 6.7 (a)-(f) show the frequency of A_{eff} for a total of 100 mobile nodes ($N = 100$)



(a) T=0 (b) T=60

(c) T=120 (d) T=180

(e) T=240 (f) T=300

Figure 6.9: Frequency of A_{eff} for $N = 200$, $R_{com} = 10$, and $d_{max} = 200$

from $T = 0$ to $T = 300$ time units. In Fig. 6.7 (a), A_{eff} is low since all mobile nodes are at their initial positions of north west corner of the area as shown in Fig. 5.2. The frequency is 50 for $T = 0$ meaning that all 50 experiments start with a this low A_{eff} value. As time progresses and coverage improves, frequency of higher A_{eff} values are observed. For example, in Fig. 6.7 (c), we see that higher A_{eff} values are achieved by more experiments at $T = 120$ time units. Since there are not enough mobile nodes to

cover the geographical area for $N = 100$, A_{eff} value and its frequency do not improve from $T = 120$ to $T = 300$ time units.

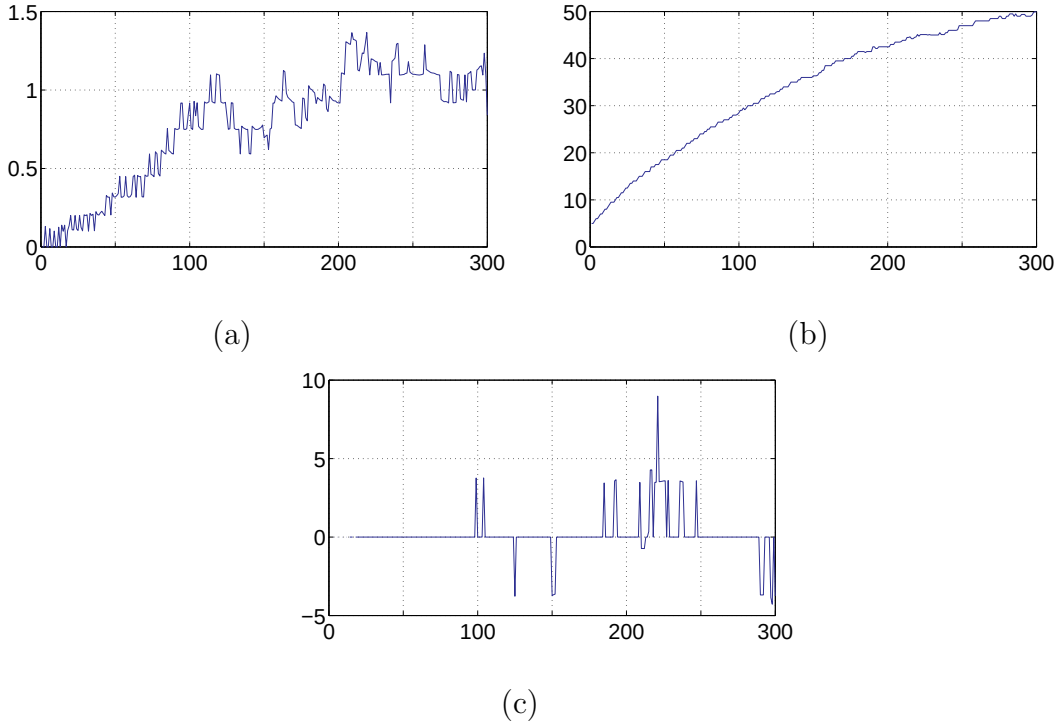
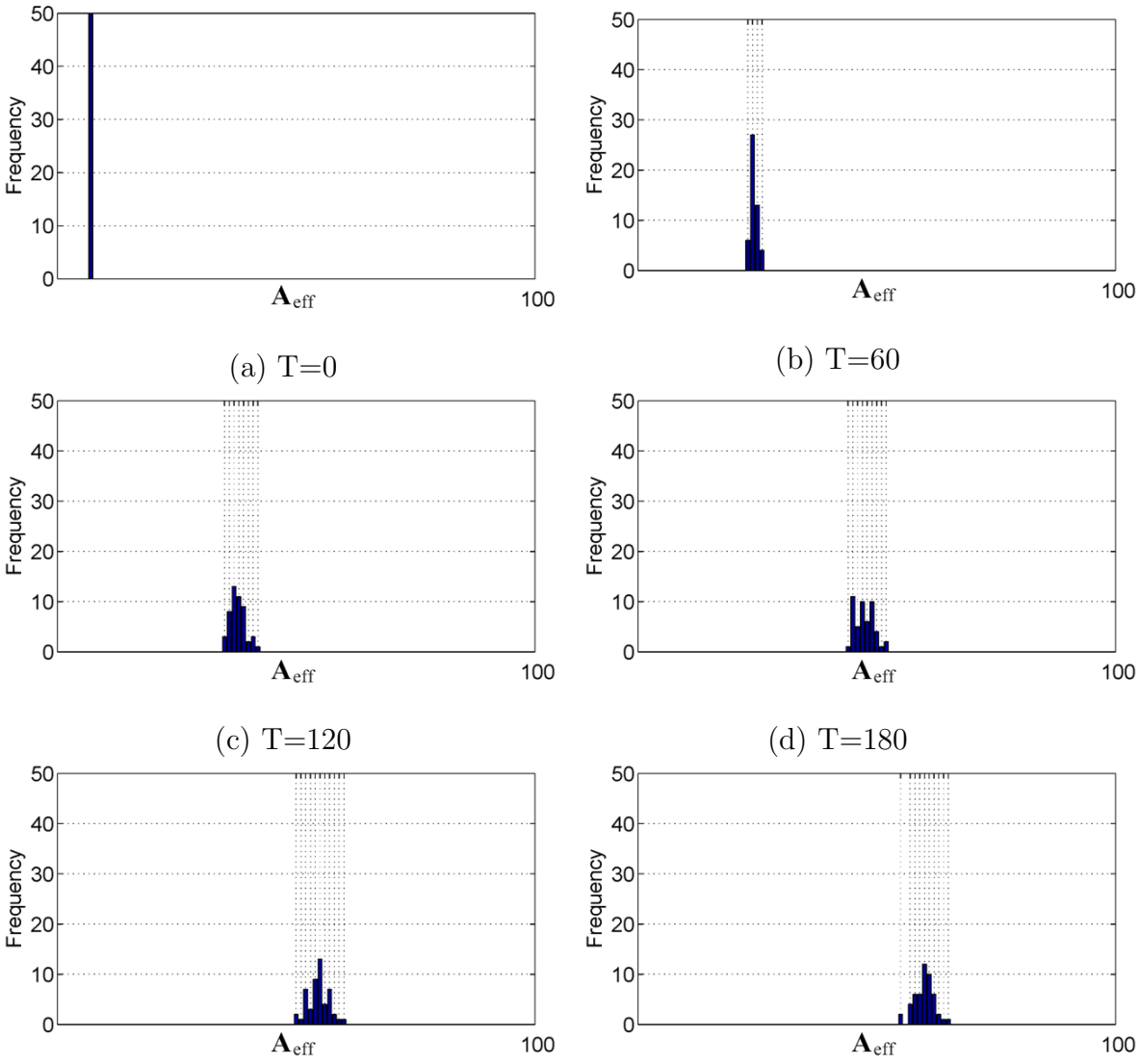


Figure 6.10: Standard deviation, mean, and skew for A_{eff} experiments in Fig. 6.9

For the experiments in Figs. 6.7 (a)-(f), the parameters of scale, location, and skew are shown in Figs. 6.8 (a)-(c), respectively. The standard deviation continuously increases in Fig. 6.8 (a). The reason for the perpetual growth from $T = 0$ to $T = 300$ time units is both small number of autonomous mobile nodes and the short range of communication capability, $R_{com} = 10$. Fig. 6.8 (b) represents the mean A_{eff} for our FGA. Fig. 6.8 (c) illustrates skew after $T = 200$ time units where the autonomous mobile nodes spread and cover reach their highest coverage. Figs. 6.7 (d)-(f) show a distribution similar to Gaussian for A_{eff}.

The frequency of A_{eff} for $N = 200$ mobile nodes from $T = 0$ to $T = 300$ time units are



(a) T=0 (b) T=60

(c) T=120 (d) T=180

(e) T=240 (f) T=300

Figure 6.11: Frequency of A_{eff} for $N = 300$, $R_{com} = 10$, and $d_{max} = 200$

displayed in Figs. 6.9 (a)-(f). The autonomous mobile nodes reach their highest area coverage at $T = 240$ time unit (hence the frequency of high A_{eff} values are observed in Fig. 6.9 (e)). The figures show a distribution similar to a normal distribution after 50 runs. Figs. 6.10 (a)-(c) support the normal distribution in Figs. 6.9 (a)-(f).

Figs. 6.11 (a)-(f) illustrate the frequency of A_{eff} for $N = 300$ to $T = 300$ time

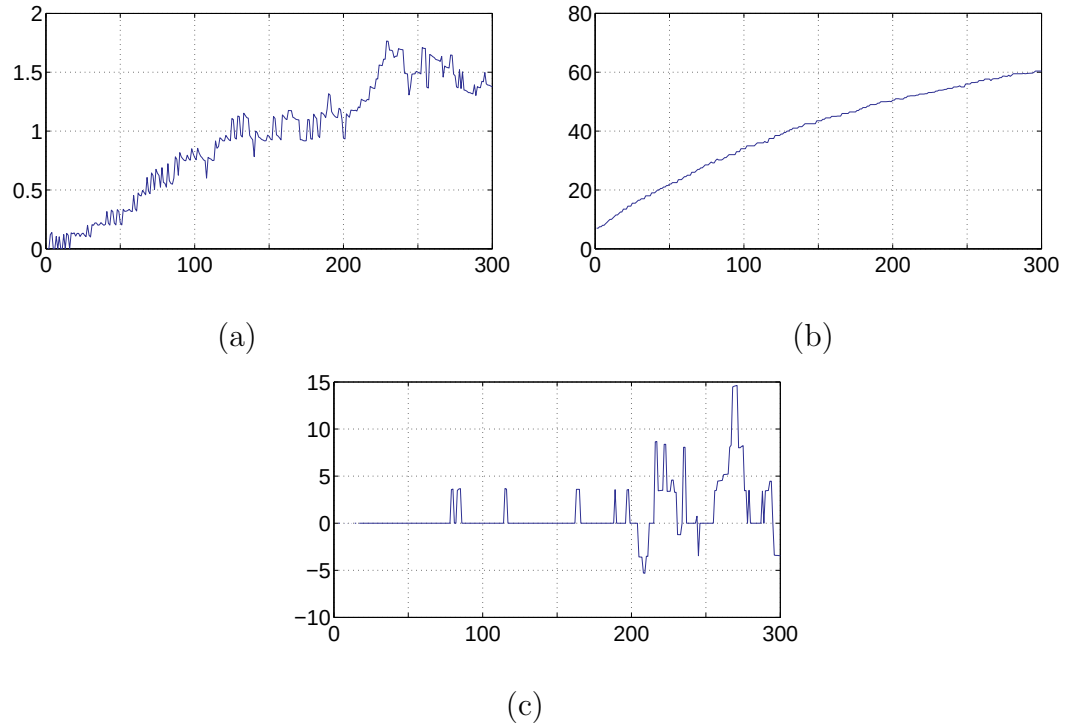


Figure 6.12: Standard deviation, mean, and skew for A_{eff} experiments in Fig. 6.11

units. The autonomous mobile nodes spread over the unknown terrain; however, they cannot cover the entire area since the maximum communication range is not large enough and there is not enough number of mobile nodes. The frequency of A_{eff} shows a Gaussian distribution shape in Figs. 6.11 (a)-(f). This distribution indicates a satisfactory performance of our FGA. Furthermore, we can claim that there is not enough mobile nodes to cover the given terrain since Figs. 6.11 (a)-(f) show Gaussian distribution. Figs 6.12 (a)-(c) also indicate the same distribution. σ reaches its maximum at $T = 230$ time units and it stays at this maximum value in Fig. 6.12 (a). 60% of the entire region is covered by the mobile agents at the end of experiment after 300 time units as seen in Fig. 6.12 (b). Compared to the cases of $N = 100$ and 200, we see a much improved set of results for $N = 300$ since there are more mobile nodes to spread over the area.

Chapter 7

Convergence Analysis of FGA

7.1 Markov Chain

A *stochastic process* can be defined as a collection of random variables X_t indexed by time. A sequence of X_t of random variables is called a *discrete time stochastic process* if $t \geq 0$ and X_t takes values in a finite set $S = 0, 1, \cdots, N$ [123]. The finite set S is called a *state space* and possible values of X_t are called the *states* of the system. The discrete time stochastic process describes a system which traverses from state to state at each time instance. Eqn. 7.1 shows the memoryless nature of a single state (i.e. if the process at time t_1 is at state $i \, \epsilon \, S$, the process at time $t_1 + 1$ a transition to state $j \, \epsilon \, S$).

$$P(X_0 = x_0, \cdots, X_{n-1} = x_{n-1}) = \prod_{k=0}^{n-1} P(X_k = x_k | X_{k-1} = x_{k-1}, \cdots, X_0 = x_0) \quad (7.1)$$

A stochastic process has the *Markov property* if the conditional probability distribution of future states of the stochastic process depends only upon the present state and not the past as shown in Eq. 7.2. A stochastic process that satisfies the Markov property is known as a *Markov process*.

$$P(X_k = x_k | X_{k-1} = x_{k-1}, \cdots, X_0 = x_0) = P(X_k = x_k | X_{k-1} = x_{k-1}) \tag{7.2}$$

It is important to note that Eq. 7.2 holds for all states of the system $k \, \epsilon \, S$.

A Markov process called a *Markov chain* is defined by:

- initial probability distribution $\nu(x) = P(X_0 = x)$
- transition probabilities from state i to state j (i, j, ϵ, S) p_{ij}

Using these two properties, Eq. 7.2 can be written as:

$$P(X_0 = x_0, X_1 = x_1, \cdots, X_{n-1} = x_{n-1}) =$$

$$\nu(x_0) p(x_0, x_1) p(x_1, x_2) \cdots p(x_{n-2}, x_{n-1}) \tag{7.3}$$

where $\nu(x_0)$ is the initial probability distribution and $p(x_i, x_j)$ is the transition probability from state i to state j $(i, j \, \epsilon \, S)$.

A Markov chain is a suitable probability model for certain systems where the observation at a given time maps to the category into which an individual falls. This mapping is done by using a *stochastic matrix* (i.e., *transition matrix*) which contains

the transition probabilities of a Markov chain over a finite state space S. If p_{ij} shows the probability value of moving from state i to state j (where $i, j \epsilon S$) in one time unit, the transition matrix P is given by using p_{ij} as an element at i^{th} row and j^{th} column. For example:

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nn} \end{pmatrix}$$

The *transition matrix* P includes all possible transition probabilities between states and $p(x_i, x_j) = p_{ij} \geq 0$ and $\sum_{k \epsilon S} p_{ik} = 1$, where $i, j \epsilon S$.

For an arbitrary distribution of S denoted as ν to transition to another distribution μ, the transition matrix is applied such that $\nu P = \mu$. It is important to note that the Markov chain exhibits the *memoryless* characteristic where no knowledge of past states is needed to determine the future behavior of the system. It is only important to know the current state of the system and apply the transition matrix to determine future behavior. Assuming that the system transitions through time denoted by X_t (for $t = 1, 2, \cdots$) the memoryless quality can be represented by

$$Pr(X_t = \mu_t | X_0 = \mu_0, \cdots, X_{n-1} = \mu_{n-1}) = Pr(X_n = \mu_n | X_{n-1} = \mu_{n-1}) \qquad (7.4)$$

FGA, like all GA-based approaches, uses different sets of chromosomes in every population and, therefore, can be modeled by Markov chain since the state of population at time $t+1$ depends only the state of population at time t. As the genetic operators (see Section 3.1.2) that create the population at time $t+1$ out of the population at time t depend on several chance events, like the drawing of one certain string to the mating pool or the signal t mutate one certain bit, the dependence of the state of population at time $t+1$ on the state of population at time t is stochastic [97]. Our FGA is run by each mobile node as a topology control mechanism to decide its next speed and movement direction. Thus, each autonomous mobile node's movement decision is stochastic. These considerations imply that an exact description of the behavior of autonomous mobile nodes in a MANET has to use a Markov chain where the states of the chain are the possible states of ourGA-based topology control framework.

7.2 Convergence Analysis of Homogeneous Markov Chain Model of our FGA

A Markov chain is called *homogeneous* if the transition matrix is stationary with respect to time (i.e. the transition probabilities from state i to state j $(i, j \epsilon S)$ p_{ij} do not change) [124]. The homogeneous Markov chain is a popular and often effective model to describe *state*-based systems and discrete stochastic systems that may be modeled as having a set of n *states*.

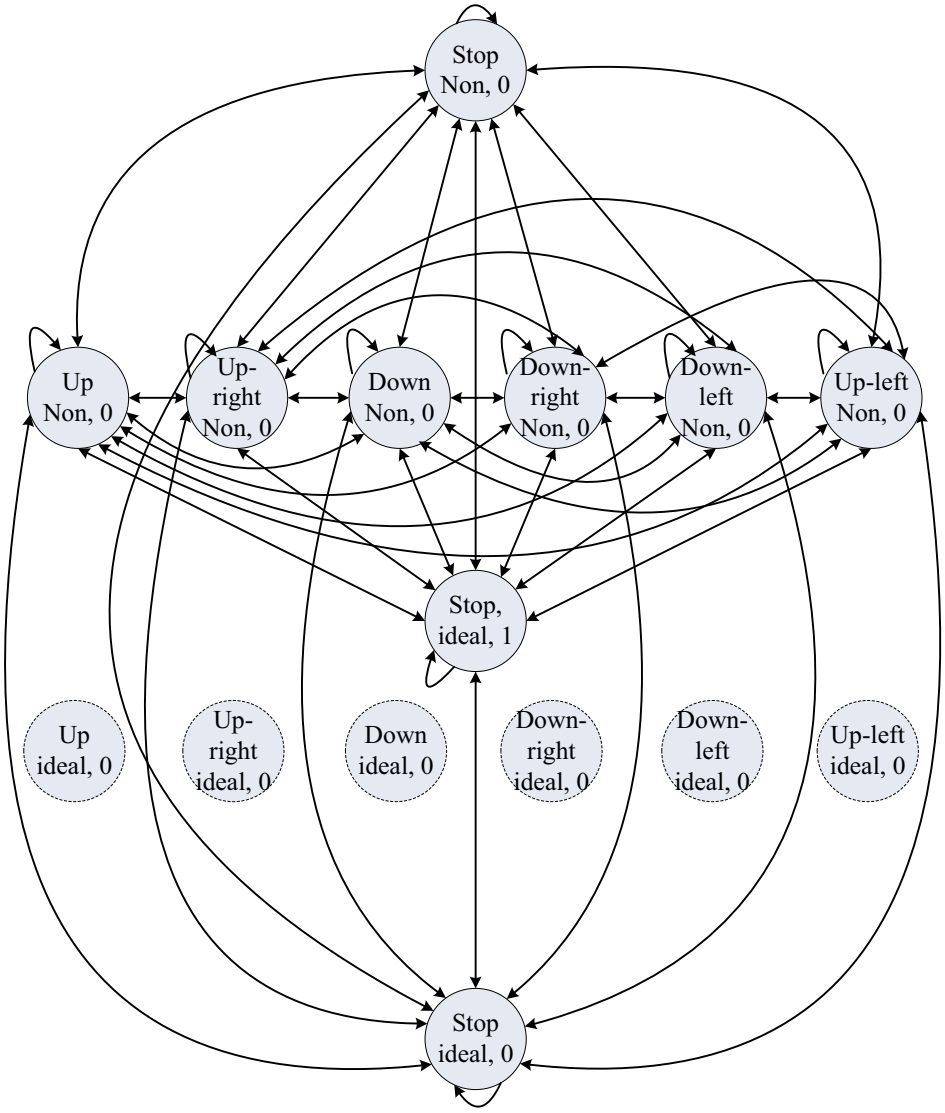


Figure 7.1: A Markov chain model for our FGA (each state is connected to each other but not shown for simplicity)

Probabilistic changes of the genes within the population at time $t + 1$ (P_{t+1}) are captured by the transition matrix in the Markov kernel since P_{t+1} only depends on the population at time t (P_t). The genetic operators that create P_{t+1} from P_t hinge upon several chance events, such as the crossover point, or the random mutation in a particular bit. This makes the Markov chain a perfect tool to analyze the convergence properties including convergence speed of GA-based algorithms. One of the most

important aspects of modeling a GA using a Markov chain is to define a realistic number of states. GAs use different sets of chromosomes in every population. The number of states in the Markov model are proportioned to the mapping from the solution space of S to the Markov chain states.

The behavior of autonomous mobile nodes running our FGA as software agents can be modeled as a finite homogeneous Markov chain, called hMC_{FGA}. In our approach, the values of hMC_{FGA} transition matrix are determined through estimation instead of exact representation as theoretically represented in [69]. Using the direct approach for our FGA quickly becomes unwieldy and computationally unfeasible since the number of states grow exponentially with the chromosome length (see Section 4.1). To keep the analysis of the behavior for FGA simple, the characteristics of an autonomous mobile node have been compressed to the following parameters: the mobile node's speed (moving or immobile), its fitness (high or low), and direction (up, up-left, left, down-left, down, down-right, right, or up-right based on the hexagonal lattice explained in Section 3.3) as shown in Fig. 7.1. Without this simplification, the number of the states in this model would be prohibitively large. For example, using fitness resolution of 10^5, 10 different speeds, and six movement directions would yield 10^7 states.

In this dissertation, the estimation of an autonomous mobile node's state is found by using the empirical probability of traversing from one state to another obtained experimentally (i.e. the relative frequency of moving from one state to the other is recorded while running the FGA at each time instant). As the FGA determines the

mobile node's speed and movement direction using local neighborhood information including neighbors and obstacles, we assume that an autonomous mobile node traverses from one state to another in our Markov chain model shown in Fig. 7.1. By conducting a large number of experiments, statical anomalies can be smoothed out resulting in a model that closely approximates the behavior of a mobile node in a MANET.

The number of neighbors that a given node has is an important metric defined in the Markov model of our FGA. The number of neighbors for a given node N_i is defined as $d(N_i)$ (node degree). The ideal number of neighbors that creates the maximum amount of network coverage is defined as $\bar{N}$ (see Section 3.3.1) [100]. The *ideal* state in hMC_{FGA} occurs when $\bar{N} - 1 \leq d(N_i) \leq \bar{N} + 1$. All values outside of the bounds are considered *non-ideal* (Fig. 7.1).

As shown in Fig. 7.1, the reduced hMC_{FGA} is defined with 15 states. The states can first be divided into two categories, either moving or stationary (i.e., immobile). Any movement is described in 12 states, half with an ideal number of neighbors ($\bar{N} - 1 \leq d(N_i) \leq \bar{N} + 1$) and the other half with a non-ideal number of neighbors ($d(N_i) < \bar{N} - 1$ or $d(N_i) > \bar{N} + 1$). Obviously an autonomous mobile node with a non-ideal number of neighbors is moving probabilistically in search of the correct number of neighbors and a low external force value (i.e. high fitness). An autonomous mobile node that has the desired number of neighbors may suffer from poor orientation with neighbors and also continue to search for a location with lower virtual force. The three stop states (a mobile node in this state is immobile) include a non-ideal number

of neighbors and low fitness (**Stop, Non, 0**), an ideal number of neighbors (see Section 3.3.1) and low fitness (**Stop, ideal, 0**) and an ideal number of neighbors and high fitness (**Stop, ideal, 1**). The first two states often occur when a mobile node is physically trapped for a given instant in time by obstacles or other mobile nodes around it. Note that these situations are non-lasting and the nature of our FGA will encourage the mobile node to move to a better location as quickly as possible. In (**Stop, ideal, 1**) state, our GA-based topology control framework will keep the mobile node stationary as long as the condition persists. This is an important characteristic for our FGA because the network has the ability to stabilize and conserve power when a good configuration has been reached as well as reconfigure itself when network conditions deteriorate.

As explained earlier, a homogeneous Markov chain is characterized by having a transition matrix of P that is equal at every time step for a finite state space S, that has an initial distribution ν. A *metrically transitive* (i.e., *ergodic*) Markov chain is characterized by having both *irreducible* and *aperiodic* properties, as explained below.

***Definition* 8:** *States i and j ($i, j \in S$) in hMC_{FGA} are said to be communicating states if and only if each of them is reachable from the other one. It is denoted by $i \leftrightarrow j$.*

***Definition* 9:** *A state i in hMC_{FGA} is called an absorbing state if it is impossible to leave it (i.e., $p_{ii} = 1$).*

Let us first show that S for hMC_{FGA} can be partitioned.

Lemma 7: *State space S of hMC_{FGA} can be partitioned into r disjoint subsets*

$$S = \bigcup_{i=1}^{r} c_i \tag{7.5}$$

where $c_i \cap c_j = \oslash$, $i \neq j$ and c_i communicates with at least one other subset c_k.

Proof : In hMC_{FGA}, S is composed of a combination of states with three parameters ($c_i = \{s_i, d_i f_i\}, c_i \in S$) namely speed ($s_i$), movement direction ($d_i$), and fitness value ($f_i$). Hence,

$$\begin{aligned} S &= \bigcup_{i=1}^{r} c_i \\ &= c_1 \cup c_2 \cup \cdots \cup c_n \\ &= s_1 d_1 f_1 \cup s_2 d_2 f_2 \cup \cdots \cup s_j d_j f_j \cup \cdots \cup s_r d_r f_r \end{aligned} \tag{7.6}$$

Any solution combination of s_j, d_j, and f_j assigned by FGA at any time must belong to one of the states in hMC_{FGA} and, therefore, part of a subset $c_j = \{s_j, d_j, f_j\}$. $c_i \cap c_j \neq \oslash$ implies that at least one component of a state is the same in c_i and c_j. Since the states in hMC_{FGA} are the combination of sped, direction, and fitness score, they are unique and cannot be analyzed as individual component (i.e., speed, direction, and fitness score). Thus,

$$c_i \cap c_j \neq \oslash \Leftrightarrow \{s_i d_i f_i\} \cap \{s_j d_j f_j\} \neq \oslash \Rightarrow c_i = c_j \tag{7.7}$$

Therefore, we can conclude that $S = \bigcup_{i=1}^{r} c_i$ holds. ∎

Now, let us show that to reach any state c_i ($c_i \in S$) is possible in a finite number of steps in hMC_{FGA}.

***Lemma* 8:** *hMC_{FGA} is irreducible if and only if $P(\tau_{c_j} < \infty | x_0 = c_i) > 0$ for every $c_i, c_j \in S$ where $P^0(x_0 = c_i)$ and τ_{c_j} is the smallest number of steps to move from state c_i to c_j.*

Proof : The probability of moving from any state c_i in hMC_{FGA} to an arbitrary state c_j ($c_i \neq c_j$, $c_i, c_j \in S$) requires l finite number of steps (Def. 8): $P^l_{c_i c_j} > 0$.

As a contradiction suppose there is a state c_j an absorbing (Def. 9) such that $p_{c_j c_j} = 1$. By Lemma 7, $c_j = \{s_j d_j f_j\} = \bigcup_{n=1}^{m} s_j d_j f_n$ and $m < \infty$ (i.e., the state c_j is a combination of a certain speed and movement direction with all possible fitness values assigned by our FGA and Eq. 4.2). In this case, there is a fitness value (f_r) in the state j such that $P_{s_j d_j f_r \; s_j d_j f_r} = 1$ ($\{s_j d_j f_r\} \subset c_j \subset S$ and $1 \leq r \leq m$). Since $\{s_j d_j f_r\}$ is an absorbing state, if a mobile node's speed, movement direction, and fitness value define it as being in this state, the node cannot exit it. In other words, there is no better solution (at any time) for the corresponding mobile node. If a mobile node's fitness is optimal, our GA-based framework assigns the speed and movement direction to zero for the corresponding autonomous mobile node so $s_j = 0$ and $d_j = 0$. As seen in Eq. 4.2, the objective function uses the neighboring nodes' location information. If the neighboring nodes are not at the same absorbing state (each node runs its own FGA), they may be at the state with non-zero speed (i.e., mobile). Therefore, the nodes will be pushed out of the optimal state and must move to a better location.

Also, even if the entire system is at state $s_j d_j f_r$, due to malfunction or communication loss, mobile nodes may need to change their positions and therefore take the entire system out of equilibrium. Thus, $s_j d_j f_r$ cannot be an absorbing state and we can conclude that hMC_{FGA} is irreducible. ∎

***Definition* 10:** *The periodicity (d_{c_i}) of a state c_i ($c_i \in S$) in a Markov chain is defined by $d_{c_i} = gcd\left\{n \geq 1 : p^n_{c_i c_i} = P_r(x_n = c_i | x_0 = c_i) > 0\right\}$ where gcd represents "greatest common denominator." The periodicity is found by determining the number of steps for every possible path to leave and return to a particular state, then by determining the largest number that can be divided by every path count. $d_{c_i} = \infty$ if $p^n_{c_i c_i} = 0$, $\forall\ n \geq 1$. A state c_i is said to be aperiodic if $d_{c_i} = 1$.*

Let us show that hMC_{FGA} is aperiodic.

***Lemma* 9:** *hMC_{FGA} is aperiodic.*

Proof : By Def. 10, hMC_{FGA} is aperiodic if the probability of moving from any state c_i ($c_i \in S$) to itself is not zero, $p_{c_i c_i} > 0$.

Let us suppose that an autonomous mobile node is in the state $c_i = \{s_i d_i f_i\}$ at time t. If there is no change in the neighborhood conditions between from t to $t+1$, FGA does not change the speed, direction, and fitness score of a node (Eq. 4.2). Therefore, in hMC_{FGA}, at each state $c_i \in S$, there is a self-loop to represent the cases where neighborhood information remains the same at time $t+1$ as in time t. Thus hMC_{FGA} is aperiodic. ∎

***Lemma* 10:** *hMC_{FGA} is ergodic if there exists a time t such that $P^t(c_i, c_j) > 0$, $\forall$ c_i, c_j S.*

Proof : Based on Lemma 8, hMC_{FGA} is irreducible implying that the probability of moving any state c_i to an arbitrary state c_j requires a finite number of steps. From Lemma 9, hMC_{FGA} is aperiodic with $d_{c_i} = 1$ (Def. 10). Therefore, from Lemmas 8 and 9, MC_{FGA} is ergodic. ∎

7.2.1 Analysis of Convergence Properties for Ergodic Homogeneous Finite Markov Chain

To prove the convergent behavior of a homogeneous Markov chain, the following measures are given. The total variation between the distributions of random variables in a given set is represented by

$$\|\mu - \nu\| = \sum_{n} |\mu(x) - \nu(x)| \tag{7.8}$$

where S is a finite set and distributions μ and ν are on S. This metric is the underlying property for Dobrushin's contraction coefficient [75] that provides an approximate measure of the orthogonality between the distributions of a transition matrix (specifically, the rows of a right-stochastic matrix and the columns of a left-stochastic

matrix). This is given as

$$c(P) = \frac{1}{2} \cdot \underbrace{max}_{x,y} |P(x,\cdot) - P(y,\cdot)| \tag{7.9}$$

where c is the contraction coefficient, P is a transition matrix, and x and y are the rows of P. Explicitly, the contraction coefficient represents half of the largest total variation of all combinations of rows in the transition matrix. $c(P) = 1$ when any rows of P are completely disjoint. $c(P) = 0$ when every distribution in $P(x,\cdot)$ is equivalent. These metrics are used to show that the interaction of distributions with an ergodic system reduces the orthogonality between distributions.

***Lemma* 11:** *(taken from [74]) Let P and Q be transition matrices and let μ and ν be probability distributions :*

$$\begin{aligned} |\mu P - \nu P| &\leq c(P)\,|\mu - \nu|\,,\ c(PQ) \leq c(P)c(Q) \Rightarrow \\ &\leq |\mu - \nu|\,,\ |\mu P - \nu P| \leq 2 \cdot c(P) \end{aligned} \tag{7.10}$$

Proof : Proof is given in [74]. ∎

In Lemma 11, starting from initial distributions of μ and ν using P are represented as μP and νP, respectively. $|\mu P - \nu P|$ is the total variation between μ and ν. Therefore, as μ and ν progress in time for a transition matrix of P, the orthogonality between the disjoint distributions of μ and ν reduces.

For an ergodic Markov chain, P is a primitive. Rows then become similar as time passes.

***Lemma* 12:** *(taken from [74]) For every time step of the transition matrix P, the sequence $(c(P^t))t \geq 0$ decreases.*

Proof : Proof is given by [74] using Lemma 11:

$$c(P^{t+1}) \leq c(P)c(P^t) \leq c(P^t) \tag{7.11}$$

∎

Here [74] states that each transition of a homogeneous Markov chain reduces the orthogonality of the rows in the transition matrix. To put differently, the differences in the probability distribution from one time step to the next become smaller.

***Lemma* 13:** *(taken from [74]) If the transition matrix P is primitive, the sequence of contraction coefficients reduces to* 0.

Proof : Proof is given in [74] by using Lemma 12:

$$c(P^t) \leq (Q^k P^{t-\tau \cdot k}) \leq c(Q)^k \cdot c(P^{t-\tau \cdot k}) \leq c(Q)^k \tag{7.12}$$

where $Q = P^\tau$ and τ is the smallest number of steps for any state to reach any other state. Therefore, Q is a simplified transition matrix where each state can reach all

other states in a single *large* step. k is defined as the number of *large* steps where $\tau \cdot k \geq t$. Assuming P is primitive and therefore Q is strictly positive (i.e., every state is reachable) then $c(Q) < 1$ without equality and $c(P^t)$ will go to zero as t goes to infinity [74]. ∎

By Lemmas 12 and 13, some final distribution exists if a transition matrix P is primitive. The probability distribution ν converges to the limiting distribution μ as t becomes large. Moreover, the limiting distribution μ does not depend on the initial condition.

***Theorem* 1:** *(taken from [74]) For a primitive P on a finite space with a strictly stationary distribution μ, beginning at any distribution ν, $\nu P^t \to \mu$ as $t \to \infty$.*

Proof : Proof is given in [74]. By Lemma 12, the sequence $lim_{t\to\infty} c(P^t) = 0$ and by Lemma 11:

$$\left|\nu P^t - \mu\right| = \left|\nu P^t - \mu P^t\right| \leq |\nu - \mu|\, c(P^t) \leq 2 \cdot c(P^t) \tag{7.13}$$

∎

In summary, Lemmas 12 and 13 show that as a metrically transitive Markov chain transitions through each generation, it converges towards a strictly stationary distribution. This result is explicitly stated for any initial distribution in Theorem 1.

Based on Lemma 10 and Theorem 1, we state that hMC_{FGA} will converge in time to a stationary behavior.

Theorem 2: *hMC_{FGA} is ergodic and therefore it must converge to a stationary distribution.*

Proof : As demonstrated in Lemma 10, the transition matrix P is irreducible and aperiodic for hMC_{FGA} and therefore is ergodic. Based on Theorem 1, As P is ergodic, any probability distribution ν converges to a limiting distribution μ without depending on the initial condition as time t passes. Therefore, hMC_{FGA} must converge to a stationary distribution. ∎

0	0	0.1438	0.15498	0.16097	0.1654	0.15057	0.1458	0.01218	0.0133	0.01356	0.01467	0.0125	0.01225	0
0	0	0.01988	0.02206	0.02284	0.02326	0.02114	0.01992	0.13548	0.14871	0.15774	0.1564	0.13913	0.13344	0
0.4616	0.07456	0.12982	0.0253	0.06764	0.10329	0.07423	0.01949	0.00724	0.0024	0.00878	0.00964	0.00847	0.00184	0.00569
0.47459	0.06812	0.01489	0.19027	0.02479	0.0625	0.06832	0.05191	0.00111	0.01004	0.00303	0.00606	0.00816	0.00639	0.0098
0.48826	0.06542	0.04026	0.03022	0.22021	0.02244	0.04843	0.043	0.00221	0.00272	0.01233	0.00142	0.00351	0.0046	0.01498
0.48101	0.06655	0.05245	0.03887	0.02905	0.23481	0.03272	0.03194	0.0029	0.00377	0.00262	0.00788	0.00249	0.00368	0.00927
0.46006	0.0715	0.07047	0.08602	0.08021	0.0196	0.14545	0.01938	0.00685	0.01089	0.00911	0.00105	0.00892	0.00181	0.00868
0.47776	0.07667	0.01615	0.07522	0.07857	0.06908	0.01697	0.13935	0.00134	0.01059	0.01237	0.00955	0.00251	0.00823	0.00564
0.16963	0.58118	0.02176	0.00322	0.00531	0.00667	0.00836	0.00371	0.03554	0.01158	0.04267	0.03911	0.04478	0.01304	0.01343
0.15633	0.52148	0.00317	0.05238	0.00309	0.00648	0.01096	0.00853	0.00571	0.08386	0.01717	0.02399	0.04597	0.03508	0.02579
0.15747	0.51363	0.00666	0.0047	0.05978	0.0018	0.0082	0.00732	0.01246	0.01823	0.11057	0.00634	0.02635	0.02679	0.03969
0.1734	0.54992	0.00942	0.00678	0.0042	0.03621	0.00631	0.00749	0.01842	0.02696	0.015	0.06837	0.01663	0.02823	0.03267
0.15314	0.52382	0.00932	0.00822	0.00655	0.00163	0.04987	0.00409	0.03349	0.04934	0.04351	0.00612	0.07595	0.01422	0.02073
0.15728	0.53641	0.0025	0.0111	0.0106	0.01008	0.00239	0.04168	0.00668	0.04888	0.05386	0.03857	0.01123	0.05559	0.01313
0.03404	0.04138	0.00292	0.00212	0.00267	0.0041	0.00197	0.0016	0.00029	0.00125	0.00161	0.00156	0.00075	0.0003	0.90345

Figure 7.2: An example of transition matrix P for $N = 100$ and $R_{com} = 10$

This theorem allows us to state that our FGA must converge to a stationary behavior. Also, experimentally we can observe a close approximation of the final stationary distribution of the Markov chain for our FGA. We can compare the rate of convergence for various parameters of the mobile nodes using our FGA as a software agent.

7.2.2 Simulation Experiments for hMC_{FGA}

In order to study the convergence rate of hMC_{FGA} for a uniform distribution of knowledge sharing mobile nodes, we used our simulation software explained in Section 5.1.

We consider an experimental setup where the autonomous mobile nodes enter an unknown geographical area of A_{RoI} without any a priori information and without a central control unit. Two different types of experiments were set up to analyze the effects of different network parameters on the convergence rate of hMC_{FGA}. For the first experimental setup, we ran the experiments for networks with $N = 100$ and different communication range of $R_{com} = 5$, 10, 12, and 15, we, thus, are able to analyze the effects of different communication ranges on our model. In the second set of experiments, we changed the number of autonomous mobile nodes in a MANET, $N = 100$, 125, and 150 with a fixed communication range of $R_{com} = 10$. For each experiment, the area of deployment is set to be 100x100 units with all mobile nodes initially placed in the north-west corner of the terrain as shown in Fig. 5.2. To smooth out the noise and to increase accuracy in transition probabilities of hMC_{FGA} transition matrix, each experiment is repeated for 75 times with the same initial values (i.e., initial speed and initial movement direction) and the same initial node deployments (i.e., initial location). As explained in the previous sections, without loss of generality, each mobile node has the same movement capabilities (e.g., speed and movement direction) and the limited communication range of R_{com}, hence, can only be aware of its neighbors and runs its own GA-based software application.

7.2.2.1 A Reduced Size Transition Matrix for hMC_{FGA}

As explained in Section 7.2.1, we use a reduced size transition matrix to model the behavior of our FGA using an ergodic homogeneous Markov chain so that we can

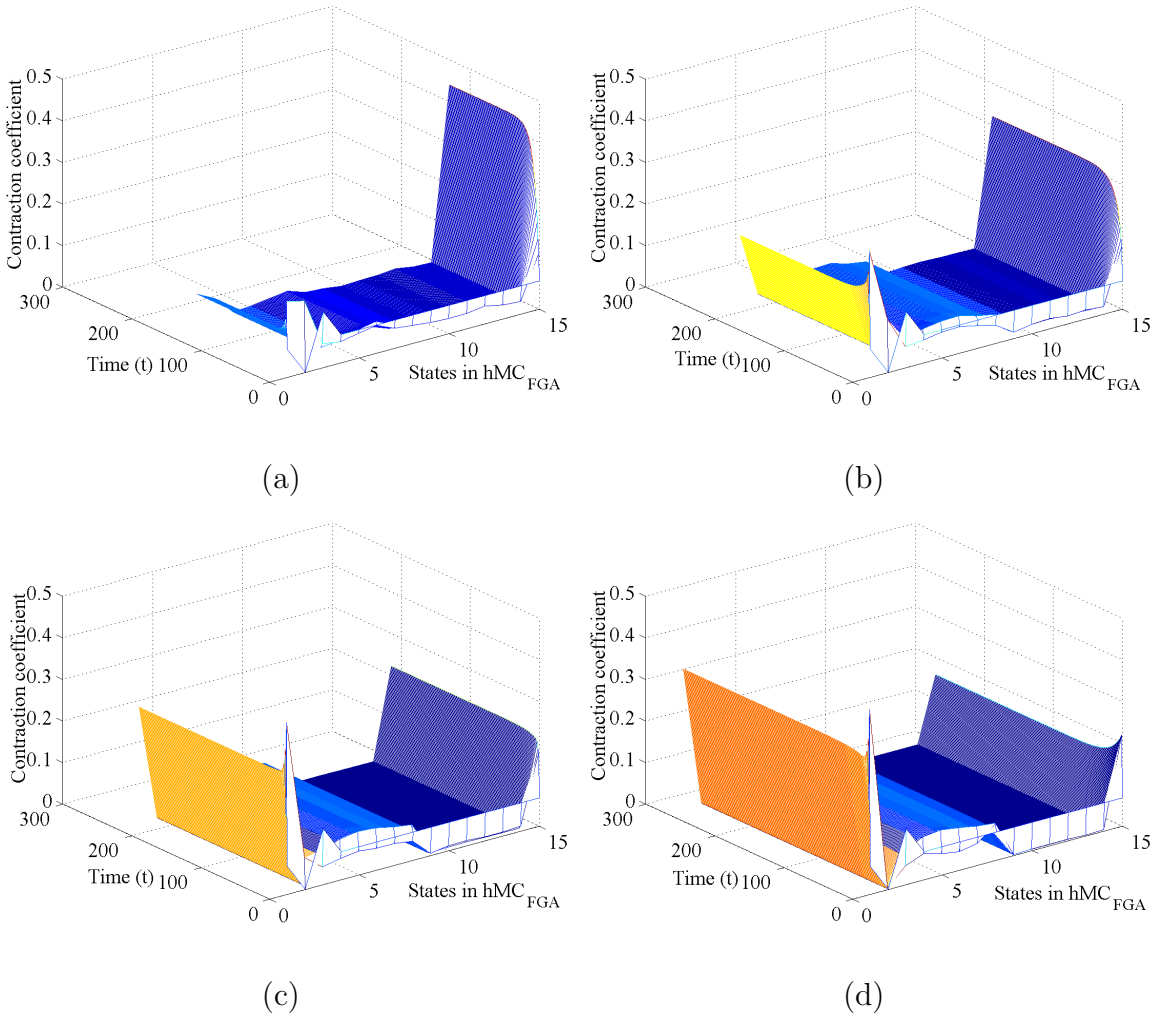


Figure 7.3: Distribution of homogeneous Markov chain states for $N = 100$ and (a) $R_{com} = 5$, (b) $R_{com} = 10$, (c) $R_{com} = 12$, and (d) $R_{com} = 15$

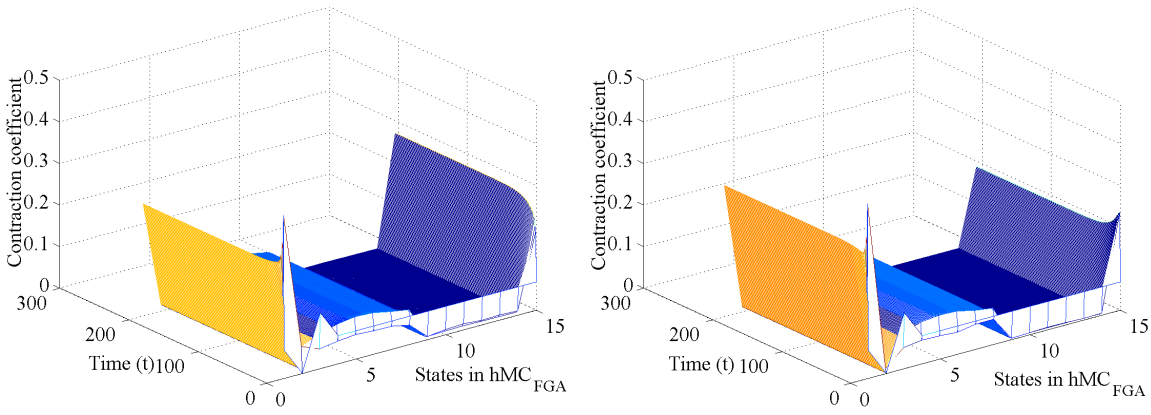
analyze and easily present the convergence rate and properties. When we use all possible states in our approach, there are in the order of 10^7 possible states for a mobile node as explained in the previous section. It is not feasible to display and analyze this information in this dissertation. In order to keep the convergence analysis as simple as possible, 10^7 states were merged into 15 states shown in Fig. 7.1. This simplification makes hMC_{FGA} easy to present and analyze with all properties including the states and the transition matrix. Fig. 7.2 shows the hMC_{FGA} transition matrix

for $N = 100$ mobile nodes with a communication range $R_{com} = 10$. As seen from Fig. 7.2, two properties of a transition matrix P are satisfied as:

1. $p_{ij} \geq 0$

2. $\sum_{k \epsilon S} p_{ik} = 1$

7.2.2.2 Convergence Rate of hMC_{FGA}



(a) (b)

Figure 7.4: Distribution of homogeneous Markov chain states for $R_{com} = 10$ and (a) $N = 125$ and (b) $N = 150$

We ran two types of experiments for networks with $N = 100$ mobile nodes with communication ranges of $R_{com} = 5$, 10, 12, and 15 and for networks with communication range of $R_{com} = 10$ and $N = 100$, 125, and 150 mobile nodes. Figs. 7.3 (a)-(d) and 7.4 (a)-(b) present the distribution of Markov chain and show that the system evolves to a stationary distribution as time goes to infinity. It is important to note that any initial distribution will converge to the same stationary distribution based on Theorem 2. The only difference in using varied initial distributions will be the

number of steps that the system takes to reach the stationary distribution. When we think about all of the possible initial deployments for autonomous mobile nodes in a MANET, it makes practical sense. If the mobile nodes are initially dispersed such that they are close to uniform spatial distribution over the geographical terrain, then they will take very few movements to achieve a uniform distribution. In the experiments, autonomous mobile entities are placed in the worst case scenario where all of the mobile nodes are clustered in a single corner (northwest) as shown in Fig. 5.2. In this case, many mobile nodes will initially be trapped between other mobile nodes and the boundaries of A_{RoI}. This will increase the time required for the mobile nodes to reach spatial uniformity. The importance of the relationship between initial distributions of the Markov chain and the initial dispersement of mobile nodes is that the Markovian representation of the FGA, accurately represents experimental behavior.

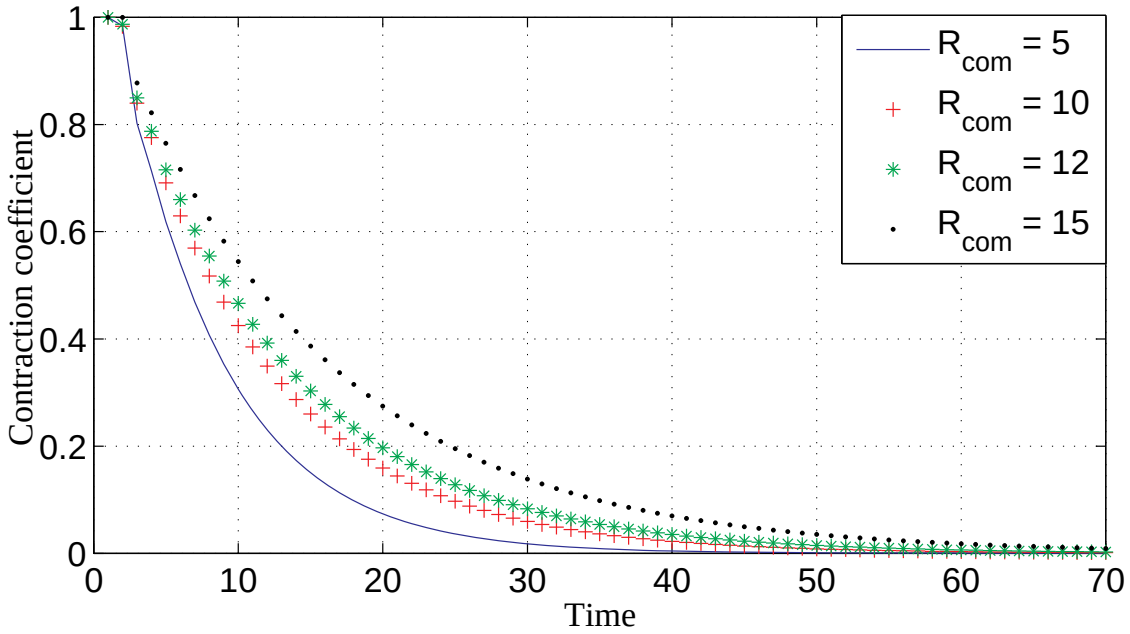


Figure 7.5: Contraction coefficient as $t \to \infty$ when the total number of mobile nodes in a MANET is fixed ($N = 100$) and the communication range varies at each experiment ($R_{com} = 5$, 10, 12, and 15)

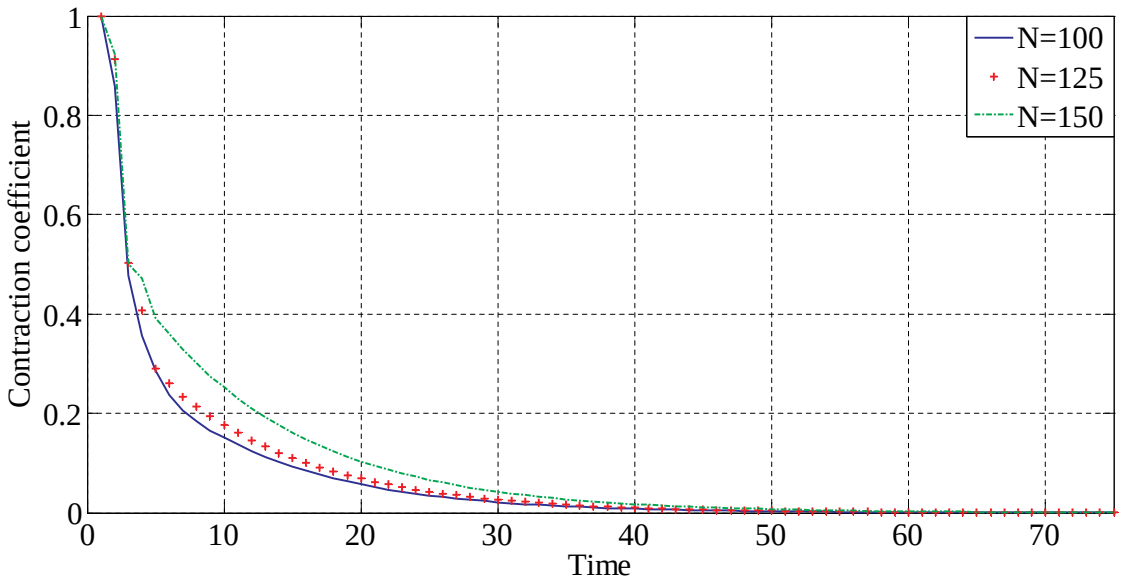


Figure 7.6: Contraction coefficient as $t \to \infty$ when the communication range is fixed ($R_{com} = 10$) and the total number of mobile nodes in a MANET varies ($N =$ 100, 125, and 150)

Figs. 7.5 and 7.6 show Dobrushin's contraction coefficient when time goes to infinity for two different types of experiments with differing communication ranges ($R_{com} = 5$, 10, 12, and 15) and the total numbers of nodes ($N = 100$, 125, and 150). respectively. The graph is only based on the transition matrix for FGA and not on various initial distributions of the Markov chain so that it is possible to make direct comparisons between experiments with varying parameters. As seen Fig. 7.5, experiments with larger communication ranges converge at a slower rate than experiments with smaller ones. It is important to note that this refers to the convergence to the ideal state which has the desired number of nodes and perfect fitness (i.e., **Stop, ideal, 1**). The system reaches the final distribution at time $t \approx 38$, 50, 60, and 70 for $R_{com} = 5$, 10, 12, and 15, respectively. When the autonomous mobile nodes have smaller communication ranges, they collect less local neighborhood information and the FGA generates

next speed and movement direction based on this limited information. Furthermore, the reduced aggregation of external forces results in faster convergence. As seen in Fig. 7.6, experiments with larger numbers of nodes converge at slower rates than experiments with smaller numbers of nodes. This is due to the fact that more autonomous mobile nodes are initially trapped with limited mobility. As nodes at the periphery of the cluster quickly begin to spread, the mobility of nodes in the center of the cluster also begins to increase. The Markov kernel for our FGA reaches the final distribution when $t \approx 50$ for all experiments. The convergence for varied initial distributions can be added to these graphs (Figs. 7.5 and 7.6) by finding $|\nu P^t - \mu P^t|$ with respect to t for any of the experimental cases. It is important to note that for each experiment, any initial distribution (ν) will fall below the corresponding line and hence converges faster than the contraction coefficient graph of the Markov kernel as in the proof of Theorem 2.

Figs. 7.7 (a)-(d) and 7.8 (a)-(b) represent the possible outcome percentages of each state in hMC_{FGA} for varying communication ranges and numbers of nodes, respectively. In Fig. 7.7 (a), the probability of being in the ideal state for a mobile node is higher than the other states (44%) when $R_{com} = 5$. It demonstrates that nearly half of the mobile nodes reach the state where they have the desired number of neighbors and locations that result in minimal external force. The probability of reaching the stop state with a non-ideal number of neighbors is approximately 10%. The remaining states that are not explicitly labeled with values ranging from 2% to 8% represent states where the node is moving and has an ideal number of neighbors. As seen from

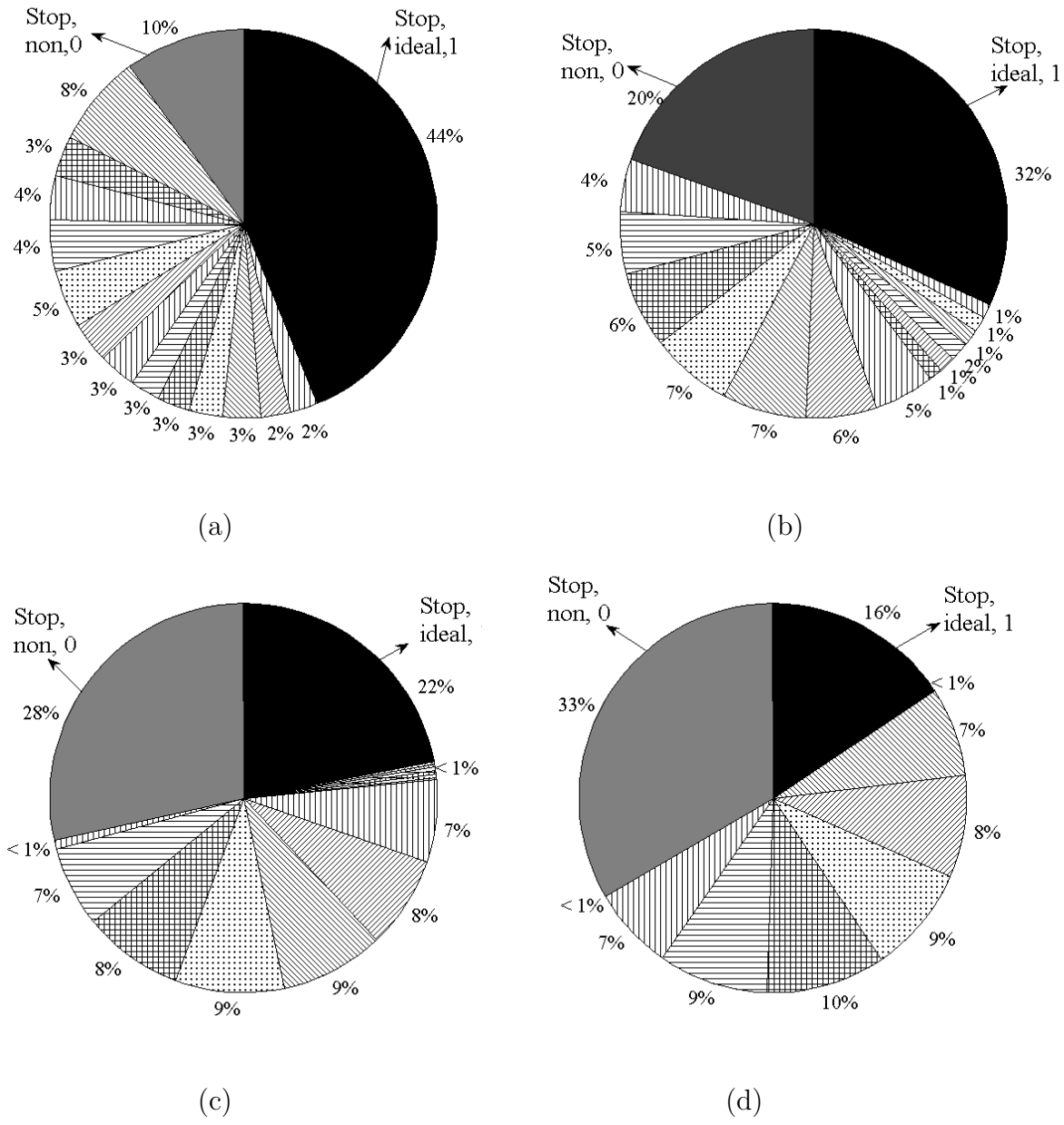
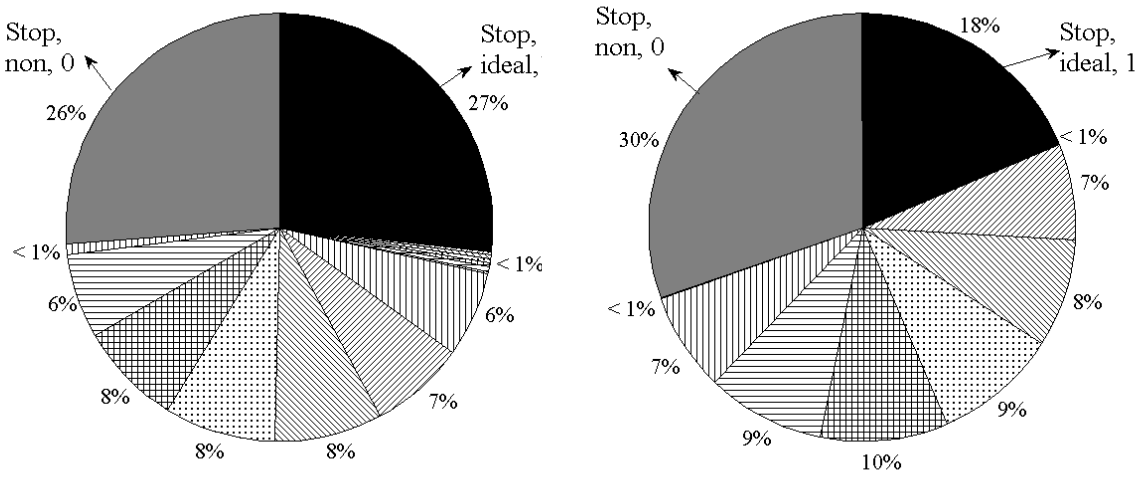


Figure 7.7: Output distribution of each state in hMC_{FGA} for $N = 100$ and (a) $R_{com} = 5$, (b) $R_{com} = 10$, (c) $R_{com} = 12$, and (d) $R_{com} = 15$

Figs. 7.7 (b)-(d), when $R_{com} = 10$, 12, and 15, the probabilities of being in the desired state are 32%, 22%, and 15%, respectively. As seen from these results, to reach and stay at the desired state for a mobile node is less probable when communication range increases. It is an expected result since larger communication range means more local

neighborhood information and more neighboring nodes that results in a less stable position (aggregated force on a mobile node is not zero). Figs. 7.8 (a)-(b) show the possible outcome percentages of each state in hMC_{FGA} for the experiment comparing varying numbers of nodes with $R_{com} = 10$ and $N = 125$ and 150, respectively. Referring back to Fig. 7.7 (b), $R_{com} = 10$ and $N = 100$, the probability of being in the ideal state for a mobile node is (32%), the probability of being in the stopped, non-ideal state is only (20%) and the sum of all remaining moving states is (48%). When $N = 125$ it can be seen in Fig. 7.8 (a) that the probability of being in the ideal state is (27%), the probability of being in the stopped, non-ideal state is now (26%) and the sum of all remaining moving states is now (47%). As presented in Fig. 7.8 (b), the probability of being in the ideal state for a mobile node is 18% and the probability of reaching the stop state with non-ideal number of neighbors is 30% when there are 150 mobile nodes in a MANET. These data reveals that as the network area is overcrowded with autonomous mobile nodes, increased energy is consumed in the search for optimal spatial orientation. As crowding increases, fewer mobile entities will be able to find an optimal orientation and more nodes will continue to search for better spatial configuration. Fig. 7.6 demonstrates that these will eventually converge to a stationary distribution, but as seen in Figs. 7.8 (a)-(b), when the number of mobile nodes increases beyond the minimal number necessary to completely cover the network area, more nodes will be in a stopped state with a non-ideal number of neighbors (due to overcrowding) than the stopped state with an optimal number of neighbors and high fitness.



(a) (b)

Figure 7.8: Output distribution of each state in hMC_{FGA} for $R_{com} = 10$ and (a) $N = 125$ and (b) $N = 150$

Effectiveness in area coverage (A_{eff}) is an important performance metric of our FGA approach as discussed in Section 4.3. Fig. 7.9 shows the A_{eff} for $N = 100$ and $R_{com} =$ 5, 10, 12, and 15. Comparison with the convergence rate of the same experiments discussed in Fig. 7.5 appears to bring a new perspective with the experiment $R_{com} =$ 15 achieving a high percentage of A_{eff} rather quickly and the experiment with $R_{com} =$ 5 gradually maximizing its network area coverage as time progresses. This high level of area coverage comes at the cost of a great deal of redundancy and a situation where mobile nodes will continue to make large adjustments in their spatial orientation for a great deal of time. When $R_{com} = 5$, the autonomous mobile nodes spread over quickly. Many of the autonomous mobile entities are in a state of high fitness and movements that continue to maximize A_{eff} are minute. The effect of increasing the communication range of R_{com} is to increase the rate that network will attain

a high level of network coverage, while at the same time decreasing the rate that the network will converge to a stationary behavior. It will also cause an increasing amount of network overlap and consumption of power through continued movement.

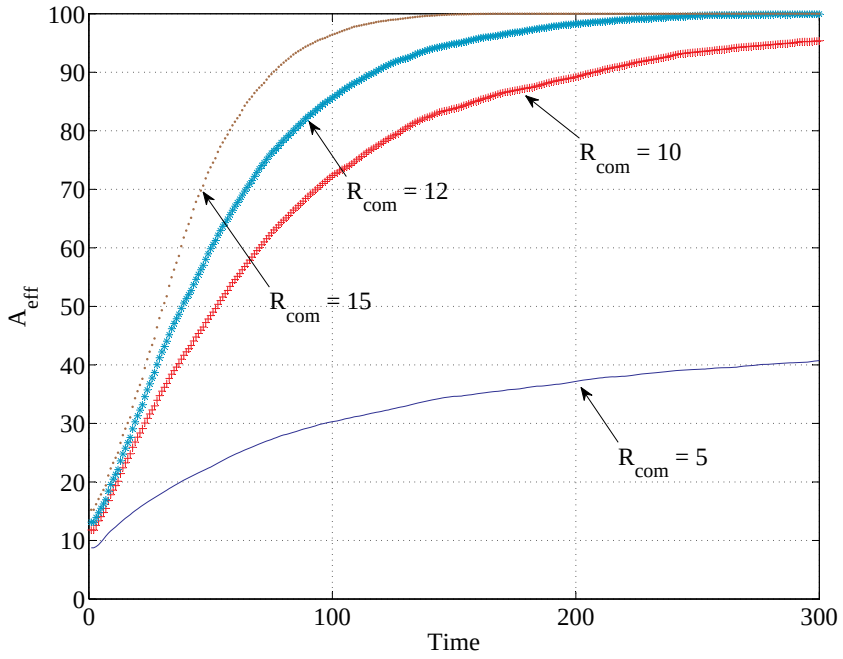


Figure 7.9: Effectiveness in area coverage (A_{eff}) for $N = 100$ and $R_{com} =$ 5, 10, 12, and 15

7.3 Convergence Analysis of Inhomogeneous Markov Chain Model of our FGA

In this section, we extend our hMC$_{\text{FGA}}$ (given in Section 7.2) and present here an inhomogeneous Markov chain with a Markov kernel called iMC$_{\text{FGA}}$. In iMC$_{\text{FGA}}$ model, our approach in terms of constructing Markovian model and reducing the number of states given in Section 7.2 are the same. The difference of iMC$_{\text{FGA}}$ model, the Markov

kernel (i.e., transition matrix) is different for every time step (i.e., $P_t = P_1, P_2, \cdots$, where t= 1, 2, $\cdots$) for a given finite space S, with any initial distribution of ν. The distribution of states $x \epsilon S$ at times $t \geq 0$ is given by

$$P^{(t)}(x_0, \ \cdots, \ x_t) = \nu(x_0)P_1(x_0, \ x_1)\cdots P_t(x_{t-1}, \ x_t) \tag{7.14}$$

This model has the benefit over a homogeneous model by preserving the time-based precision of experimental data explained below.

7.3.1 Analysis of Convergence Properties for Inhomogeneous Markov Chain

As a first step to prove the convergence of our inhomogeneous Markov model, let us show the existence of a limit distribution by the following lemma.

***Lemma* 14:** *(taken from [74]) If μ_t, $t \geq 1$, are probability distributions on S such that $\sum_t ||\mu_{t+1} - \mu_t|| < \infty$ then there is a probability distribution μ_∞ such that $\mu_t \to \mu_\infty$ as $t \to \infty$.*

Proof : For $s < t$

$$||\mu_t - \mu_s|| \leq \sum_{r \geq s} ||\mu_{r+1} - \mu_r|| \tag{7.15}$$

Proof is given in [74]. ∎

Lemma 14 states that if there is a sequence of distributions μ_t that have a sum less than infinity, a limiting distribution μ_∞ exists. This is because S is a finite and closed space and any limiting sequence must exist inside of S. This is commonly known as a *Cauchy sequence* [125].

State space S for our FGA is finite, point-wise convergence, and convergence in the L^1-norm as shown by the following lemma.

***Lemma* 15:** *The state space S for i*MC_{FGA} *is a finite and closed space.*

Proof : hMC_{FGA} and iMC_{FGA} have the same state space S, and, hence, Eq. 7.5 can be used for iMC_{FGA}. Therefore, the proof for Lemma 7 holds for iMC_{FGA}. ∎

Lemma 14 leads to the following theorem, commonly coined to its founder R. L. Dobrushin [75].

***Theorem* 3:** *(taken from [74]) Let P_t, $t \geq 1$, be transition matrices, each with an invariant probability distribution μ_t. Given that the following conditions exist*

$$\sum_t ||\mu_{t+1} - \mu_t|| < \infty \tag{7.16}$$

$$\lim_{t\to\infty} c(P_i \cdots P_t) = 0 \text{ for every } i \geq 1 \tag{7.17}$$

Then $\mu_\infty = lim_{t\to\infty}\mu_t$ exists and starting from any distribution ν.

$$\nu P_i \cdots P_t \rightarrow \mu_\infty \; as \; t \rightarrow \infty \tag{7.18}$$

Proof : Proof is shown in [74]. ∎

Here Dobrushin states that an inhomogeneous Markov chain on a finite set S will have a limiting distribution as long as the sum of the total variation between its one-step output distributions is finite and that the contraction coefficients for the transition matrices go to zero for any starting point i.

Based on Lemma 15 and Theorem 3 we assert that iMC_{FGA} will converge to a stationary behavior:

***Theorem* 4:** *The set of inhomogeneous transition matrices for the simplified* FGA *fulfills both conditions set by Theorem 3 and, therefore, it will converge to a stationary distribution.*

Proof : The existence of limit μ_∞ was proved since the state space S for iMC_{FGA} is a finite and closed space as shown by Lemma 15. The set of inhomogeneous Markov kernels for our FGA is has a finite sum in the one step total variation of output distributions and that the contraction coefficient of the transition matrix will converge to zero from any starting point. Therefore, using Theorem 3, it will converge to a stationary distribution. ∎

This theorem allows us to state that our GA-based topology control approach must converge to a stationary behavior. Also, we experimentally observe a close approxi-

mation of the final stationary distribution of iMC$_{\text{FGA}}$. This allows us to compare the rate of convergence for various network parameters including number of mobile nodes and communication range.

7.3.2 Simulation Experiments for iMC_{FGA}

In order to study the convergence rate of iMC$_{\text{FGA}}$ for topology control of mobile nodes in a MANET, we used our simulation software explained in Section 5.1. We consider the same experimental setup in Section 7.2.2. There are two different types of experimental setups to analyze the effects of different network parameters on the convergence rate of iMC$_{\text{FGA}}$. We ran the experiments for networks with $N = 100$ and different communication range of $R_{com} = 5$, 10, 12, and 15, we, thus, are able to analyze the effects of different communication ranges on our model in the first experimantal setup. In the second experimental setup, we changed the number of mobile nodes in a MANET, $N = 100$, 125, and 150 with a communication range of $R_{com} = 10$. The area of deployment is set to be 100x100 units with all nodes initially placed randomly in the north-west corner of the terrain for all setups. To smooth out the noise and to obtain accurate iMC$_{\text{FGA}}$, each experiment is repeated for 75 times with the same initial values and the same initial node deployments. Without loss of generality, each mobile node has the same limited communication range (R_{com}), and, hence, can only be aware of its neighbors and runs its own GA-based software application.

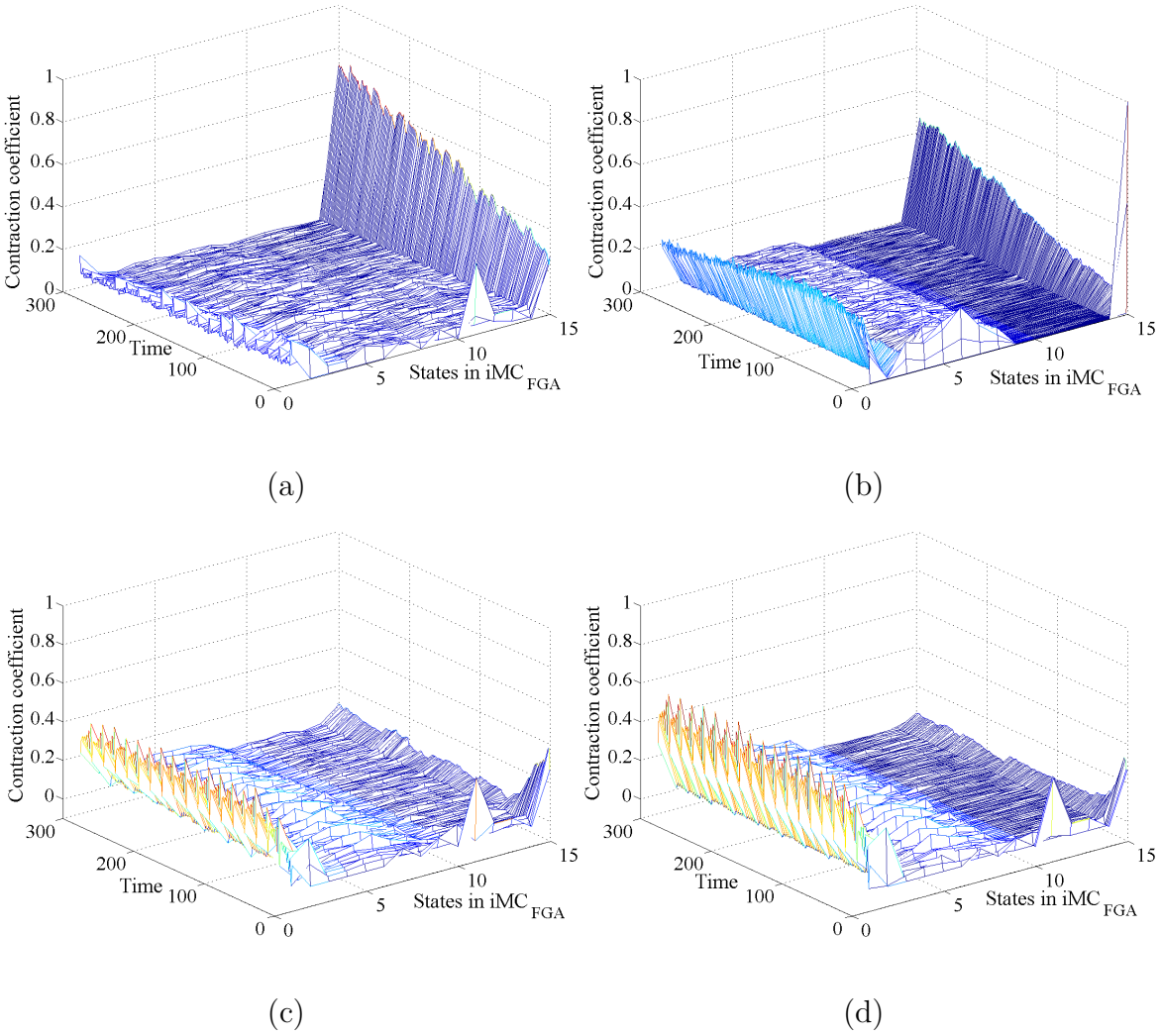
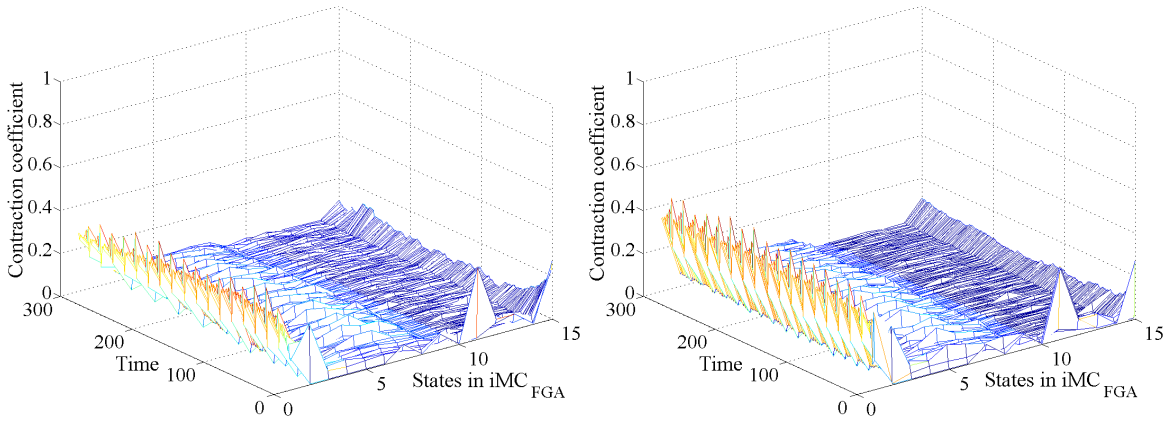


Figure 7.10: Distribution of inhomogeneous Markov chain states for $N = 100$ and (a) $R_{com} = 5$, (b) $R_{com} = 10$, (c) $R_{com} = 12$, and (d) $R_{com} = 15$

7.3.2.1 Convergence Rate of iMC_{FGA}

Figs. 7.10 (a)-(d) and 7.11 (a)-(b) present the distribution of the reduced Markov chain model of our FGA and show that how the system evolves to a stationary distribution as time goes to infinity. In Figs. 7.10 (a)-(d) and 7.11 (a)-(b), the x-axis shows each Markov state of our FGA in Fig. 7.1 from one to fifteen. The 15^{th} state is the (**Stop, ideal, 1**) state where the mobile node does not move because it has an ideal number

of neighbors with a fitness of 1. The y-axis is the time and the z-axis is the probability. As seen in Figs. 7.10 (a) and (b), the system evolves to the (**Stop, ideal, 1**) state as time increases. At this point, it is important to note that any initial distribution will converge to the same stationary distribution based on Theorem 4. The only difference in Figs. 7.10 (a)-(d) and 7.11 (a)-(b) will be the convergence time for the different initial distributions. It makes practical sense. If the mobile nodes are initially distributed in an area such that they are closer to uniform distribution, then they need less time to reach achieve uniform distribution. In our experiment, the mobile nodes enter the geographical area from one entry point, that is considered the worst case scenario to reach uniform node distribution. In Figs. 7.10 (c)-(d) and 7.11 (a)-(b), there is no convergence to the stationary distribution. In fact, this result is expected since the experimental setup for these figures with larger R_{com} and more mobile nodes represent overcrowded region, therefore, less mobile nodes find suitable positions with good fitness and desired number of nodes.



(a) (b)

Figure 7.11: Distribution of inhomogeneous Markov chain states for $R_{com} = 10$ and (a) $N = 125$ and (b) $N = 150$

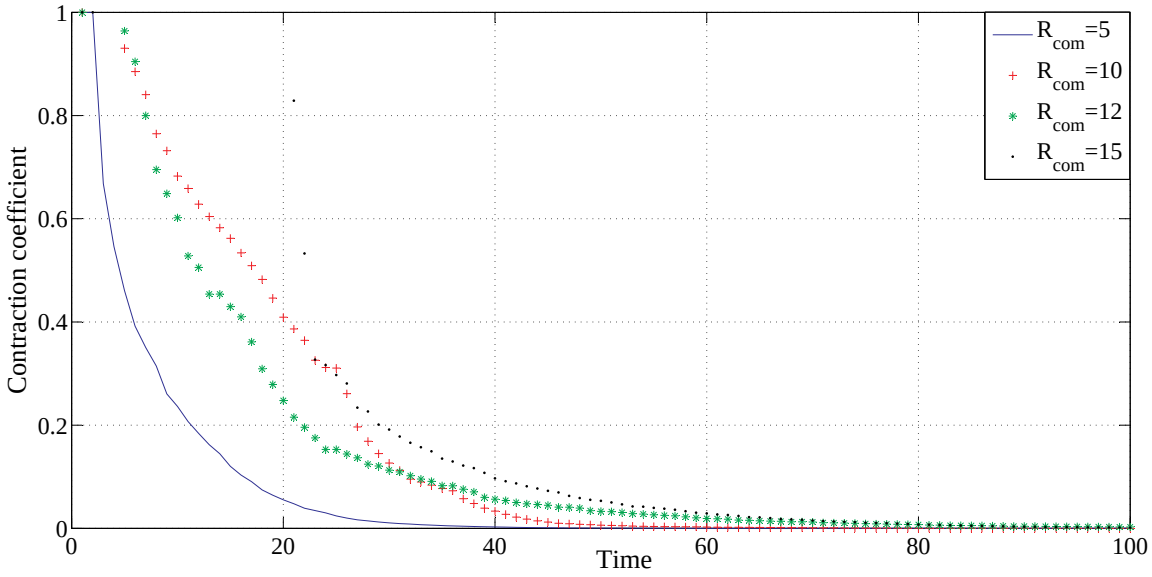


Figure 7.12: Contraction coefficients as $t \to \infty$ when the total number of mobile nodes in a MANET is fixed ($N = 100$) and the communication range varies at each experiment ($R_{com} = 5$, 10, 12, and 15)

Figs. 7.12 and 7.13 show a rough measure of orthogonality between consecutive instants of time in our reduced Markov chain model (i.e., Dobrushin's contraction coefficient) when time goes to infinity. It is important to note that these figures are only based on the transition matrices and not on various initial distributions of our reduced Markov model. Therefore, it gives a direct result of the convergence of iMC$_{\text{FGA}}$ model. As seen in Fig. 7.12, the system reaches its final distribution at time $t \approx$ 40, 50, 80, and 95 for the different communication ranges of $R_{com} = 5$, 10, 12, and 15, respectively. As shown in Fig. 7.13, when there are more mobile nodes in the RoI, convergence to final distribution takes more time. As explained in Section 7.2.2, this is due to the fact that more nodes are initially trapped with limited mobility. Another important observation from Figs. 7.12 and 7.13, the first approximately 10 time units represent volatility in the system due to the mobile

nodes initial placement in one conner of the given terrain. After this period, the nodes have increased degrees of freedom and begin to converge towards to uniform distribution.

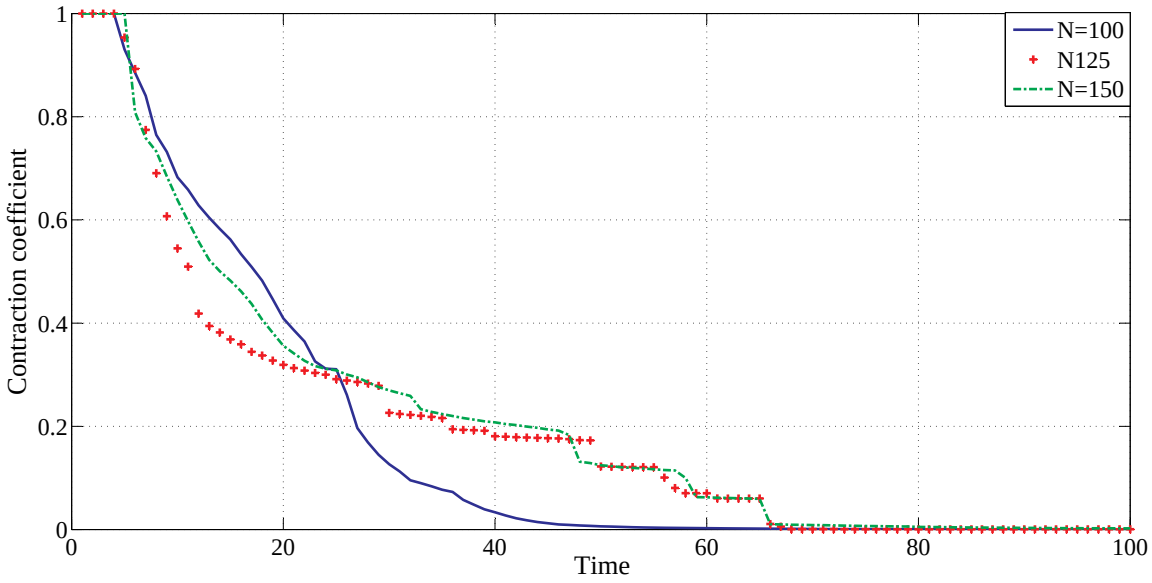


Figure 7.13: Contraction coefficients as $t \to \infty$ when the communication range is fixed ($R_{com} = 10$) and the total number of mobile nodes in a MANET varies ($N =$ 100, 125, and 150)

Figs. 7.14 (a)-(d) and 7.15 represent the outcome distribution of each state in our Markov model: the stop state with high fitness (SSHF), the stop state with low fitness (SSLF), and the aggregation of all moving states (MSHL). As seen by the SSHF plot in Figs. 7.14(a)-(b), the probability of stop with high fitness increases when time goes to infinity. More mobile nodes find desired number of neighbors in the correct position in which the aggregate force on a corresponding mobile node is approximately zero and stay immobile. When time reaches 300, more than half of the mobile nodes for $R_{com} = 5$ and nearly half of the mobile nodes for $R_{com} = 10$ have good fitness and are immobile. On the other hand, the probability of mobile nodes reaching

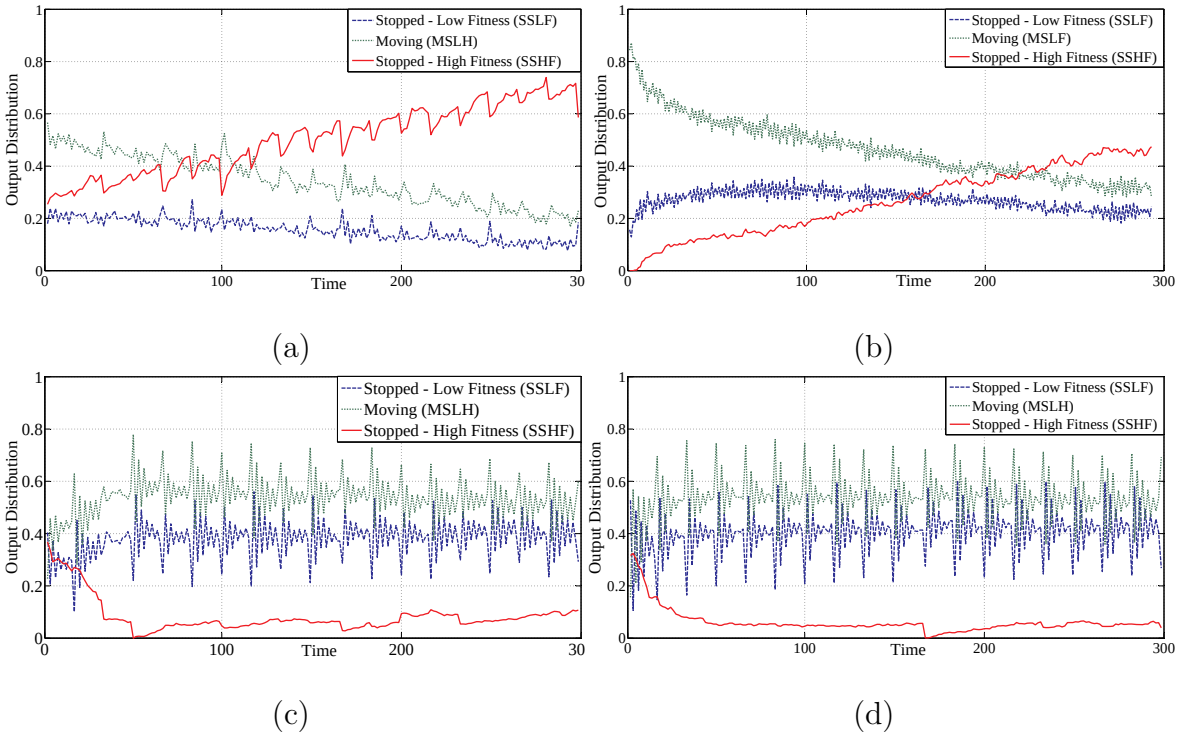
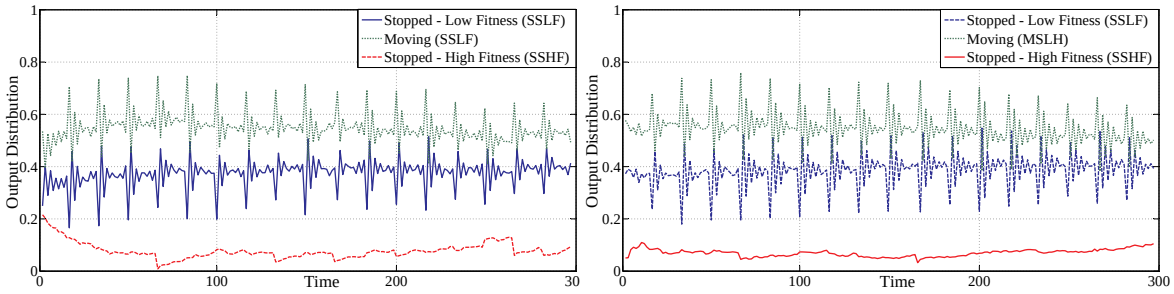


Figure 7.14: Output distribution of each state in iMC_{FGA} for $N = 100$ and (a) $R_{com} = 5$, (b) $R_{com} = 10$, (c) $R_{com} = 12$, and (d) $R_{com} = 15$

and/or staying at the moving states (with any direction and any speed) drops when time passes as shown in Fig.7.14 (a)-(b). A little more than one third of mobile nodes are in any moving state when $t = 300$ when $R_{com} = 5$ and 10. As shown in Fig. 7.14 (a), both MSHL and SSLF decreases as time passes. The stopped state with low fitness increases until $t = 100$ and decreases afterward as seen by the SSLF plot in Fig. 7.14 (b). Initially, the mobile nodes are located at the northwest corner of the geographical terrain and most of them cannot move because of the overcrowded vicinity. After some time passes, there are enough available hexagonal cells to move as assigned by the FGA. The final stationary distribution at $t = 300$ verifies the experimental behavior of our FGA where mobile agents achieve a distribution that is close to the uniform distribution. Some nodes continue to move slightly, these

nodes exert small external forces on neighbors who in turn readjust themselves to return to ideal fitness. For the other experimental setups (R_{com} = 12 and 15 and N = 125 and 150), SSHF has the lowest value with respect to MSHL and SSLF as shown in Figs. 7.14 (c)-(d) and 7.15 (a)-(b). This result is expected since larger communication range and more mobile nodes represent overcrowded region which results in having more negihbors than ideal number of neighbor.



(a) (b)

Figure 7.15: Output distribution of each state in iMC_{FGA} for R_{com} = 10 and (a) N = 125 and (b) N = 150

As mentioned in Section 4.3, A_{eff} is an important performance metric of our FGA approach. Fig. 7.9 shows the A_{eff} for N = 100 and R_{com} = 5, 10, 12, and 15. Comparison with the convergence rate of the same experiments discussed in Fig. 7.12 appears to show diametrically opposed outcomes with the experiment R_{com} = 15 achieving a high percentage of A_{eff} rather quickly and the experiment with R_{com} = 5 gradually maximizing its network area coverage as time progresses. This high level of area coverage comes at the cost of a great deal of redundancy and a situation where mobile nodes will continue to make large adjustments in their spatial orientation for a great deal of time. When R_{com} = 5, nodes spread out quickly. Many of the mobile entities are in a state of high fitness and movements that continue to maximize A_{eff}

are minute. The effect of increasing the communication range of R_{com} is to increase the rate that network will attain a high level of network coverage, while at the same time decreasing the rate that the network will converge to a stationary behavior. It will also cause an increasing amount of network overlap and consumption of power through continued movement.

We can also claim that the convergence analysis experiments for hMC_{FGA} and iMC_{FGA} models provide the similar result. In fact, as explained at the beginning of this section, iMC_{FGA} model provides details in time while hMC_{FGA} shows average behavior. The experimental results from both hMC_{FGA} model and iMC_{FGA} show the convergence of our GA-based topology control approach based on Theorems 2 and 4, respectively.

Chapter 8

Concluding Remarks

In this dissertation, we introduced a bio-inspired topology control approach for efficient self deployment of autonomous mobile nodes in a MANET. Our GA-based topology control framework is used by each mobile node as a stand-alone software agent to decide its next speed and movement direction. It does not require any central control unit or a priori knowledge and only uses local neighborhood information including the mobile nodes and obstacles in the communication range of R_{com} of a corresponding node. Our FGA provides a simple, efficient, and decentralized topology control method to deploy large number of autonomous mobile nodes in an unknown A_{RoI} and is used by each mobile node so that mobile entities do not require any complex and centralized control mechanism. Furthermore, our GA-based topology control approach maintains a satisfactory area coverage, provides high levels of uniformity, and is resilient to mobile node losses (due to the malfunction or destroyed by hostile attacks) while sustaining energy efficient self-deployment.

8.1 Results and Future Research Directions

We provide a formal analysis of the stability and convergence of our FGA by introducing homogeneous and inhomogeneous Markov chain models and providing their convergence.

In Chapter 4, we formally prove that our GA-based topology control application does not let a mobile node to lose communication with its neighboring nodes in Lemma 1. In addition, in the same chapter, we use the Lyapunov stability theorem to show the convergence of the fitness function of FGA given in Eqn. 4.2 to an asymptotically stable state in Lemma 2. We use two important performance metrics, namely A_{eff} and uniformity, in this dissertation to analyze our FGA as defined in Sections 4.3 and 4.4, respectively. Using our uniformity definitions for two mobile nodes and average for all mobile nodes given in Definitions 4 and 5, we formally show the convergence of our GA-based topology control application.

Details of our simulation software and testbed implementations are given in Chapter 5. We, in this chapter, experimentally show the convergence to uniform distribution of autonomous mobile nodes in a MANET. In Section 5.1, the autonomous mobile nodes running our GA-based topology control application as software agent are successfully deployed in A_{RoI} under harsh conditions similar to those found in military applications (e.g., after losing assets during an operation, the remaining mobile nodes should reposition themselves to compensate the loss in coverage and network connectivity). One of the most important results from these experiments is to show

that our topology control algorithm is resilient to loss of assets and is successful in distributing mobile nodes in the presence of obstacles. We also provided details about four different testbeds which use different technologies and components namely FPGA Virtex-II™ with laptops and desktops, VMware™, small robots (iRobots™) controlled by gumstix™ processors, and laptops and PDAs in Section 5.2. Experimental results from our testbed implementations show that our FGA delivers promising results for uniform node distribution of knowledge sharing mobile nodes over an unknown geographical terrain. Moreover, the results from our testbed experiments validate our simulation software results since both show similar outputs in terms of A_{eff}.

Statistical inference is used for statistical analysis of our FGA in Chapter 6. The results presented in Chapter 6 provide us general understanding of uniform distribution behavior of autonomous mobile nodes in a MANET.

FGA, like all GA-based approaches, uses different sets of chromosomes in every population. We introduced a new model using Markov chains to analyze its convergence speed. This model appropriately represents FGA since the state of population at time $t+1$ depends only the state of population at time t. As the genetic operators (see Section 3.1.2) that create the population at time $t+1$ out of the population at time t depend on several chance events, like randomly selecting one certain individual to be included in the mating pool or randomly mutating one certain bit in an individual, the dependence of the state of population at time $t+1$ on the state of population at time t is stochastic [97]. Our FGA is run by each mobile node as a topology control mechanism to decide its next speed and movement direction. In Chapter 7,

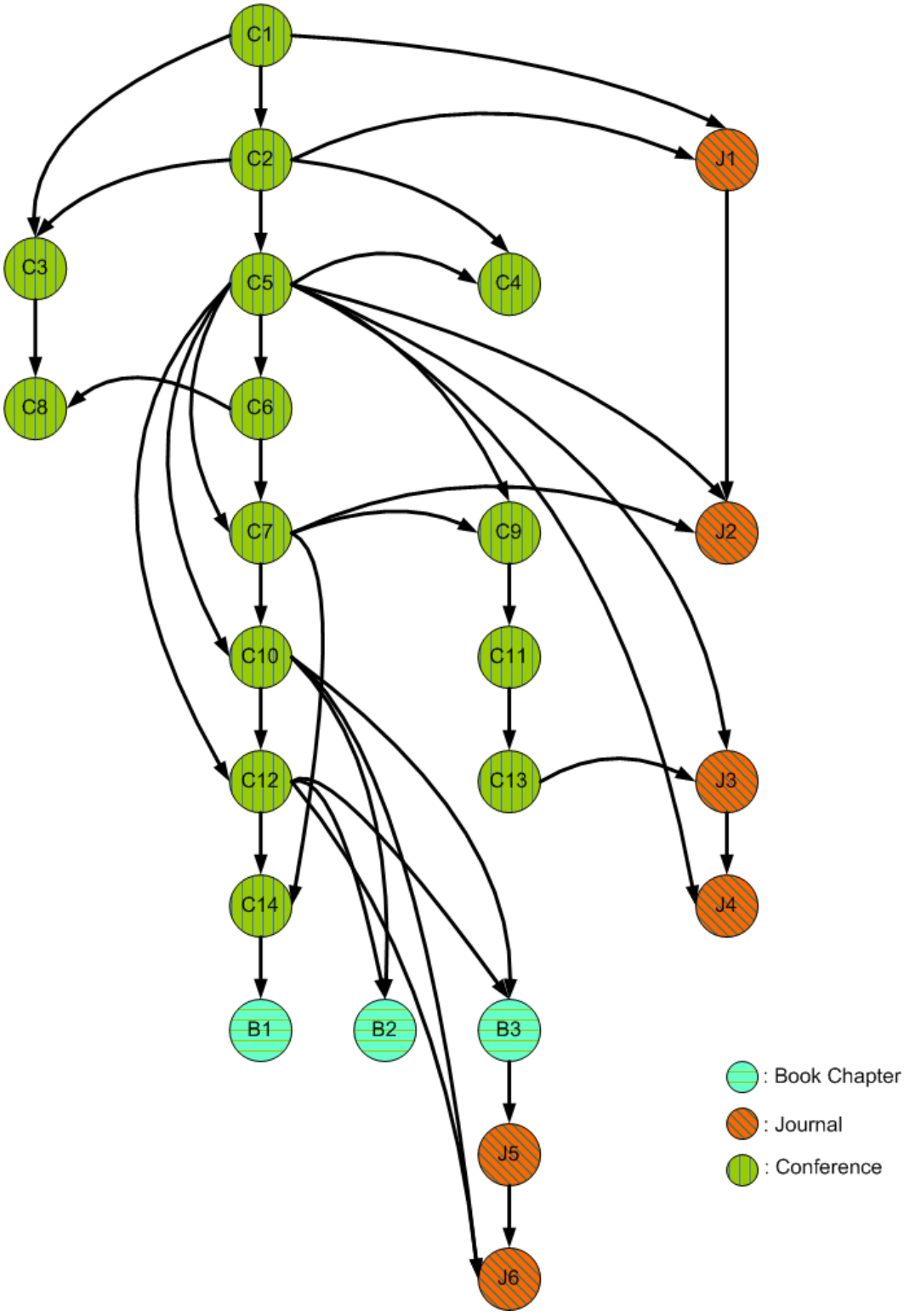


Figure 8.1: Dependency graph of our publications resulted from our GA-based topology control research

we present effectiveness and convergence speed of our GA-based topology control approach by using homogeneous and inhomogeneous Markov chains. Using Dobrushin's contraction coefficients, we show that our FGA converges to a stationary behavior (i.e., the desired state shown in Sections 7.2 and 7.3) while maximizing the area coverage and providing a fully connected network for hMC_{FGA} and iMC_{FGA} models. We also use hMC_{FGA} and iMC_{FGA} models to analyze the effects of different network parameters including communication range and the total number of mobile nodes in a MANET on the convergence of our FGA. Results from our Markov models using our simulation software indicate that the mobile nodes using shorter communication ranges require less movement and converge faster. Larger communication ranges quickly increase the A_{eff}, but unnecessarily increase the communication among the near neighbors of each node, making convergence decisions harder; as the network area becomes fully covered, many nodes will have overlapping coverage and continue to search for an optimal spatial orientation that will consume more energy.

In summary, in this dissertation, mathematical, analytical, statistical, and experimental results show that our GA-based topology control framework delivers promising results to provide fully connected network and to uniformly separate mobile nodes without any central control unit or a priori knowledge about terrain. Moreover, the MANET in which mobile nodes run our FGA is resilient to mobile node malfunction and losses.

As an extension of this work, we will concentrate on further investigation of robustness, efficiency, convergence properties, and reliability of our GA-based topology

control approach in MANETs. We expect to further decrease of time of uniform distribution of autonomous mobile nodes which will result in less battery usage.

8.2 Our Publications Based on GA-based Topology Control Research

Throughout this research, we extensively published results of our studies on GA-based topology control approach. This section presents a list of journal articles, book chapters, and conference papers, all of which were refereed by the experts in this field. A dependency graph for our publications is presented in Fig. 8.1. C1 represents the first paper we published in the International Conference on Artificial Intelligence and Pattern Recognition (AIPR07) in 2007. In Fig. 8.1, an edge directed from node Ci to Cj denotes that a conference paper Cj is based on the results of a conference paper Ci. Book chapters and journal papers are depicted as Bi and Ji, respectively. For each publication, a short synapse of its content is presented. In order to show the interdependencies among the publications, we present them in the chronological order that they were published in each category.

8.2.1 Refereed Journal Papers

J1. E. Urrea, C. S. Sahin, I. Hokelek, M. U. Uyar, M. Conner, G. Bertoli, and C. Pizzo, "Bio-inspired Topology Control for Knowledge Sharing Mobile Agents,"

Mobile Ad Hoc Networks, Elsevier, Special Issue on Bio-Inspired Computing, Vol. 7, No. 4, pp. 677-689, 2009.

This journal paper presented different GA-based approaches for knowledge sharing bio-inspired mobile nodes to obtain a uniform distribution of the nodes over a geographical terrain. With an analytical model, we showed that the best fitness value was obtained when the number of neighbors for a mobile agent was equal to the mean node degree. In this journal paper, we use the different GA-based cases introduced in conference papers C1 and C2 (in Section 8.2.3). Using the mean node degree called $\overline{N}$ from [22], our mobility model for MANET nodes is introduced in this paper.

J2. C. S. Sahin, E. Urrea, M. U. Uyar, M. Conner, G. Bertoli, and C. Pizzo, "Design of Genetic Algorithms for Topology Control of Unmanned Vehicles," International Journal of Applied Decision Sciences, Special Issue on Decision Support Systems for Unmanned Vehicles 2010 (in press).

Statistical model of autonomous mobile nodes running our FGA is defined in this journal paper. Statistical model defined Section 6 in this dissertation is used with a set of mathematical equations describing the behavior of mobile nodes in terms of random variables and their associated probability distributions. This paper uses our FGA and mobility model introduced in conference paper C5 in Section 8.2.3 and journal paper J1 above, respectively.

J3. J. Kusyk, C. S. Sahin, M. U. Uyar, E. Urrea, and S. Gundry, "Self Organization of Nodes in Mobile Ad Hoc Networks Using Evolutionary Game," Journal of

Applied Research (invited paper) (in review).

A node spreading evolutionary game, called NSEG, is introduced in this journal paper. To play NSEG with the neighboring nodes, a corresponding node only requires limited synchronization, and does not require a priori knowledge of the terrain. We also show the formal analysis of our evolutionary game to prove the convergence to Nash Equilibrium. In this journal paper, FGA which was introduced in conference paper C5 (Section 8.2.3) is used as an algorithm to select next location for an autonomous mobile node.

J4. J. Kusyk, E. Urrea, C.S. Sahin, and M.U. Uyar, "Game Theory and Genetic Algorithm Based Approach for Self Positioning of Autonomous Nodes," Ad Hoc & Sensor Wireless Networks (in review).

In this journal paper, we present our node spreading potential game (NSPG) for MANET nodes to position themselves in an unknown geographical terrain with obstacles. NSPG is a distributed and scalable game participated by autonomous nodes. The decisions about node movements are based on localized data while the best next location to move is selected by our FGA that was introduced in conference paper C5 (Section 8.2.3).

J5. C. S. Sahin, E. Urrea, and M. U. Uyar, "Self Organization for Area Coverage Maximization and Energy Conservation in MANETs," Springer Transaction in Computer Science, Special Issue on Advances in Autonomic Computing: Formal Engineering Methods for Nature-Inspired Computing Systems (in press).

We present a formal analysis of the effectiveness of our genetic algorithm using

theorems and lemmas and introduce an inhomogeneous Markov chain model to prove its convergence. The formal methods given in book chapter B3 (Section 8.2.2) are extended in this paper.

J6. C. S. Sahin, S. Gundry, E. Urrea, and M. U. Uyar, "Bio-inspired algorithms - Homogeneous Markov chain analysis on self-organizing MANETs," IEEE Transactions in Evolutionary Computations, (in review).

The stochastic behavior of FGA, like all GA-based approaches, makes it difficult to analyze the effects that various MANET characteristics have on its convergence speed. Metrically transitive homogeneous Markov chains have been used to analyze the convergence of our FGA with respect to various communication ranges of mobile nodes and also the number of nodes in various scenarios. The Dobrushin contraction coefficient explained in conference papers C10 and C12 (Section 8.2.3) is used to develop inhomogeneous Markov chain model for our GA-based topology control application.

8.2.2 Refereed Book Chapters

B1. C. S. Sahin, E. Urrea, and M. U. Uyar, "Decentralized Topology Control for Autonomous Mobile Agents in MANETs," Book Chapter for Formal and Practical Aspects of Autonomic Computing and Networking: Specification, Development and Verification, IGI Global (in press).

In this chapter, we provided detail analysis of formal (using a dynamical model)

and practical aspects of convergence properties of FGA. Our GA-based topology control approach was treated as a dynamical system in order to provide formalism to study its convergence trajectory in the space of possible populations. Discrete time dynamical system model is used for calculating the cumulative effects of our FGA operators such as selection, mutation, and crossover as a population of possible solutions evolves through generations. To demonstrate applicability of FGA to real-life problems and evaluate its effectiveness, we implemented a simulation software system and several different testbed platforms. The simulation and testbed experiment results indicated that, for important performance metrics such as normalized area coverage (NAC) and convergence rate, FGA can be an effective mechanism to deploy nodes under restrained communication conditions in MANETs operating in unknown areas. Dynamical model of our GA-based topology control application is introduced in conference paper C14 (Section 8.2.3) and is extended in this book chapter.

B2. C. S. Sahin, E. Urrea, and M. U. Uyar, "Bio-Inspired Algorithms for Self-organization in MANETs," IGI Book Chapter for Biologically Inspired Communications Networks (in press).

We presented a formal analysis of the effectiveness of our GA-based approach with respect to convergence speed and area coverage uniformity using Lyapunov stability theory. We also extend homogeneous Markov model in conference papers C10 and C12 (Section 8.2.3) to provide convergence analysis of our FGA.

B3. C. S. Sahin, S. Gundry, E. Urrea, and M. U. Uyar, "A Bio-Inspired Approach to

Self-organization of Mobile Nodes in Real-time Mobile Ad Hoc Network Applications," Book Chapter for Variants of Evolutionary Algorithms for Real-World Applications, Springer (in press).

In this book chapter, formal and practical aspects of convergence properties of our FGA are presented. Formal convergence analysis is provided by iMC_{FGA} model. The Dobrushin contraction coefficient explained in conference papers C10 and C12 (Section 8.2.3) is used to develop inhomogeneous Markov chain model for our GA-based topology control application.

8.2.3 Refereed Conference Publications

C1. E. Urrea, C. S. Sahin, M. U. Uyar, M. Conner, H. Sharif, I. Hokelek, and G. Bertoli, "Simulation Experiments for Knowledge Sharing Agents Using Genetic Algorithms in MANETs," In Proceeding of International Conference on Artificial Intelligence and Pattern Recognition (AIPR 07), pp. 369-376, July 2007.

In this paper, we introduced a set of simulation experiments for knowledge sharing agents usingGAs. In the simulation experiments, we used two different fitness functions, namely Gaussian and step distributions and discussed their performance for our objectives including uniform mobile nodes distribution.

C2. E. Urrea, C. S. Sahin, M. U. Uyar, M. Conner, H. Sharif, I. Hokelek, G. Bertoli, and C. Pizzo, "Uniform MANET Node Distribution for Mobile Agents Using Genetic Algorithms," 2007 International Conference on Genetic and Evolutionary

Methods - (GEM 07), CSREA Press, pp. 24-30, July 2007.

We present three different sets of GA-based experiments based on the mobility adjustment procedures, where mobile agents:

- constantly move (called Case 1),
- stop intermittently once a perfect fitness is obtained until other nodes move into their vicinity (called Case 2),
- stop permanently once a perfect fitness is obtained (called Case 3).

The performance of these approaches is analyzed with respect to their normalized area coverage, average fitness values, and convergence towards a uniform distribution. This paper uses the fitness functions introduced in conference paper C1 above with additional mobility adjustment procedures.

C3. C. Dogan, M. U. Uyar, E. Urrea, C. S. Sahin, I. Hokelek, "Testbed Implementation of Genetic Algorithms for Self Spreading Nodes in MANETs," 2008 International Conference on Genetic and Evolutionary Methods - (GEM 08), CSREA Press, pp. 10-16, July 2008.

A testbed using Xilinx ML310 development boards with Virtex-II Pro™ FPGA devices, desktop and laptop computers implementing GAs for self-spreading mobile nodes in MANETs is presented in this paper. The fitness function from conference paper C1 and GA (Case 2) from conference paper C2 are used in this paper.

C4. E. Urrea, C. S. Sahin, M. U. Uyar, M. Conner, I. Hokelek, G. Bertoli and C.

Pizzo, "Comparative Evaluation of Genetic Algorithms for Force-Based Self-Deployment of Mobile Agents in MANETs," 2008 International Conference on Genetic and Evolutionary Methods - (GEM 08), CSREA Press, USA, pp. 90-95, July 2008.

This paper compares two different self-deployment algorithms for mobile agents in wireless mobile ad-hoc networks based on genetic algorithms using potential field techniques called FGA and cluster-based GA called CBGA. Comparative evaluation of the two approaches are based on the metrics such as area coverage, deployment time and the distance traveled by the mobile nodes. Our FGA from conference paper C5 and the mobility model from conference paper C2 are used in this paper.

C5. C. S. Sahin, E. Urrea, M. U. Uyar, M. Conner, I. Hokelek, G. Bertoli and C. Pizzo, "Genetic Algorithms for Self-Spreading Nodes in MANETs," The 10^{th} Annual Conference on Genetic and Evolutionary Computation (GECCO), pp. 1141-1142, July 2008.

We introduce our FGA for self-spreading mobile nodes uniformly over a geographical area. Simulation experiments are provided and they had encouraging results for the performance of our FGA with respect to normalized area coverage. The mobility model in this paper is from conference paper C2.

C6. C. S. Sahin, E. Urrea, M. U. Uyar, M. Conner, I. Hokelek, G. Bertoli and C. Pizzo, "Uniform Distribution of Mobile Agents Using Genetic Algorithms for Military Applications in MANETs," IEEE International Conference on Military

Communications (MILCOM), pp.10-16, Nov 2008.

We presented a military application example such that in the observed occurrence of a threat situation, if the number of autonomous mobile nodes change with time (e.g., losing assets during an operation), the remaining agents should reposition themselves to compensate the lost in coverage and network connectivity. In this paper, we use our simulation software, which is from conference paper C5, to evaluate the effectiveness of our FGA within these types of military applications.

C7. C. S. Sahin, E. Urrea, M. U. Uyar, M. Conner, I. Hokelek, G. Bertoli and C. Pizzo, "Self-deployment of Mobile Agents in MANETs for Military Applications," Army Science Conference, pp. 1-8, Dec 2008.

In this paper, in addition to the military scenario in conference paper C6, autonomous mobile nodes randomly malfunctioned and stopped communication for a certain time to listen environment (called silent mode) in case of jamming by enemy. During silent mode, the mobile nodes did not run their GA-based topology control approach.

C8. C. Dogan, C. S. Sahin, M. U. Uyar, and E. Urrea, "Testbed for Node Communication in MANETs to Uniformly Cover Unknown Geographical Terrain Using Genetic Algorithms," The NASA/ESA Conference on Adaptive Hardware and Systems (AHS09), pp.273-280, San Francisco, CA, July 2009.

We introduce a testbed having integrated gumstix/iRobot platforms and also compared the results of this testbed with the testbed in conference paper C3.

The GA used in this article comes from conference paper C8.

C9. J. Kusyk, M. U. Uyar, E. Urrea, C. S. Sahin, M. A. Fecko, and S. Samtani, "Efficient Node Distribution Techniques in Mobile Ad Hoc Networks Using Game Theory," IEEE Intl. Conf. on Military Communications (MILCOM 2009), pp. 1-7, October 2009.

Using our distributed game (NSPG-G1) for MANET nodes to position themselves in an unknown geographical terrain to obtain the highest area coverage, we show that, combined with our FGA to determine the next best location to move, NSPG-GA1 can provide a near uniform node spreading. GA is inspired from conference papers C5 and C7.

C10. C. S. Sahin, S. Gundry, E. Urrea, M. U. Uyar, M. Conner, G. Bertoli and C. Pizzo, "Markov Chain Models for Genetic Algorithm Based Topology Control in MANETs," Applications of Evolutionary Computation, EvoApplications 2010 (EvoComNet), pp. 41-50, April 2010.

In this paper, we provide our initial formal analyzes for the convergence properties of FGA. Homogeneous Markov chain model for our FGA is introduced. We run our simulation software to study the effects of varying communication range on the convergence of our FGA. Simulation experiments indicate that the increased communication range for the mobile nodes does not result in a faster convergence. The topology control algorithm is used from conference papers C5 an C7.

C11. J. Kusyk, M. U. Uyar, E. Urrea, C. S. Sahin, "Game Theory Based Autonomous Mobile Nodes Distribution in MANETs," 33rd IEEE Sarnoff Symposium, pp. 1-5, April 2010.

Improved version of our approach in conference paper C9 is presented with the formal proof of convergence.

C12. C. S. Sahin, S. Gundry, E. Urrea, M. U. Uyar, M. Conner, G. Bertoli and C. Pizzo, "Convergence Analysis of Genetic Algorithms for Topology Control in MANETs," 33rd IEEE Sarnoff Symposium, pp. 1-5, April 2010.

We analyze the homogeneous Markov chain model for our FGA with respect to varying number of autonomous mobile nodes that is the different part of this paper than conference paper C10.

C13. J. Kusyk, E. Urrea, C. S. Sahin, M. U. Uyar, "Resilient Node Self-positioning Methods for MANETs Based on Game Theory and Genetic Algorithms," IEEE Intl. Conf. on Military Communications (MILCOM) (in press).

A new node spreading potential game, called Rel-NSPG is introduced in this paper. This game runs at each node, autonomously makes movement decisions based on localized data while the best next location to move is selected by a GA. This paper is an improved version of conference paper C11.

C14. E. Urrea, C. S. Sahin, M. U. Uyar, M. Conner, G. Bertoli and C. Pizzo, "Estimating Behavior of a GA-based Topology Control Mechanism for Self-Spreading Nodes in MANETs," IEEE Intl. Conf. on Military Communications (MILCOM)

(in press)

This paper introduces a dynamical system model for our FGA.

Bibliography

[1] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, *Mobile ad hoc networking*. Hoboken, NJ: Wiley-Interscience, 2004.

[2] P. Song, J. Li, K. Li, and L. Sui, "Researching on optimal distribution of mobile nodes in wireless sensor networks being deployed randomly," *Computer Science and Information Technology, International Conference on*, vol. 0, pp. 322–326, 2008.

[3] N. Heo, "An intelligent deployment and clustering algorithm for a distributed mobile sensor network," in *In Proceedings of the IEEE International Conference On Systems Man And Cybernetics*, 2003, pp. 4576–4581.

[4] Y. M. Chen and S.-H. Chang, "Purposeful deployment via self-organizing flocking coalition in sensor networks," *International Journal of Computer Science & Applications*, vol. 4, no. 2, pp. 84–94, 2007.

[5] E. Cayirci and T. Coplu, "Sendrom: Sensor networks for disaster relief operations management," *Journal Wireless Networks*, vol. 13, pp. 409–423, 2007.

[6] N. Heo and P. Varshney, "A distributed self spreading algorithm for mobile wireless sensor networks," *IEEE Wireless Communications and Networking (WCNC)*, vol. 3, no. 1, pp. 1597–1602, 2003.

[7] H. Wang, B. Crilly, W. Zhao, C. Autry, and S. Swank, "Implementing mobile ad hoc networking (manet) over legacy tactical radio links," in *Military Communications Conference, 2007. MILCOM 2007. IEEE*, Oct. 2007, pp. 1 – 7.

[8] M. Mitchell, *An Introduction to Genetic Algorithms*. MIT Press, Boston, MA, 1998.

[9] D. Yuret and M. de la Maza, "Dynamic hill climbing: Overcoming the limitations of optimization techniques," in *In The Second Turkish Symposium on Artificial Intelligence and Neural Networks*, 1993, pp. 208–212.

[10] F. Glover and M. Laguna, *Tabu search*. Kluwer Academic Publishers, 1997.

[11] J. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[12] G. Bekey and A. Agah, "A genetic algorithm-based controller for decentralized multi-agent robotic systems," in *In Proc. of the IEEE International Conference of Evolutionary Computing*, 1996, pp. 431–436.

[13] H. R. Miryazdi and H. Khaloozadeh, "Application of genetic algorithm to decentralized control of robot manipulators," in *ICAIS '02: Proceedings of the 2002*

IEEE International Conference on Artificial Intelligence Systems (ICAIS'02). Washington, DC, USA: IEEE Computer Society, 2002, p. 334.

[14] G. Ping-An, C. Zi-Xing, and Y. Ling-Li, "Evolutionary computation approach to decentralized multi-robot task allocation," *International Conference on Natural Computation*, vol. 5, pp. 415–419, 2009.

[15] C. S. Sahin, E. Urrea, M. U. Uyar, and S. Gundry, "A bio-inspired approach to self-organization of mobile nodes in real-time mobile ad hoc network applications," *Variants of Evolutionary Algorithms for Real-World Applications*, vol. accepted, 2011.

[16] C. S. Sahin, S. Gundry, M. U. Uyar, and E. Urrea, "Bio-inspired algorithms - homogeneous markov chain analysis of self-organizing manets," *IEEE Transactions on Evolutionary Computation*, vol. in review.

[17] Y. Mei, Y.-H. Lu, C. S. G. Lee, and Y. C. Hu, "Energy-efficient mobile robot exploration," in *In Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006 (ICRA 2006)*, May 2006, pp. 505 – 511.

[18] C. S. Sahin, E. Urrea, M. U. Uyar, M. Conner, I. Hokelek, G. Bertoli, and C. Pizzo, "Self-deployment of mobile agents in manets for military applications," in *Army Science Conference, 2008*, 2008, pp. 1 –8.

[19] ——, "Uniform distribution of mobile agents using genetic algorithms for military applications in manets," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, 16-19 2008, pp. 1 –7.

[20] C. S. Sahin, E. Urrea, and M. U. Uyar, "Self-organization of manet nodes using bio-inspired techniques," *Bio-Inspired Computing and Communication*, vol. in review.

[21] C. S. Sahin, E. Urrea, M. U. Uyar, M. Conner, I. Hokelek, G. Bertoli, and C. Pizzo, "Genetic algorithms for self-spreading nodes in manets," in *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2008, pp. 1141–1142.

[22] I. Hokelek, M. Uyar, and M. A. Fecko, "A novel analytic model for virtual backbone stability in mobile ad hoc networks," *Wireless Networks*, vol. 14, pp. 87–102, 2008.

[23] C. S. Sahin, M. U. Uyar, E. Urrea, and S. Gundry, "Self organization for area coverage maximization and energy conservation in manets," *Advances in Autonomic Computing: Formal Engineering Methods for Nature-Inspired Computing Systems*, vol. in review.

[24] C. S. Sahin, S. Gundry, E. Urrea, M. U. Uyar, M. Conner, G. Bertoli, and C. Pizzo, "Markov chain models for genetic algorithm based topology control in manets," in *EvoApplications (2)*, ser. Lecture Notes in Computer Science. Springer, 2010, pp. 41–50.

[25] ——, "Convergence analysis of genetic algorithms for topology control in manets," in *Sarnoff Symposium, 2010 IEEE*, 12-14 2010, pp. 1 –5.

[26] C. S. Sahin, E. Urrea, and M. U. Uyar, "A bio-inspired approach to self-organization of mobile nodes in real-time mobile ad hoc network applications," *Variants of Evolutionary Algorithms for Real-World Applications*, vol. accepted, 2011.

[27] G. Wang, G. Cao, and T. F. L. Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 640–652, 2006.

[28] ——, "Proxy-based sensor deployment for mobile sensor networks," in *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, 25-27 2004, pp. 493 – 502.

[29] M. Ma and Y. Yang, "Adaptive triangular deployment algorithm for unattended mobile sensor networks," *Computers, IEEE Transactions on*, vol. 56, no. 7, pp. 946 – 847, 07 2007.

[30] G. Tan, S. A. Jarvis, and A.-M. Kermarrec, "Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks," in *ICDCS '08: Proceedings of the 2008 The 28th International Conference on Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 429–437.

[31] I. Hasircioglu, H. R. Topcuoglu, and M. Ermis, "3-d path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms," in *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2008, pp. 1499–1506.

[32] R. Khanna, H. Liu, and H.-H. Chen, "Self-organisation of sensor networks using genetic algorithms," *International Journal of Sensor Networks*, vol. 1, no. 3-4, pp. 241–252, 2006.

[33] J. Chen, S. Li, and Y. Sun, "Novel deployment schemes for mobile sensor networks," *Sensors*, vol. 7, no. 11, pp. 2907–2919, 2007.

[34] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, no. 1, pp. 78 – 92, 01 2005.

[35] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, vol.2*, vol. 2, 30 2003, pp. 1293 – 1303.

[36] M. R. Pac, A. M. Erkmen, and I. Erkmen, "Towards fluent sensor networks: A scalable and robust self-deployment approach," in *Adaptive Hardware and Systems, NASA/ESA Conference on*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2006, pp. 365–372.

[37] M. Garetto, M. Gribaudo, C.-F. Chiasserini, and E. Leonardi, "A distributed sensor relocatlon scheme for environmental control," in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2007, pp. 1–10.

[38] W. Kerr, D. F. Spears, W. M. Spears, and D. R. Thayer, "Two formal fluids models for multiagent sweeping and obstacle avoidance," in *roaches to Agent-Based Systems, Third InternationalWorkshop, FAABS 2004*, vol. 3228. Springer-Verlag, 2004, pp. 111–130.

[39] J. H. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Robotics and Autonomous Systems*, pp. 171–194, 1999.

[40] I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots: Formation of geometric patterns," *SIAM Journal on Computing*, pp. 1347–1363, 1999.

[41] F. Bai and A. Helmy, "A survey of mobility modeling and analysis in wireless ad hoc networks," *Wireless Ad Hoc and Sensor Networks*, pp. 1–30, 2006.

[42] G. Lin, G. Noubir, and R. Rajaraman, "Mobility models for ad hoc network simulation," in *Proceedings of Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004, pp. 454–463.

[43] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications & Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, pp. 483–502, 2002.

[44] I. F. Akyildiz, Y.-B. Lin, W.-R. Lai, and R.-J. Chen, "A new random walk model for pcs networks," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 7, pp. 1254 –1260, Jul. 2000.

[45] I. Hokelek, "Analytic models and distributed robotics applications for mobile ad hoc networks," Ph.D. dissertation, The Graduate Center of the City Univeristy of New York, 2006.

[46] A. Gupta, F.-C. Chang, and W.-J. Huang, "Some skew-symmetric models," *Random Operators and Stochastic Equations*, vol. 10, no. 2, pp. 133–140, 2002.

[47] E. Mykytka and J. Ramberg, "Fitting a distribution to data using an alternative to moments," in *IEEE, Winter Simulation Conference*, 1979, pp. 361–374.

[48] M. Genton, "Discussion of the skew-normal," *Scandinavian Journal of Statistics*, vol. 32, pp. 189–198, 2005.

[49] M. Chiogna, "A note on the asymptotic distribution of the maximum likelihood estimator for the scalar skew-normal distribution," *Statistical Methods and Applications*, vol. 14, pp. 331–341, 2005.

[50] A. Azzalini, "A class of distribution which includes the normal ones," *Scandinavian Journal of Statistics*, vol. 12, pp. 171–178, 1985.

[51] P. McCullagh, "What is a statistical model?" *The Annals of Statistics*, vol. 30, no. 5, pp. 1225–1310, 2002.

[52] A. Eriksson, L. Forsberg, and E. Ghysels, 2004, approximating the probability distribution of functions of random variables: a new approach.

[53] E. Parzen, "Some recent advances in time series modeling," *IEEE Transactions on Automatic Control*, vol. AC-19, no. 6, pp. 723–730, Dec. 1974.

[54] C. Liew, U. Choi, and C. Liew, "A data distortion by probability distribution," *ACM, Transactions on Database Systems*, vol. 10, no. 3, pp. 395–411, Sept. 1985.

[55] J. Suzuki, "A distributed self spreading algorithm for mobile wireless sensor networks," *IEEE transactions on systems, man, and cybernetics*, vol. 4, pp. 655–659, 1995.

[56] F. Schmitt and F. Rothlauf, "On the importance of the second largest eigenvalue on the convergence rate of genetic algorithms," in *Proceedings of the 14th Symposium on Reliable Distributed Systems*, 2001.

[57] L. Barolli, A. Koyama, and N. Shiratori, "A qos routing method for ad-hoc networks based on genetic algorithm," in *DEXA '03: Proceedings of the 14th International Workshop on Database and Expert Systems Applications*. Washington, DC, USA: IEEE Computer Society, 2003, p. 175.

[58] G. Rudolph, "Local convergence rates of simple evolutionary algorithms with cauchy mutations," *IEEE Transactions on Evolutionary Computation*, vol. 1, 1998.

[59] L. Ming, Y. Wang, Yiu-Ming, and C. Xidian, "On convergence rate of a class of genetic algorithms," in *Proc. of the World Automation Congress (WAC'06), 2006*, Budapest,Hungary, 2006, pp. 1–6.

[60] H. Aytug, S. Bhattacharrya, and G. J. Koehler, "A markov chain analysis of genetic algorithms with power of 2 cardinality alphabets," *European Journal of Operational Research*, vol. 96, pp. 195–201, 1997.

[61] A. E. Nix and M. D. Vose, "Modeling genetic algorithms with markov chains," *Annals of Mathematics and Artificial Intelligence*, vol. 5, pp. 79–88, 1992.

[62] T. Nakama, "Markov chain analysis of genetic algorithms applied to fitness functions perturbed by multiple sources of additive noise," *Studies in Computational Intelligence*, vol. 149, pp. 123–136, 2008.

[63] J. S. Baras and X. Tan, "Control of autonomous swarms using gibbs sampling," *CDC. 43rd IEEE Conference on Decision and Control, 2004.*, vol. 5, pp. 4752–4757, 2004.

[64] C. A. V. Campos and L. F. M. de Moraes, "A markovian model representation of individual mobility scenarios in ad hoc networks and its evaluation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, no. 35946, p. 14, 2007.

[65] J. Jarvis and D. Shier, *Graph-theoretic analysis of finite Markov chains. In: Shier, D.R., Wallenius, K.T. (Eds.), Applied Mathematical Modeling: A Multidisciplinary Approach.* Cambridge, MA, USA: CRC Press, 1999.

[66] P. Hong and W. Xing-Hua, "The convergence rate estimation of genetic algorithms with elitist," *Chinese Science Bulletin*, vol. 42, pp. 144–147, 1997.

[67] G. Rudolph, "Local convergence rates of simple evolutionary algorithms with cauchy mutations," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 223–233, 1998.

[68] G. Rudolph and L. Xi, "Convergence rates of evolutionary algorithms for a class of convex objective functions," *Control and Cybernetics*, vol. 26, pp. 375–390, 1997.

[69] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE Transactions on Neural Networks*, vol. 5, 1994.

[70] J. Horn, "Finite markov chain analysis of genetic algorithms with niching," in *Proceedings of the Fifth International Conference on Genetic Algorithms*. Morgan Kaufmann, 1993, pp. 110–117.

[71] Z. Jiangshe, X. Zongben, and L. Yee, "Global annealing genetic algorithm and its convergence analysis," *Science in China Series E: Technological Sciences*, vol. 40, pp. 414–424, 1997.

[72] C. F. Huang, "A markov chain analysis of fitness proportional mate selection schemes in genetic algorithm," in *GECCO '02: Proceedings of the Genetic and Evolutionary Computation Conference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, p. 682.

[73] N. Takéhiko, "Markov chain analysis of genetic algorithms in a wide variety of noisy environments," in *GECCO '09: Proceedings of the 11th Annual conference*

on Genetic and evolutionary computation. New York, NY, USA: ACM, 2009, pp. 827–834.

[74] G. Winkler, *Image Analysis, Random Fields and Markov Chains Monte Carlo Methods*. Springer-Verlag Berlin Heidelberg, 2006.

[75] R. Dobrushin, "Central limit theorem for nonstationary markov chains. ii," *Teor. Veroyatnost Primenen*, pp. 365–425, 1956.

[76] N. Z. Chaiyaratana, "Recent developments in evolutionary and genetic algorithms: theory and applications," in *Genetic Algorithms In Engineering Systems:Innovations And Applications, 2-4 Sept. 1997. GALESIA 97. Second International Conference On (Conf. Publ. No. 446)*, 1997, pp. 270 – 277.

[77] T. Shinchi, M. Tabuse, T. Kitazoe, and A. Todaka, "Khepera robots applied to highway autonomous mobiles," *Artif. Life Robotics*, 2003.

[78] V. Gesu, B. Lenzitti, G. L. Bosco, and D. Tegolo, "A distributed architecture for autonomous navigation of robots," in *Proc. of the Fifth IEEE Int. Workshop on Computer Architectures for Machine Perception*, 2000.

[79] X. Ma, Q. Zhang, and Y. Li, "Genetic algorithm-based multi-robot cooperative exploration," in *IEEE Int. Conference on Control and Automation*, June 2007, pp. 1018 – 1023.

[80] Y. Hu and S. Yang, "A knowledge based genetic algorithm for path planning of a mobile robot," in *Proc. of the 2004 IEEE Int. Conference on Robotics & Automation*, 2004.

[81] R. Leigh, S. Louis, and C. Miles, "Using a genetic algorithm to explore a*-like path finding algorithms," in *IEEE Symposium on Computational Intelligence and Games*. Honolulu, HI, USA: IEEE, 2007, pp. 72 – 79.

[82] A. F. Winfield, "Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots," *Distributed Autonomous Robotic Systems*, vol. 4, pp. 273–282, 2000.

[83] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proceedings of the International Conference on Distributed Autonomous Robotic Systems*, 2002, pp. 299–308.

[84] F. Tang and L. E. Parker, "Asymtre: Automated synthesis of multi-robot task solutions through software reconfiguration," in *In Proceedings of IEEE International Conference on Robotics and Automation*, 2005, pp. 1513–1520.

[85] A. Franchi, L. Freda, and M. Vendittelli, "A randomized strategy for cooperative robot exploration," in *Robotics and Automation, 2007 IEEE International Conference on*, 10 2006, pp. 768 –774.

[86] R. L. Stewart and R. A. Russell, "A distributed feedback mechanism to regulate wall construction by a robotic swarm," *Adaptive Behavior*, vol. 14, pp. 21–51, 2006.

[87] M. A. Joordens, T. Shaneyfelt, K. Nagothu, S. Eega, A. Jaimes, and M. Jamshidi, "Applications and prototype for system of systems swarm

robotics," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Singapore, 2008, pp. 2049–2055.

[88] S. Xue and J. Zeng, "Sense limitedly, interact locally: the control strategy for swarm robots search," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, 2008, pp. 402–407.

[89] J. Werfel, "Robot search in 3d swarm construction," in *Proceedings of the First International Conference on Self-Adaptive and Self-Organizing Systems*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 363–366.

[90] M. Saska, M. Macas, L. Preucil, and L. Lhotska, "Robot path planning using particle swarm optimization of ferguson splines," in *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation*, Prague, 2006, pp. 833–839.

[91] E. Tuci, R. Groß, V. Trianni, F. Mondada, M. Bonani, and M. Dorigo, "Cooperation through self-assembly in multi-robot systems," *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 2, pp. 115–150, 2006.

[92] Z. Li, B. Xu, L. Yang, J. Chen, and K. Li, "Quantum evolutionary algorithm for multi-robot coalition formation," in *GECCO '09: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2009, pp. 295–302.

[93] J. Soto and K.-C. Lin, "Using genetic algorithms to evolve the control rules of a swarm of uavs," *International Symposium on Collaborative Technologies and Systems*, pp. 359–365, 2005.

[94] A. Naghsh, J. Gancet, A. Tanoto, and C. Roast, "Analysis and design of human-robot swarm interaction in firefighting," in *Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication*, 2008, pp. 255–260.

[95] A. Winfield, C. Harper, and J. Nembrini, "Towards the application of swarm intelligence in safety critical system," in *Proceedings of the 1st Institution of Engineering and Technology International Conference on the System Safety*, 2006, pp. 89–95.

[96] T. ruey Hsiang, E. M. Arkin, M. A. Bender, S. P. Fekete, and J. S. B. Mitchell, "Online dispersion algorithms for swarms of robots," in *Proceedings of the 19th Annual Symposium on Computational Geometry*, 2003, pp. 382–383.

[97] H. Dawid, *Adaptive Learning by Genetic Algorithms: Analytical Results and Applications to Economic Models*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1996.

[98] S. Misra, I. Woungang, and S. C. Misra, *Guide to Wireless Ad Hoc Networks*. Springer Publishing Company, Incorporated, 2009.

[99] R. Hekmat, *Ad-hoc networks: Fundamental properties and network topologies*. The Netherland: Springer, 2006.

[100] C. S. Sahin, E. Urrea, M. U. Uyar, M. Conner, I. Hokelek, G. Bertoli, and C. Pizzo, "Bioinspired topology control for knowledge sharing mobile agents," *Mobile Ad Hoc Networks, Elsevier, Special Issue on BioInspired Computing*, vol. 7, no. 4, pp. 677–689, 2009.

[101] ——, "Design of genetic algorithms for topology control of unmanned vehicles," *Special Issue of the Int. J. of Applied Decision Sciences on Decision Support Systems for Unmanned Vehicles, in press*, 2010.

[102] M. Kawski, "Applications of homogeneity to nonlinear adaptive control," in *Computation and Control II: Proceedings of the Second Bozeman Conference*. Boston, MA: Birkhauser Boston, 1990, pp. 225–236.

[103] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ: Prentice Hall, 1996.

[104] L. Barreira and Y. B. Pesin, *Lyapunov exponents and smooth ergodic theory*. Providence, RI: American Mathematical Society, 2000.

[105] P. Dang, F. L. Lewis, and D. O. Popa, "Dynamic localization of air-ground wireless sensor networks," in *Control and Automation, 2006. MED '06. 14th Mediterranean Conference on*, June 2006, pp. 1 –7.

[106] D.-W. Lee, S.-W. Seo, and K.-B. Sim, "Online evolution for cooperative behavior in group robot systems," *International Journal of Control, Automation, and Systems*, pp. 282–287, 2008.

[107] Z. Qu, *Cooperative control of dynamical systems: Applications to autonomous vehicles*. London, UK: Springer-Verlag, 2009.

[108] D. W. Gage, "Command control for many-robot systems," *Unmanned Systems*, pp. 28 –34, Fall 1992.

[109] Xilinx, "Ml310 user guide, virtex-ii pro embedded development platform," February 1, 2007. [Online]. Available: http://www.xilinx.com/support/documentation/boards_and_kits/ug068.pdf

[110] *iRobot developer and educators site*, http://store.irobot.com/deved/index.jspl.

[111] *Gumstix developer site*, Users manual, http://www.gumstix.net/Documentation/109.html.

[112] *Workstation Users Manual, Workstation 6.5*, VMware Inc., 2008.

[113] C. Dogan, M. Ü. Uyar, E. Urrea, C. S. Sahin, and I. Hökelek, "Testbed implementation of genetic algorithms for self spreading nodes in manets," in *GEM*, 2008, pp. 10–16.

[114] C. Dogan, C. S. Sahin, M. U. Uyar, and E. Urrea, "Testbed for node communication in manets to uniformly cover unknown geographical terrain using genetic algorithms," *Adaptive Hardware and Systems, NASA/ESA Conference on*, vol. 0, pp. 273–280, 2009.

[115] W. M. Bolstad, *Introduction to bayesian statistics*. Hoboken, NJ: Wiley-Interscience, 2007.

[116] T. Dean, S. Leach, and H. Shatkay, "Graphical models for learning dynamical systems," 1996.

[117] M. H. Degroot, *Probability and statistics*. Reading, MA: Addison Wesley, 1986.

[118] D. R. Bickel, "Robust and efficient estimation of the mode of continuous data: the mode as a viable measure of central tendency," *Journal of Statistical Computation and Simulation*, vol. 73, no. 12, pp. 899–912, Dec. 2003.

[119] R. G.Staudte and S. J. Sheather, *Robust estimation and testing.* New York, NY: Wiley, 1990.

[120] P. J. Bickel and E. L. Lehmann, "Descriptive statistics for nonparametric models: Iii. dispersion," *The Annals. of Statistics*, vol. 4, no. 6, pp. 1139–1158, Nov. 1976.

[121] D. Levin, "Point estimation for parametric families of probability distributions."

[122] P. Bickel and K. Doksum, *Mathematical statistics, basic ideas and selected topics, Vol.I.* Upper Saddle River, NJ: Prentice Hall, 2001.

[123] G. F. Lawler, *Introduction to Stochastic Process.* Chapman and Hall, NY, USA, 1955.

[124] Y. Gong and W. Xu, *Machine Learning for Multimedia Content Analysis (Multimedia Systems and Applications).* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[125] B. P. Palka, *An introduction to complex function theory.* New York, NY: Springer-Verlag, 1990.