Game-theoretic and Bio-inspired Techniques for Self-positioning Autonomous Mobile Nodes

by

JANUSZ KUSYK

A dissertation submitted to
the Graduate Faculty in Computer Science
in partial fulfillment of the requirements for the degree
of Doctor of Philosophy, The City University of New York

2012

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

Prof. M. Ümit Uyar
Electrical Engineering, City College, CUNY
Computer Science, The Graduate Center, CUNY

_____ _____

Date Chair of Examining Committee

Prof. Ted Brown
Computer Science, The Graduate Center, CUNY

_____ _____

Date Executive Officer

Supervisory Committee

Prof. Ravindran Kaliappa
Computer Science, City College, CUNY

Prof. Simon Parsons
Computer and Information Science, Brooklyn College, CUNY

Dr. Eric van den Berg
Telcordia Technologies, NJ

Abstract

Game-theoretic and Bio-inspired Techniques
for Self-positioning Autonomous Mobile Nodes

by

Janusz Kusyk

Autonomous mobile nodes that position themselves over an unknown terrain can ameliorate many of the problems which mobile ad hoc networks (MANETs) face. Achieving good spatial placement of mobile agents leads to superior network topology with improved area coverage, reduced power consumption, enhanced spectrum utilization, and simplified routing procedures. However, autonomous decision-making by nodes may also increase uncooperative and selfish behavior by these independent agents. Since it is impractical in MANETs to sustain complete and accurate information at each node about the entire network layout and decision of an individual node about its position should only be based on local information with limited coordination among agents. These characteristics recommend game theory (GT) as a tool for modeling, analyzing, and designing many MANET applications. At the same time, biologically inspired computation techniques such as genetic algorithms (GAs) can be used for finding desirable solutions in a prohibitively large search space and, in the case of MANETs, reduce the computational complexity needed for a node to determine its next location.

We introduce several novel approaches for autonomous MANET nodes to distribute themselves uniformly over a dynamically changing environment without a centralized controller or *a priori* information about the deployment terrain or the state of other mobile agents. Our methods combine concepts from traditional GT, evolutionary game the-

ory, and bio-inspired algorithms to effectively and efficiently guide autonomous MANET nodes in finding the best positions. We show that myopic actions of each individual node lead the entire network towards a stable and uniform distribution. We present formal analysis of our methods, and prove their important properties including convergence, area coverage, and uniformity characteristics.

We developed a Java-based modeling platform to simulate performances of mobile networks, which allows for real-time visualization of ongoing network dynamics and collection of all critical data for evaluation purposes. Our analysis and experimental results demonstrate that GT and GA can be successfully combined for autonomous node placement to provide useful and resilient methods for optimizing network topology.

Acknowledgments

One of the most valuable lessons I learned while being a graduate student is that without the support and encouragement of my colleagues, family, and friends, my academic goals would be much harder to achieve. Fortunately, I have always had the luck to have a multitude of wonderful people around.

First, I would like to thank my academic supervisor M. Ümit Uyar, whose broad knowledge, experience, and diligence will keep inspiring me for years to come. Your guidance and lucid explanations made even the most distant ideas accessible and nothing I present in this thesis would be possible without you. Also, your ardor for academic work and research combined with your patience and devotion with students make you a model professor—I feel privileged being one of your students.

This manuscript has also been greatly strengthened by suggestions, supports, and encouragements from my friends and colleagues in the City College of the City University of New York: Stephen Gundry, Cem Şafak Şahin, Elkin Urrea, and Jianmin Zou and the Graduate Center of the City University of New York: Art Diky, Emir Ganic, Erdal Kose, Karen Kletter, Cefan Daniel Rubin, and Suzanne Tamang. It is hard to overstate the impact of your aid and enthusiasm on this manuscript.

I would also like to give my gratitude to the faculty of Departments of Computer Science, the Graduate Center, CUNY and Brooklyn College, CUNY for their tireless involvement into my education, to my employer Stuart Pivar for being so concerned about me, and to my dear friends Magdalena Baczewska, Marian Jedras, Iwona Pietruczuk, and Glynn Wood for always alleviating any of my slightest concerns.

Initial stages of this research was supported by U.S. Army CECOM (Communications Electronics Command) contracts W15P7T-06-C-P217 and W15P7T-09-C-S021

and by the National Science Foundation grants ECS-0421159 and CNS-0619577.The contents of this document represent the views of the authors and are not necessarily the official views of, or are endorsed by, the U.S. Government, Department of Defense, Department of the Army or the U.S. Army Communications-Electronics RD&E Center.

And last but not least, I would like to express huge gratitude to my mom Anna Kusyk, dad Jan Kusyk, and my siblings Adam, Andrzej, and Ania for never stopping believing in me and always backing my educational aspirations, as well as to my wife Ilona Kusyk for her patience and understanding and my son Dorian Kusyk, who has been my greatest inspiration since he was born.

Contents

List of Tables

List of Algorithms

List of Figures

List of Abbreviations

3D-GA	Three-dimensional Genetic Algorithm
ADT	Average Distance Traveled
AG	Auction Game
AP	Access Point
AUV	Autonomous Underwater Vehicles
BG	Bayesian Game
BioGame	Node-spreading Bio-inspired Game
BNE	Bayes-Nash Equilibrium
CE	Competitive Equilibrium
CN	Cognitive Network
CR	Cognitive Radio
DSS	Dynamic Spectrum Sharing
DSSG	Dynamic Spectrum Sharing Game
EGT	Evolutionary Game Theory
ESS	Evolutionary Stable Strategy

FGA	Force-based Genetic Algorithm
GA	Genetic Algorithm
GT	Game Theory
HUV	Holonomic Unmanned Vehicles
MANET	Mobile Ad Hoc Network
MASON	Multi-agent Simulation of Networks
NAC	Network Area Coverage
NBG	Nash Bargaining Game
NBS	Nash Bargaining Solution
NE	Nash Equilibrium
NSBG	Node-spreading Bio-inspired Game
NSEG	Node-spreading Evolutionary Game
NSPG	Node-spreading Potential Game
OSI model	Open Systems Interconnection Model
PO	Pareto Optimal
POA	Price of Anarchy
PU	Primery User

Rel-NSPG	Resilient Node-spreading Potential Game
RSBG	Rubinstein-Stahl Bargaining Game
RW	Random Walk
SDR	Software-defined Radio
SG	Stackelberg Game
SINR	Signal to Interference Plus Noise Ratio
SPG	Shortest Path Graph
SPP	Stackelberg Pricing Problem
SU	Secondary User
TC	Topology Control
$\mathscr{U}_A$	Voronoi Cell Area Based Uniformity Measure
$\mathscr{U}_C$	Voronoi Cell Center of Mass Uniformity Measure
UWSN	Underwater Sensor Networks
VCG	Vickrey-Clarke-Groves Auction

1 Introduction

A mobile ad-hoc network (MANET) topology is the basic infrastructure for many important networking tasks such as routing, data collection, and information exchange, thus having far-reaching impact on the entire network performance. One of the factors influencing a MANET topology is the physical distribution of its nodes in an area of deployment. Achieving a better placement of nodes may promote low-power consumption, simplified routing procedures, and better spectrum utilization with stable network throughput. Important objectives for optimal distribution of nodes are connectivity among the mobile agents, uniformity of the node distribution, and maximization of the total area covered by all nodes.

Mobile nodes that autonomously position themselves over an unknown terrain can contribute to the improvement of an entire network's performance. However, autonomous decision-making processes may also promote uncooperative and selfish behavior of independent agents. Furthermore, in a dynamically changing environment, which is a common characteristic of MANETs, it is often impractical to sustain complete and accurate information about each node for every moderate size network. Therefore, decision regarding movement direction and speed of an individual node should be based on only its local information and require a limited coordination among agents. These features make game theory (GT) a promising tool to model, analyze, and design many MANET applications.

GT is a framework for analyzing behavior of a rational players in strategic

situations where the outcome depends on actions of all participants. It is a well-researched area of applied mathematics with a broad set of analytical tools readily applied to many areas of computer science. When designing a MANET using a game-theoretical approach, incentives and deterrents can be built into the game structure to guarantee an optimal or near-optimal solution while eliminating a need of broad coordination and cooperation among nodes.

As in many optimization problems with a prohibitively large domain for an exhaustive search, finding the best new location for a node that satisfies certain requirements (e.g., a uniform distribution over a geographical terrain, the best strategic location for a given set of tasks, or efficient spectrum utilization) is difficult. Traditional search algorithms for such problems look for a result in an entire search space by either sampling randomly (e.g., random walk (RW)) or heuristically (e.g., hill climbing and gradient decent). However, they may arrive at a local maximum point or miss the group of desired solutions altogether. Genetic algorithms (GAs) are promising alternatives for problems where heuristic or random methods cannot provide satisfactory results. GAs are evolutionary algorithms working on a population of possible solutions instead of a single one. As opposed to an exhaustive or random search, GAs look for the best genes (i.e., the best solution or an optimum result) in an entire problem set using a fitness function to evaluate the performance of each chromosome (i.e., a candidate solution). In our approaches, various GAs are used to speed up a network convergence, improve precision of finding new locations by nodes, and to reduce the computational complexity of selection processes.

Because our approaches are partially based on GT and GAs, we will refer to a node as a player or a mobile agent, interchangeably, and its location, represented by a chromosome, will be often identified as strategies. Following the tradition of the publications in area of GT, we also refer to a player in the feminine form without any malevolent intentions.

In this thesis, we present three distributed and scalable approaches for autonomous MANET nodes to place themselves over an unknown geographical terrain without a centralize controller. Node-spreading potential game (NSPG) for MANET nodes to position themselves in an unknown geographical terrain with obstacles is an approach that combines an ordinal potential game and GT to provide a resilient approach for autonomous nodes. In our node-spreading evolutionary game (NSEG), traditional GT, evolutionary game theory (EGT), and force-based genetic algorithm (FGA) are combined to assist a node in finding a new location. Our node-spreading bio-inspired game, called BioGame, applies a FGA and spatial game to determine the best new location to move.

Autonomous mobile nodes run NSPG, NSEG, or BioGame locally and make movement decisions independently. Nodes require only wireless communication and sensing capability within a limited range, and make their decisions solely based on information from the neighborhood of their nearest neighbors. In this chapter, we introduce our node-spreading techniques separately and present the motivation behind their developments. For each of our node self-positioning techniques, the fitness functions used by GA and FGAs are different and employ distinct mechanisms for nodes to find their next profitable locations. However,

as shown by our experimental results and formal analysis, all of our biologically inspired computations provide effective mechanisms for discovering improved new positions in a mobile agent's surrounding.

1.1 Our NSPG

Our NSPG [34] provides a method for MANET nodes to position themselves in an unknown geographical terrain with obstacles. NSPG is a distributed and scalable game participated by autonomous mobile nodes. The decisions about node movements are based on localized data while the best next location to move is selected by a GA. Our approach requires only a limited synchronization among the closest neighbors of a player and does not demand *a priori* knowledge of environment. We show here that NSPG belongs to the class of ordinal potential games, which assures that it has a Nash equilibrium solution. In order to reduce the computational complexity associated with finding a new position, each node in NSPG runs our GT to determine its best move. The GT uses the payoff function of an individual player to evaluate the *fitness* of each of its possible new positions. A resilient node-spreading potential game (Rel-NSPG), which is an extension of NSPG, was first presented by us in [35]. We demonstrate in Chapter 4 that networks with nodes running Rel-NSPG respond gracefully to both decreases in number of MANET nodes due to equipment malfunction and hostile activities or increases due to the redeployment of additional resources.

1.2 Our NSEG

In [34], we combined GT and our FGA to create a topology control mechanism, which we called NSPG. Our experimental data and formal analysis demonstrated that NSPG can be an effective method for topology control of autonomous MANET nodes. Although the results achieved with NSPG were satisfactory, the number of steps required to reach near-uniform distribution seemed to offer an opportunity for improvement.

In NSEG, we improved our previous results from NSPG for convergence speed of MANET nodes by combining FGA, traditional GT, and evolutionary GT, into a new approach called NSEG. The goal for both NSPG and NSEG is to maximize the area covered by all nodes while providing a uniform node distribution. However, the bio-inspired techniques used in each are quite different as our experiences in NSPG led us to explore different approaches in this domain. In NSPG, player's potential function determines the quality of her possible locations that are found by GA as a new position to move. In contrast, in NSEG, FGA uses its own distinct fitness function to find a set of candidate solutions.

Each node in NSEG first runs FGA to determine a set of possible next locations, which are then quantized to reflect preferences over its Moore neighborhood [48] (i.e., the logical cell that it currently occupies and all eight adjacent cells). Our FGA takes only into account the current position of the neighboring nodes to find the next locations to move. However, NSEG, combining FGA with game theory, can find even better locations since it uses additional information

regarding the payoffs of the neighbors. After determining a set of candidate new positions, the player computes an expected payoff by means of a spatial game set up among her and her neighbors. In NSEG, if expected improvement resulting from moving according to the probabilities assigned by FGA is better than moving to the best location in its Moore neighborhood, the node moves with accordance to the FGA outcome. Otherwise, it simply moves to the cell that assures a better initial position before the next step, thus implementing the evolutionary process of *learning by imitation*. The starting deployment of nodes in NSEG may result in multiple occupants placed in the same logical cell and the goal for each node is to position itself in order to obtain a high deployment terrain coverage by all nodes and to achieve uniform distribution while keeping the network connected. This goal is obtained by a topology without disconnected nodes and where each square logical cell is occupied by at most one agent. We used the concept of evolutionary stable strategy to show that the optimal network topology of NSEG is evolutionary stable and once reached, guarantees network stability.

1.3 Our BioGame

While NSEG successfully reduced the number steps required to achieve a satisfactory final distribution, we again came across avenues for possible improvement. Intuitively we sought to provide a way for the nodes to distribute themselves in a hexagonal lattice that might be more suitable for real-life applications.

We developed our BioGame [32] to combine GAs and traditional game theory to provide a fully decentralized and scalable technique for self-positioning autonomous MANET nodes that do not require synchronization among them. As in the earlier mechanisms, each mobile node runs BioGame autonomously to make movement decisions based solely on local data. First, our FGA finds a set of preferred next locations to move. Next, favorable locations identified by FGA are evaluated by the spatial game set up among a mobile node and its current neighbors. One of the novel aspects of our BioGame is the *force scaling function* used to determine the desired number of neighbors within the communication radius of a node. Our force scaling function replaced the *expected node degree* used in the fitness function of NSEG. Unlike the expected node degree, the force scaling function eliminates the need to know the number of nodes in the network and deployment terrain. If the desired number of neighbors is small, our fitness function will promote a sparsely connected network topology where the nodes have a limited number of neighbors and with reduced overlapping communication areas. On the other hand, when the desired node degree is large, agents running our BioGrame will create a densely packed network, where each agent has multiple neighbors. We prove the basic properties of BioGame, including its convergence, uniformity, and area coverage characteristics. We show that our BioGame outperforms FGA and successfully distributes mobile nodes over an unknown geographical terrain without requiring global network information or a synchronization among nodes.

Instead of the square grid deployment space used in our NSEG, our novel

BioGame operates over a two-dimensional hexagonal lattice where the distance that any node can move at one step extends over multiple logical cells. Some of the benefits resulting from representing the deployment area by a logical hexagonal grid are that: *(a)* a hexagon more closely resembles an idealized circular propagation pattern of node's antenna and thus provides a more realistic representation of its communication range, *(b)* distance between the centers of each cell and all adjacent cells is constant, offering accurate representation of nodes movement distances, and *(c)* increases in node communication range, maximum move distance, or coverage area can be more accurately represented by an appropriate cluster containing multiple hexagonal cells.

The main difference between BioGame and NSPG is exhibited in the fact that NSPG uses GA to find possible new locations using the payoff function of a player. These locations are then evaluated by our ordinal potential game to determine their validity and future impact on network topology. Contrary to NSPG, each BioGame node runs FGA to find a set of promising new positions, which decreases a computational space for our spatial game that utilizes further information about the neighborhood of the moving node to select the most promising new location. Another difference between BioGame and NSEG is that BioGame does not require any synchronization among nodes to improve the network area coverage while NSEG expects a limited synchronization among the near neighbors that are located in the same logical square cell.

1.4 Thesis Outline

The rest of this thesis is organized as follows: Chapter 2 provides a concise overview of research in areas of node self-positioning, GT-based methods used to alleviate several MANET problems, various network-related GA applications, and a wide-range of possible applications for Voronoi tessellations. Fundamental concepts of GT, EGT, and GA and a notation for our node self-positioning techniques are presented in Chapter 3. This chapter also presents selected applications of GT to wireless ad hoc networks as well as our methods for evaluating the performance of self-positioning autonomous agents.

Chapter 4 explains our NSPG and defines the payoff function for each player as well as the global potential function reflecting changes in strategies of all network participants. In Chapter 4, we also formally analyze NSPG's properties and evaluate the results of experiments modeling its performance. Chapter 5 describes in detail our NSEG. It presents the process that a player uses to evaluate their current location, set up a spatial game with their neighbors, and decides where to move to improve their position. We analyze the convergence of our NSEG and demonstrate the experimental results of its simulation in Sects. 5.4 and 5.5 of Chapter 5, respectively. Detailed description of FGA and GT parts of BioGame are provided in Chapter 6. Sections 6.2.1–6.2.4 in Chapter 6 evaluate the performance of BioGame with respect to area coverage, average distance traveled by a node, uniform distribution of mobile agents, and BioGame's operation when the network's population fluctuate due to hostile activity. In Chapters 7

and 8, we compare the performance of NSPG, NSEG, and BioGame and present concluding remarks for our research, respectively.

Chapter 9 lists and briefly describe our peer-reviewed book chapters, journal, and conference publications and illustrate the relationship among them. A list of abbreviations is provided at the beginning of this thesis to aid the reader throughout it.

1.5 Support

Initial stages of this research was supported by U.S. Army CECOM (Communications Electronics Command) contracts W15P7T-06-C-P217 and W15P7T-09-C-S021 and by the National Science Foundation grants ECS-0421159 and CNS-0619577. The contents of this document represent the views of the authors and are not necessarily the official views of, or are endorsed by, the U.S. Government, Department of Defense, Department of the Army or the U.S. Army Communications-Electronics RD&E Center.

2 Brief Literature Review

A wide range of MANET problems can be alleviated by various GT-based applications. GT has been successfully used to control dynamic spectrum sharing (Huang et al. [24]; Ji and Liu [28]; Pan et al. [52]), routing (van Hoesel [69]; Gairing [15]), and a network topology (Komali et al. [30]; Eidenbenz et al. [12]). Several additional GT applications to wireless networks are analyzed in the work of Mackenzie and DeSilva [45] and Sect. 3.1 in Chapter 3 of this thesis presents survey of GT-based solutions to numerous networking problems.

Evolutionary game theory applications to wireless networks address issues of efficient routing and spectrum sharing. Seredynski and Bouvry [59] propose a game-based packet forwarding scheme. By employing an EGT model, cooperation could be enforced in the networks where selfishly motivated nodes base their decisions on the outcomes of a repeatedly played 2-player game. Applications of EGT to solve routing problems have been investigated by Fischer and Vöcking [13], where the traditional GT assumptions are replaced with a lightweight learning process based on players' previous experiences. Wang et al. [71] investigate the interaction among users in a process of cooperative spectrum sensing as an evolutionary game. They show that by applying the proposed distributed learning algorithm, the population of secondary users converges to the stable state.

GAs are popular in many implementations for distributed robotic and network routing applications. In [7], Chen and Zalzala present a genetic approach

for motion planning of mobile robots and Shinchi et al. [60] simulate a GA-based model for autonomous Khepera robots to safely move on a highway. Ahn and Ramakrishna [2] and Barolli et al. [4] applied GA to solve network routing problems. The FGA was modeled by Sahin et al. in [55, 56, 57] and Urrea et al. [68]. In FGA, each mobile node finds the best next location such that the artificial forces applied by its neighbors are minimized. It has been shown by Sahin et al. [56] that FGA is an effective tool for a set of conditions that may be present in military applications (e.g., avoiding arbitrarily placed obstacles over an unknown terrain, loss of mobile nodes, and intermittent communications).

Managing the movement of nodes in network models where each node is capable of changing its own spatial location has been addressed by employing diverse methods. For example, Howard et al. [22] use a concept of potential fields for an effective network area coverage and Cortes et al. [8] present the Lloyd-based algorithm to control a group of autonomous vehicles. Several GA-based techniques for decentralized topology control mechanisms were studied by Sahin in his comprehensive work on the subject [54]. To our best knowledge, there is no prior research work combining GT and GA for MANET nodes to position themselves autonomously over the deployment terrain.

Voronoi tessellation has been widely applied to analyze biological cellular models and territorial behavior of animals. Some of the other successful uses of Voronoi diagrams are in the areas of computer graphics, pattern analysis, clustering, and sensor networks. Voronoi-based quality measures for point distribution in a given area are outlined in [49] by Nguyen et al. while applications

of centroidal Voronoi tessellation method for optimal distribution of resources are examined by Du et al. in [10]. Section 3.7 in Chapter 3 presents two node-distribution gauging techniques that utilize various quantities of Voronoi cells.

3 Background

In this chapter, we present fundamental GT, GAs, EGT concepts, and the notation used in this manuscript. The interested reader can find extensive and rigorous analysis of GT in the book by Fudenberg and Tirole [14]; the fundamentals of evolutionary games are described in the books by Smith [63] and Weibull [72]. GAs were popularized by the pioneering work from John H. Holland [21] in which he presented a comprehensive foundation of GAs leading to a flourishing of research in this area. This chapter also includes a formal definition of our new quantitative techniques to assess performance of MANET nodes with respect to uniform distribution, total terrain covered by communication areas of all nodes, and distance traveled by each node before a desired network topology is reached.

3.1 Game Theory

Any game can be classified as being a representative of a few fundamental game types. It can be a *cooperative* game, where players are allowed to form coalitions, or a *non-cooperative* game, in which players try to act rationally only with respect to their own objectives and irrespectively to other players' incentives. The type of payoff assigned to each player at the end of the competition can distinguish games as the *pure competition* or *general sum* games. Constant-sum and its subclass zero-sum games are pure competition games where sum of all players' payoffs for each outcome is the same constant value or 0, respectively, and the gain for one player causes the proportional decrease for other players' pay-

offs. In general sum games, players' payoffs are not limited by any constraints and can provide mutual improvement for all players.

If some players are uncertain about the game structure (i.e., uncertain of opponents' possible strategies and/or payoffs), they are participating in a game of *incomplete information*. A *perfect information* game is one where history of other players' actions up to this point is known and in an *asymmetric* game, one player's knowledge is broader than the others.

The most popular games applied to ad hoc networks are non-cooperative competitions that can be divided into *(a)* normal-form and *(b)* extensive-form games. Normal-form games are part of the self-positioning methods for autonomous MANET nodes presented by us and are introduced in Sect. 3.1.1 of this chapter. In extensive-form games, each player can have more than one decision point (move), the players make their moves in turns, and common knowledge of all previous actions is often assumed in its implementations for solving wireless ad hoc network problems. Section 3.1.2 introduces basic concepts of extensive-form games and Sect. 3.1.3 outlines a close-related concept of sequential games.

In following Sect. 3.2, we present the recent GT applications to the area of dynamic spectrum sharing (DSS). Section 3.3 analyzes the most popular GT approaches to resolve network topology control (TC) problems. Finally, Sect. 3.4 addresses other wireless network problems for which classical GT solutions have been researched.

3.1.1 Normal-form Games

A game in a normal-form is defined by a nonempty and finite set I of m players, a strategy profile space S, and a set U of payoff functions. We indicate an individual player as $u_i \in I$ and each player u_i has an associated set S_i of possible strategies from which, in a *pure strategy* normal-form game, she chooses a single strategy $s_i \in S_i$ to be realized. A game strategy profile is defined as a vector $\boldsymbol{s} = \langle s_1, s_2, \ldots, s_m \rangle$. If $\boldsymbol{s}$ is a strategy profile played in a game, then $u_i(\boldsymbol{s})$ denotes a payoff function defining the profit for player u_i as an outcome of $\boldsymbol{s}$ being played. It is convenient to single out a strategy of player u_i and refer to strategies of all other players as a deleted strategy profile $\boldsymbol{s}_{-i}$.

If a player is randomly selecting among her pure strategies (i.e., she associates with her pure strategies a probability distribution and realizes one strategy at a time with the probability assigned to it), we say that she is playing a mixed strategy game. Consequently, a mixed strategy σ_i is a probability distribution over S_i, and $\sigma_i(s_i)$ represents a probability of s_i being played. Similar to a pure strategy game, we denote a mixed strategy profile as a vector $\boldsymbol{\sigma} = \langle \sigma_1, \sigma_2, \ldots, \sigma_m \rangle = \langle \sigma_i, \boldsymbol{\sigma}_{-i} \rangle$, where in the last case we separated a *mixed strategy* of player u_i from a deleted mixed strategy of the remaining players. However, contrary to a deterministic payoff function $u_i(\boldsymbol{s})$ for pure strategy games, the payoff function for player u_i in a mixed strategy game $u_i(\boldsymbol{\sigma})$ expresses her *expected payoff*.

A Nash equilibrium (NE) is a strategy profile in which no individual player

has an incentive to unilaterally change her action, assuming that strategies of all other players remain the same. More precisely, a mixed strategy profile $(\sigma_i^*, \boldsymbol{\sigma}_{-i}^*)$ is NE if

$$\forall_{u_i \in I},\ \forall_{s_i \in S_i} \quad u_i(\sigma_i^*, \boldsymbol{\sigma}_{-i}^*) \geq u_i(s_i, \boldsymbol{\sigma}_{-i}^*)\ . \tag{1}$$

NE is an important condition for a self-enforcing protocol which lets us predict outcomes in a game with rational players. Any normal-form game where mixed strategies are allowed has at least one NE, whereas some pure strategy games may have either unique, multiple, or no NE, which reduces its usefulness for many practical implementations; we concentrate on the possible group of repeatedly played games that always yields an equilibrium solution. D. Monderer and L. S. Shapley [47] identify a class of normal-form games, referred to as *potential games*, for which at least one NE exists. A game Γ is a potential game if there exists a potential function Φ, defined over all elements of S in Γ, such that

$$\forall_{\boldsymbol{s} \in S},\ \forall_{s_i, s_i'} \ \ (u_i(s_i', \boldsymbol{s}_{-i}) - u_i(s_i, \boldsymbol{s}_{-i})) > 0$$

$$\text{iff} \ \ (\Phi(s_i', \boldsymbol{s}_{-i}) - \Phi(s_i, \boldsymbol{s}_{-i})) > 0\ . \tag{2}$$

Eq. (2) defines an ordinal potential game, where the direction of the potential function and any individual payoff function changes have to be the same. A potential game becomes an exact potential game when

$$\forall_{\boldsymbol{s} \in S},\ \forall_{s_i, s_i'} \ \ u_i(s_i', \boldsymbol{s}_{-i}) - u_i(s_i, \boldsymbol{s}_{-i}) \ = \ (\Phi(s_i', \boldsymbol{s}_{-i}) - \Phi(s_i, \boldsymbol{s}_{-i}))\ . \tag{3}$$

Because function Φ in Eqs. (2, 3) reflects possible changes of each individual

payoff function, it becomes a convenient tool to find NE by finding its local optima.

3.1.2 Extensive-form Games

Game trees usually represent extensive-form games with vertices corresponding to the decision points and where outgoing edges represent the possible decisions available at respective points. A root node is the first decision point of the game and leaves are payoff vectors that represent payoffs for all game participants. Every sequence of edges from the root node through all tree levels up to a terminal node (a leaf) represents the *game path*. For convenience, nodes at the same level (depth) are representations of decision points for the same player and each node is labeled by a player that makes decision at it.

In order to better visualize extensive-form games and tyeir representations, we build a typical extensive-form game example based on the *merger game* analyzed by [11]. In merger game, there are 6 firms numbered $1 \ldots 6$, in the order from the largest (firm 1), to the smallest (firm 6). Firm 1 plans to buy firm 6, firm 2 intends to buy firm 5, and firm 3 wants to merger with firm 4. Decisions are made by firm 1, firm 2, and then firm 3, in this order. Figure 1 shows a merger game's tree.

In Fig. 1, the root node represents the decision point for firm 1, which is the first one to choose its strategy of buying firm 6 or not. Following firm 1's decision, firm 2 picks its strategy and, finally, firm 3 chooses its action. The leaves represent vectors with relative payoffs for firm 1, 2, and 3, in this order.

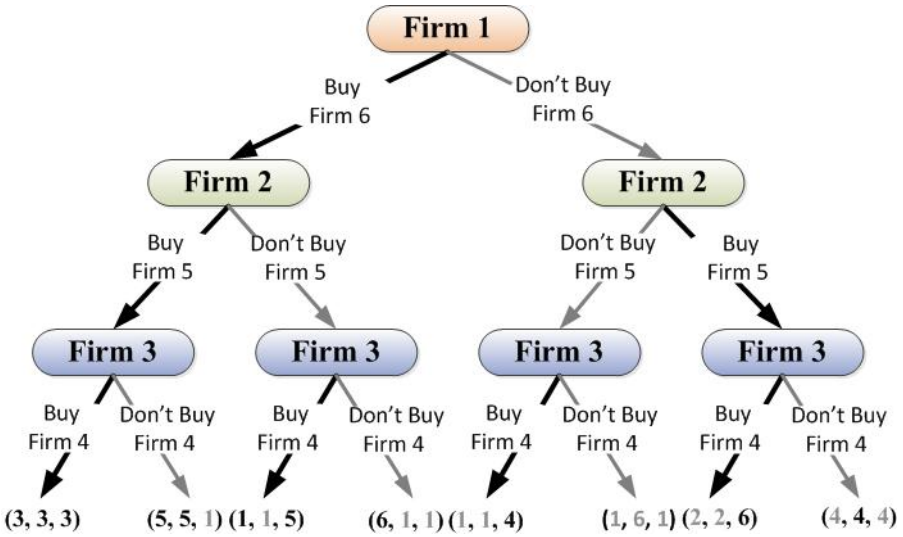


Figure 1: Game tree representation for the merger game.

Payoffs for firms 4–6 are ignored as they are passive entities in this game. The decision sequence $\{buy; don't\ buy; buy\}$ represents the game path where firm 1 chooses to buy a matching smaller firm and receives a payoff of 1. Similarly, firm 2's payoff is 1 and firm 3's payoff is 5.

A *backward induction* is a procedure that can be easily applied to any extensive form game of perfect information that can be used to find a solution called *rollback equilibrium* by the following method:

- choose the strategy giving the maximum payoff for the last level decision making nodes and remove other branches from the tree;
- move to the next-to-last level of the tree and repeat the previous procedure for the decision maker at this level;
- continue the previous step until the root node is reached;

- the remaining edges of the graph indicates an unique *equilibrium path* of the game, when followed from the root to the leaf.

The solution for the backward induction procedure applied to a merger game can be seen in Fig. 1 where the dark edges represent optimal decisions for the players at their decision nodes and, consequently, are remaining links of the tree. We can see that there is the unique game path that follows these dark links (i.e., $\{buy; buy; buy\}$) with associated payoff vector (3, 3, 3), which is the rollback equilibrium for this game.

Kuhn's theorem states that every game of perfect information with a finite number of nodes has a solution to the backward induction procedure and if the payoff to all players at all terminals is distinct, there is only one unique equilibrium. The rollback equilibrium is necessarily NE, however, there may exist NE, which cannot be realized by backward induction. For example, in Fig. 1 the path $\{don't\ buy; don't\ buy; buy\}$ is NE for this game (i.e., no firm can unilaterally divert from it to increase its payoff) but would never be realized as firm 1 always prefers $\{buy; don't\ buy; buy\}$ over $\{don't\ buy; don't\ buy; buy\}$ path. Consequently, a refinement of NE called *subgame perfection* was introduced to strengthen the equilibrium notion for games. A strategy profile is subgame perfect if for every subgame of the original game it is also the NE of that subgame, where subgame is defined as a piece of an extensive-form game starting with a single decision nude from the original game and containing all subsequent nods (e.g., in Fig. 1, any of the firm 2's or firm 3's decision node can originate a subgame). A rollback equilibrium is by definition subgame perfect equilibrium as it

is achieved by finding equilibria for all subgames of the game.

3.1.3 Sequential Games

Interactions in which a simple base game is repeated more than once are classified as *repeated games*. Unlike a game played once, a repeated game allows for a strategy to be contingent on the past moves, thus allowing for reputation effects and retribution. In repeated games, we will refer to a strategy in one round of the base game as an *action* and *strategy* will exclusively refer to a set of decisions made over all repetitions.

In a finitely repeated game where a base game has the unique NE, an optimal strategy is composed of the repetition of this NE in each stage. Since there is no incentive for any player to deviate from the equilibrium in any round of the game, there is no other favorable strategy available. Consequently, the subgame perfect equilibrium exists and is a sequence of the base game's NE. In a finitely repeated game with more than one NE for a base game, there exist many subgame perfect equilibria. Some of them may involve playing long strategies that are collectively more profitable for both players than any of the base game equilibrium. Others, may oscillate between equilibria more profitable for different players and could be triggered by the history of opponents' strategies.

In infinitely repeated games, which are often called supergames, *trigger strategies* like the *grim trigger* or *tit-for-tat* strategies, can be used to encourage cooperation. In grim trigger strategy, a player cooperates in each round until a single defection by her opponent, after which the player defects and plays a

punishment strategy hereafter. In a tit-for-tat strategy, a player responds in one period with the same action her opponent used in the previous round. Infinitely repeated games are modeled with a help of a *discount factor*, which discounts the value of the future reward. The discount factor may also be interpreted as the probability of the game ending at each stage or as a degradation of the payoff value over time. The discount factor allows for analyzing infinitely repeated games in the finitely bounded domains (i.e., as games with finite cumulative payoff and assured eventual termination). Also, the discount factor is an important tool to sustain an equilibrium strategy profile that is not achievable in a base game. A group of *folk theorems* suggest that any outcome for repeated games can be sustained as an equilibrium strategy.

3.2 Applications of GT to Dynamic Spectrum Sharing

After the development and formalization of software-defined radio (SDR) [46], which allowed for far greater flexibility in the management of transmission and reception parameters by shifting the control of transmitters and receivers from the rigid hardware domain to the software implemented domain, the next goal for more efficient utilization of the available frequency spectrum was the development of cognitive radios (CRs). CR refers to the paradigm describing an autonomous agent (the radio) that is able to sense the environment and—from this information, previous experiences, and predefined goals—make an intelligent and autonomous adjustment of its transmission and reception parameters. The

CR exploits SDR as well as intelligent antennas, which are built on the concept of spatial beam-forming, to best utilize existing recourses and meet specified objectives. Because CRs are exploring possibilities of point-to-point communication and are operating only on *physical* and MAC layers of the open systems interconnection model (OSI model), their impacts are limited to the respective localized environments.

In order to address the overall network's needs, cognitive controlling of resources at the local level is not sufficient. Consequently, the notion of a cognitive network (CN) was developed, which extends the scope of the decision processes to cover the network's overall performance and takes into account both network and end-users' objectives.

The knowledge of individual an autonomous agent in an ad hoc wireless network is often limited due to the lack of a centralized authority and/or the costs of maintaining a complete information profile of a constantly changing environment. Likewise, in decentralized networks, the actions of an individual agent may also be driven by selfish or even malicious motivations. These aspects create a convenient platform for GT applications to provide an intelligent decision-making process for CRs and CNs with regards to DSS.

In the network modeled by a group of primary users (PUs), that own frequencies in the area of interest (e.g., through a static pre-assignment or monopolized market model), and secondary users (SUs), who are interested in acquiring, on a temporary or permanent basis, unused frequency bands or time-slots from PUs, dynamic spectrum access facilitates higher resource utilization than traditional

static or centralized techniques. This is possible because available spectrum tends to be temporally and spatially used in an uneven manner and static or centralized techniques suffer from long response times that may prohibit them from taking full advantage of network fluctuations. Likewise, effective DSS techniques require spectrum reassignment decisions to be performed instantaneously based on the local information and without involvement of a central authority. Each node participating in a spectrum acquisition/selling process wants to increase its own profit without regard to others objectives. The characteristics would seem to suggest that GT may be usefully applied to these problems.

Z. Ji and K. J. R. Liu [27] identify three main areas of GT applications to DSS: *(a)* the game theoretical modeling of behavior of users, *(b)* GT algorithm design approaches, and *(c)* optimalization analysis using GT. To model a behavior of nodes, a group of generic non-cooperative distributed dynamic spectrum sharing games (DSSGs) is also proposed by Z. Ji and K. J. R. Liu [27]. In a DSSG, the set of players is comprised of all network users, which has a specific strategy space reflecting the user's objectives and thus may be significantly different for a PU and a SU.

In DSS games, the strategy space could include such factors as transmission power, carrier frequency, modulation strategy, localization, and leasing options for unused spectrum frequencies. The concepts of Nash equilibria, Pareto optimality (PO), and Nash bargaining solutions (NBSs) together with the price of anarchy (PoA)—the ratio of the social optimum solution to the the worst NE result—are often used as performance metrics to evaluate the quality of results.

3.2.1 Auction-based Approaches to DSS

The exchange of unused spectrum frequencies between PU and SU can be simulated by an auction game (AG). In AGs, the payoffs for all players are determined by the level of interest among SUs to acquire a resource offered by a PU and by PU's willingness to share it. If there are multiple PUs offering resources and many SUs willing to acquire them, the double AG is used to simulate competition among *sellers* and among *bidders*. In the double AGs, the competitive equilibrium (CE) is defined as the balance of number of sellers willing to sell for a price that attracts the same number of buyers willing to buy. The CE concept is closely related to the supply demand functions represented in Fig. 2 and which detailed analysys could be found in [27].

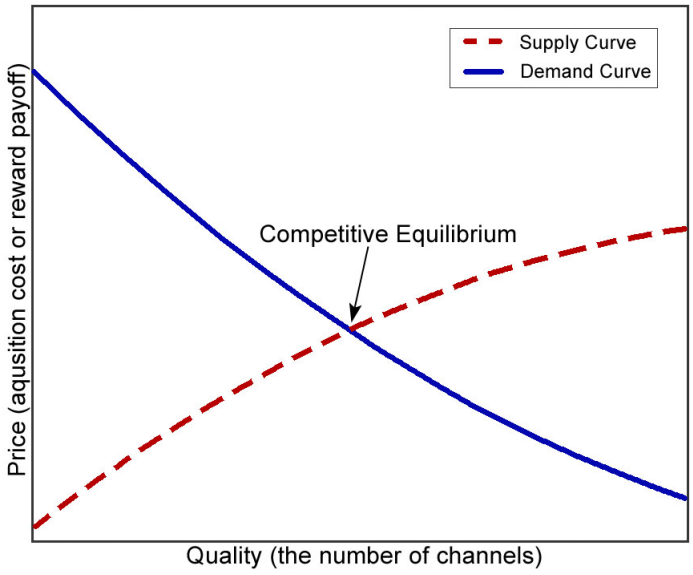


Figure 2: Graph of supply and demand functions (reprinted from [27]).

An auction-based spectrum sharing approach is proposed by J. Huang et al. [23, 24]. The authors describe a power allocation system in which SUs can purchase the amount of power they wish to use from PU through an intermediary agent provided by PUs (a *manager*). They investigate a Vickrey-Clarke-Groves auction (VCG) model for sealed-bid resource allocation. In the Vickrey auction model, bidders submit closed proposals to the manager and the highest bid wins but pays the second highest bid price proposed in the auction. In Vickrey auction, an optimal strategy for each bidder is to offer the truthfully evaluated price for the good. The VCG extends Vickrey auction by allowing for multiple items to be auctioned at the same time. In VCG, each bidder's offer contains ranked prices over the available resources and the auctioneer distributes them according to submitted bids and own objectives. However, because of interference constraints and a computationally expensive procedure, the VCG does not always yield the best solution in the DSS games and thus J. Huang et al. also propose two different *divisible good* auction mechanisms. The first auction is based on a signal to interference plus noise ratio (SINR), where a payoff matrix for each SU is composed according to the received SINR. In the second auction mechanism, a payoff matrix for each user is composed according to the received power. The authors show that their auction mechanisms produce close to optimal solution with much less computation required than the VCG based approach.

Multi-stage dynamic auction-based games for dynamic spectrum allocation are investigated by Z. Ji and K. J. R. Liu [26, 27, 28]. They propose a double AG for rational selfish players, with the base station as a PU and unlicensed mobile

users as SUs, which is resilient against coalition formation. Collusion among SUs, as presented in reprinted from [28] Fig. 3, may restrain competition by creating incentives for collusion group members to *cooperate* and, consequently, results in a disadvantageous solution for the PU.

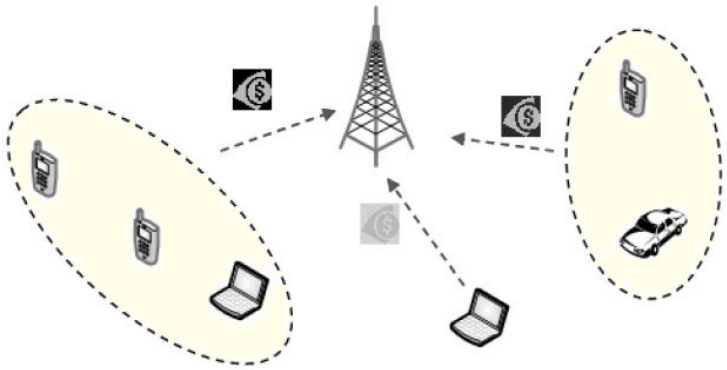


Figure 3: An example of collusion groups (reprinted from [28]).

The AG introduced by Z. Ji and K. J. R. Liu can be viewed as an infinitely repeated game with a discount factor, which may prevent network starvation by introducing incentives for players to make their decisions sooner rather than later and simplifies repeated game analysis by guaranteeing bounded cumulative payoffs for each player. A repeated game, besides being a convenient and an accurate modeling tool for spectrum occupation variations in wireless networks, also allows for player reputation and belief-assisted responses. The concept of belief-assisted pricing for DSS is presented in [26], where each user maintains her own belief matrix based on the game history and basic heuristics to predict possible future strategies of other players. Assisting a bidding process with be-

lief matrices compensates for lack of information about the preferences of other players and results in improved performance.

Further elaboration on learning processes in repeated games is presented by B. Wang et al. [70], where the authors provide simulation results for their self-learning framework showing that achieved DSS outcomes are close to those obtained by centrally controlled networks.

Auction-based spectrum sharing games presented in this section require a central and trustworthy entity (e.g., a manager) that controls auction's mechanisms, bandwidth, and power assignments. Also, global knowledge is often assumed by all or subgroup of agents, which greatly limits its applicability to MANETs.

3.2.2 Bargaining Games and DSS

Nash bargaining games (NBGs) , which try to imitate bargaining interaction among players, are one of the proposed GT solutions to various DSS challenges. In NBG, players demand some or or all existing commodities (e.g., available frequencies or time slots in FDMA and TDMA channel access methods, respectively). Within this framework, NBS is described as an outcome in which the division of the total commodities among players forms a partition of these commodities.

L. Cao and H. Zheng [6] propose a local bargaining approach to achieve a conflict-free spectrum assignment in dynamically distributed systems. In the proposed scheme, when the network's activities result in underutilized or con-

flicting exploitation of resources (e.g., due to node movement or users joining or leaving the network) a local bargaining group is formed to resolve the introduced inefficiency. The bargaining group can be formed between two adjacent nodes, to perform one-to-one bargaining, or among many nodes, where a one-buyer-multiple-seller strategy is used. At the same time, many bargaining groups can perform spectrum allocation reassignment but they have to be isolated from each other and no single node can participate in more than one group, as can be seen in Fig. 4 (reprinted from [6]).

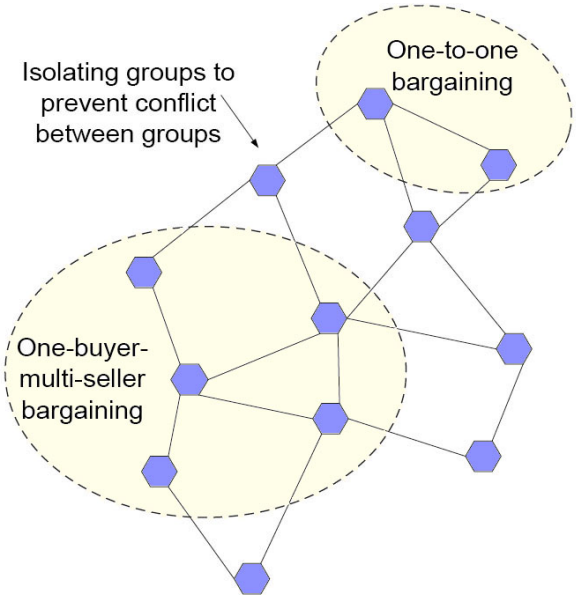


Figure 4: An example of bargaining groups (reprinted from [6]).

Note that, in this approach, all users must be willing to take part in the auction and eventually sacrifice their own objectives if it results in the better collective throughput for the group. Besides being altruistic, players must form bargain-

ing groups according to some predefined criteria and objectives and be able to evaluate the group's performance, which are not trivial problems on their own. The initial assumptions of L. Cao and H. Zheng include a well-formed network at the beginning of the game.

A Rubinstein-Stahl bargaining game (RSBG) can be viewed as a two-players extensive-form game where players move in turns proposing a division of some good, whose total value sums to unity. At each step, one of the players proposes a division and the other may reject or accept it. If proposed splitting is accepted, the game ends with this division. Otherwise, the second player makes an offer and the first one has the options of accepting or rejecting it. In the case of rejecting the second player's offer by the first one, the cycle is repeated. M. Pan et al. [52] explore RSBG to assist in the trading of available spectrum channels by proposing the algorithm outlined in Fig. 5 (reprinted from [52]).

The subject of the trading is not an available channel, however, but the proportion of the profit resulting from being able to utilize it, as the total profit for each channel equals 1. Consequently, PU's and SU's earnings sum up to 1 and represents a pure competition constant-sum game.

As a consequence of the fact that bargaining occurs only between two entities (i.e., PU and SU) in the proposed model, the involved players' profits from it may be suboptimal. This problem could be alleviated by allowing the PU to engage into multiple simultaneous RSBGs over the same channel and/or the SU to be able to try acquisition of multiple channels that satisfies its criteria. However, these approaches require a large amount of synchronization, thus losing the advantages

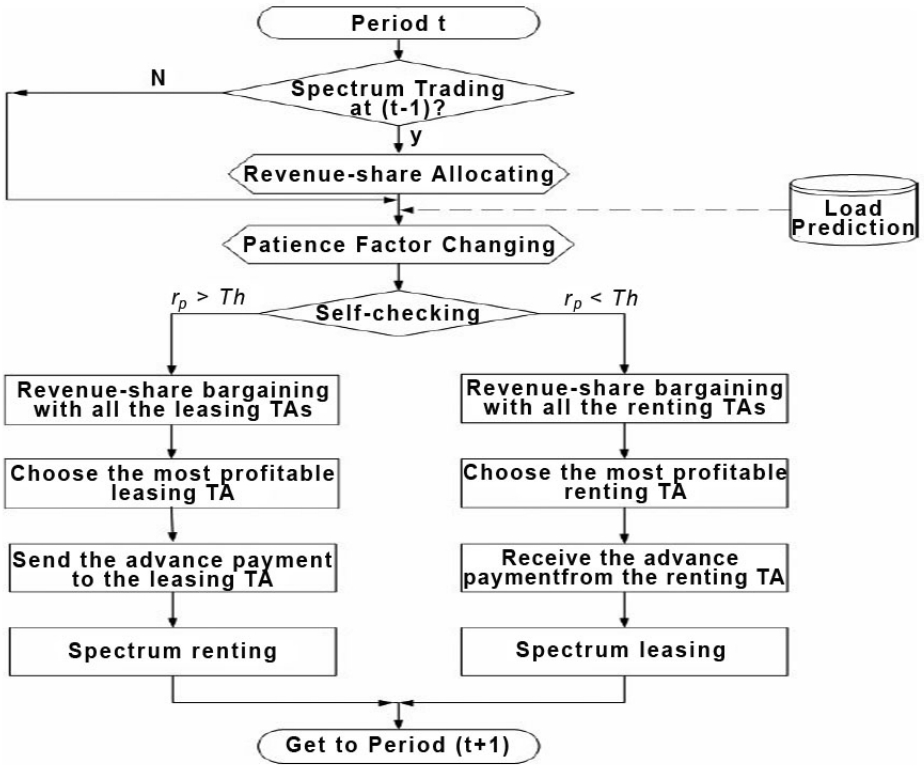


Figure 5: RSBG based algorithm presented by M. Pan et al. (reprinted from [52]).

of localization in the proposed scheme.

In [20], M. Halldorsson et al. analyze the channel assignment problem as a bargaining game between service providers and access points (APs). In their game, PoA is specified as a ratio between the total APs coverage in the worst NE case and what the socially optimal coverage would be. Based on this definition, the authors provide bounds on the PoA depending on the assumed underlying network and the type of bargaining allowed between service providers and APs. Considered assumptions are uniform/inconsistent spreading of users and homo-

geneous/heterogeneous antennas' propagation powers. The two bargaining types are 2-buyer-1-seller and 1-buyer-multiple-seller. The authors prove that for networks with uniformly spread nodes or homogeneous signal power of antennas, the POA is bounded by a constant for both types of bargainings.

3.3 Applications of Game Theory to Topology Control

A node distribution can be modeled by deployment into predefined locations or a random scattering throughout an area of interest. Furthermore, ongoing network dynamics can be simulated by random motion models, random nodes expiration and redeployment, or predefined motion patterns. Improvement of area coverage can also be achieved by superfluous deployment; however this approach may be viable only with inexpensive agents and may result in a prohibitive increase of network maintenance overhead.

Recent developments in mobile sensor network technologies as well as adaptations of ad hoc communication techniques to mobile autonomous agents allow for control of the layout of a MANET through the autonomous relocation of individual agents in a dynamic environment. Achieving better spatial placement may lead to an improvement in area coverage with reduced sensing overshadows and limited blind spots, as well as to the reduction of power consumption, better spectrum utilization, and the simplification of routing procedures. However, the lack of a central omniscient authority controlling all the nodes and limited coordination, typical for medium to large ad hoc networks, may result in

a degradation of overall network performance due to the selfishly determined relocations by individual agents. Additional concerns are related to the unknown geographical territory and the hostile environment in which the network may have to operate. As a result, we observe an increasing interest in GT-based solutions to TC related problems.

Topology control in MANETs can be viewed from two distinct perspectives. In one of them, the goal is to control the communication links among nodes in the model where agents are immobile or their mobility are controlled by external factors. Section 3.3.1 presents some GT approaches to this group of problems.

Another TC method achieves the network's desired objectives by controlling movement and location of nodes. Managing the movement of nodes, in network models where each node is capable of changing its own spatial location, could be achieved by implementing potential field concepts [22, 51, 41, 74]. For example, W. Xi et al. [74] propose a solution for control and coordination of autonomous swarms relying on Markov random fields and Gibbs sampling. Other proposed solutions are based on Lloyd algorithm or nearest neighbor rules [8, 25], respectively. Our autonomous node-spreading methods presented in Chapters 4, 5 and 6 fall into this category of TC approaches.

3.3.1 TC by Propagation Power Adjustment

The simplest methods to control a network topology are based on the assumption that the position of the nodes is known in advance, which is an overly stringent requirement for most applications. These approaches try to achieve a desired

global topology by preassigning appropriate transmission characteristics to each node in advance. Other approaches consider node locations and movement patterns as random factors (e.g., random waypoint, random direction, or Brownian motion based models) or simulate random redeployment and expiration of nodes. One common TC application is directed towards minimization of transmission power for each mobile node while assuring network connectivity with a high probability (see [58] for the presentation of various non-game theoretical approaches applied in these situations). This section presents two GT based TC solutions that adjust the power level of individual transmission antennas to control network connectivity. Both approaches presented here may operate on top of a dynamically changing network and are able to converge to a stable state as long as the changes in underlying network structure are slower than the necessary convergence time of presented solutions.

Eidenbenz et al. [12] explore the concept of NE for normal-form games in MANETs to control the topology by choosing an appropriate power level for the individual nodes. The authors present two kind of *connectivity games*. In *strong connectivity games* each network node is required to be connected (not necessary directly) to all other network nodes. In *strongly k-connected games* each node can reach every other node via k internally disjoint paths. The games proceed in synchronized steps among the players, where only one player is active at a time. A player reduces her radio propagation power to the level that preserves desired network characteristics.

The power level as a decisive element of underlined topology is also ex-

ploited by R. S. Komali et al. [29, 30] with potential games for power-efficient connected networks. In [29], the authors present *Power reduction potential games* that maintain a desired topology with the minimal power consumption and transmission power of individual devices. It is assumed in their work that a current state of topology is known for each user at each step of the computation, all links to be considered useful must be bidirectional, and all intermediate nodes will cooperate in forwarding messages once the topology had been established by selfishly motivated decisions. They define an energy efficiency objective as a point where no node can further reduce its transmission power without disconnecting the network. Furthermore, the graph $G(V, E)$ represents an *efficient* underlying graph where V is a set of all nodes and the set of edges E reflects feasible links between nodes propagating with the highest possible powers.

The transmission level of node i for all $i \in N$ is represented by p_i, where N is a set of nodes (players), and p_{ij} represents the power required to support an edge (i, j) in G. A link state variable l_{ij} is defined as

$$l_{ij}(p_i) = \begin{cases} 1 & \text{iff } p_i \geq p_{ij} \ ; \\ 0 & \text{otherwise} \end{cases}$$

and the set of i neighbors is

$$N_i(p_i) = \{j \mid l_{ij}(p_i) \cdot l_{ji}(p_j) = 1\} \ .$$

The authors define $g(p)$ as a collection of all nodes' neighbors, where p is a vector containing all users' power levels, and the same payoff function for player

$i \in N$ is defined as

$$u_i(p) = \varphi_i(g(p)) - p_i \ ,$$

which represents the benefit that node i derives from being a part of the topology $G\,(g(p))$.

The algorithm presented by R. S. Komali et al. [29] proceeds in steps. In the first step, each node i propagates a *neighbor discovery* message with the maximum power and creates a relative set N_i of own neighbors from the received responses. Then, a topology of the entire network is stored by each node via connecting all N_i into a multihop communication network. The second step is a repeated cycle in which each node tries to maximize its payoff profit. It is assumed that after each modification, the updated topology is known to all participants. Fig. 6 presents a topology update cycle proposed by [29].

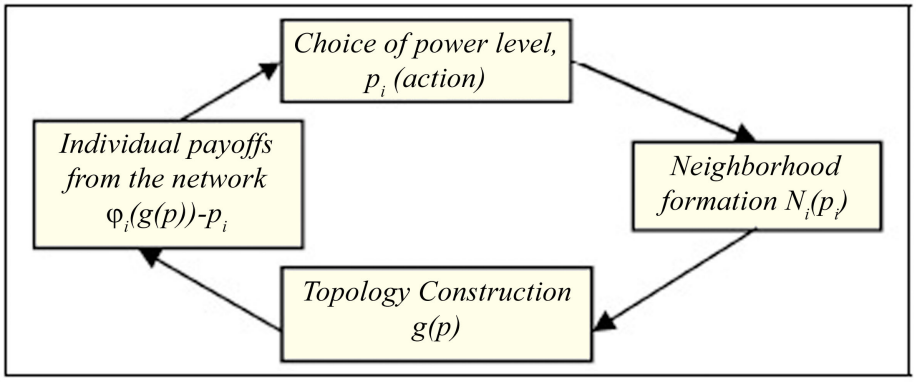


Figure 6: Topology control cycle (reprinted from [29]).

The authors define a value of $M \geq \max_{i \in N}\{p_{i,max}\}$, which reflects how nodes rate a network connectivity, and propose two topology control games that

are both defined with respect to M. In the first game $\Gamma_1 < N, A, u >$, the non-decreasing function $f_i(p)$ is defined as a number of nodes that can be reached by a node i and a payoff function defined as

$$u_i(p) = M f_i(p) - p_i \, .$$

The ordinal potential function for Γ_1 is defined as

$$\Phi(p) = M \sum_{i \in N} f_i(p) - \sum_{i \in N} p_i \, ,$$

which makes Γ_1 an ordinal potential game.

The game $\Gamma_2 < N, p, u >$ is an exact potential game and has a payoff function

$$u_i(p) = \begin{cases} M - p_i & \text{if network is connected ;} \\ -p_i & \text{otherwise} \end{cases}$$

and corresponding exact payoff function

$$\Phi(p) = \begin{cases} M - \sum_{i \in N} p_i & \text{if network is connected ;} \\ - \sum_{i \in N} p_i & \text{otherwise .} \end{cases}$$

Γ_2 is an exact potential game that has a power level equilibrium that is equal to the (local) maximum of the corresponding potential functions.

Both solutions for NT presented in this section assume a global knowledge of the current state of the entire network for each user at each step of the computation and full synchronization among all nodes, which may result in their impracticality in managing many realistic networks.

3.4 Other Applications of Game Theory to Wireless Ad Hoc Networks

Wireless ad hoc network challenges that cannot be neatly assigned into DSS or TC groups and for which effective GT solutions have been presented are discussed here. For clarity of presentation, this section is organized according to the game types used instead of the solution concepts they are addressing. Section 3.4.1 presents Stackelberg pricing games and their use for the network control as well as their possible applications to DSS and TC. Games of incomplete information model players' limited knowledge about the circumstances under which they are played, and are suitable to mimic many of the realistic ad hoc network scenarios. Section 3.4.2 presents this type of game and their use in alleviating network access and routing problems.

3.4.1 Stackelberg Games

The Stackelberg game (SG) is defined by a set of players, containing the *the leader* that commits to her strategy first and remaining players that select their strategies sequentially after the leader, called *followers*. In order for the Stackelberg strategy to be effective, the leader's move must be observable and common knowledge for all players (i.e., every player knows the leader's move and knows that everyone knows it). Furthermore, rational followers act selfishly and try to maximize their profits considering the action chosen by the leader. In a two-player SG, there is the leader and one follower and, consequently, the game has

only two stages. Two-player SGs are used to model simple interaction between hierarchical entities with the one having the *first move advantage*. As a complete and perfect information SG with rational players has a subgame perfect NE that could be found by the process of backward induction (often referred in the literature as the Stackelberg solution or strategy), it becomes a convenient tool to model and analyze many real and theoretical events.

In [61], M. Simaan and J. B. Cruz analyze some of the important characteristics of the Stackelberg strategy for broad class of games and derive necessary and sufficient conditions under which its existence is guarantied. In [9], the same authors extend their concepts from the publication [61] to incorporate an idea of *ordinal games*, where players are able to rank-order their decisions according to the own preferences against opponents' choices and use their own rankings, instead of payoff-based evaluation, to determine their optimal strategies. Both of these publications present a good overview of SGs and provide solid bases for potential applications of SGs to various areas of wireless ad hoc networks.

The Stackelberg pricing problem (SPP) has two levels of decision making: the tariff set by an operator (a leader) and the selection of the cheapest alternative by consumers (followers). In SPPs for networks, an operator sets up the prices for a subset of the network's edges and the consumers determine the cheapest path between the two nodes they want to connect. The objective for the operator is to maximize revenue from the utilization of the tariff edges while the consumers aim to pick the least expensive and shortest paths which meet their needs. It is assumed that if a follower has more than one path to choose with the same cost,

he chooses the one that offers the greatest payoff for an operator.

Since SPP is composed of two separate subproblems (the pricing problem and the shortest path problem), it is often represented by a bi-linear (bi-level) model. At the upper level, the leader attempts to maximize his revenue and at the lower level, followers try to minimize the cost of routing their packages through the network.

An application of the Stackelberg pricing model to network control is presented by S. von Hoesel [69]. To assure that the problem is bounded, the author assumes an upper bound on each link price that a user is willing to pay. The network is represented by a directed graph $G = (V, E)$ where the set of edges E is partitioned into E_f, a set of edges with fixed prices, and E_t, a set of tariff edges, where the leader controls prices. The source and destination of the commodity k is described as a pair (s_k, t_k) and referred as a path p_k. The cost of routing a unit of demand on path p_i is defined as a sum of prices for all fixed price and tariff edges that comprise to this path. To clarify this concept, we consider the example from [69] of a 1-commodity network with two tariff edges as shown in Fig. 7 and the corresponding payoff table (Table 1) (reprinted from [69]).

The shortest path graph (SPG) of graph G is the graph $G^*(V^*, \{E_t^*, E_f^*\})$ with all the corresponding vertices and tariff edges from G (i.e., $V^* = V$ and $E_t^* = E_t$) and E_f^* consisting of all passible paths formed by a collection of fixed price edges replaced by links with cumulative prices of underlining paths. In the case of multiple followers, a SPG is constructed for each of them. It can be noticed that the relative costs of corresponding p_i's in G and G^* are the same.

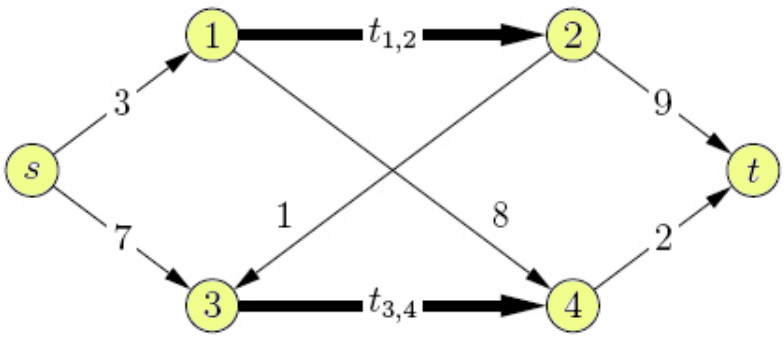


Figure 7: 1-commodity network with two tariff arcs and its payoff table, both reprinted from [69].

Table 1: Payoff table for 1-commodity network model depicted in Fig. 7 (reprinted from [69]).

Path	Length	Profit
$s\,1\,2\,3\,4\,t$	$6 + t_{1,2} + t_{3,4}$	$t_{1,2} + t_{3,4}$
$s\,1\,2\,t$	$12 + t_{1,2}$	$t_{1,2}$
$s\,3\,4\,t$	$9 + t_{3,4}$	$t_{3,4}$
$s\,1\,4\,t$	13	0

Furthermore, the SPG's set of fixed price edges with higher than the minimal fixed price edge can be eliminated as they will not be considered for any set of tariffs. Also, tariff edges with the cumulative cost of reaching some point from s and then reaching t from this point that are relatively high (e.g., if this cumulative cost is higher than direct *fixed* path from s to t) can be eliminated.

S. von Hoesel describes the basic branch-and-bound algorithm for pricing problems that will be outlined here for two-player games only. In the first part of the algorithm, the SPG is computed and is *trimmed* to consist of only relevant

paths for the follower. If we define $c_{p_k^u}$ as an upper bound for the highest fixed cost of p_k and $c_{p_k^l}$ as a lower bound for the smallest fixed cost of p_k, then $c_{p_k^u} - c_{p_k^l}$ represents an upper bound on the possible revenue for the leader. Next, a branch for each of the relevant paths of the client is created by selecting it based on an upper bound revenue criterion. Finally the tree is bounded by computing feasible solutions.

The Stackelberg pricing game can be applied to pricing problems which model the assignment of maximal possible revenue to the retailer's set of products while taking into account the consumers' preferences. Multi-player one-round Stackelberg network pricing games in which the leader can set prices for a subset of all edges in the graph and the followers choose the minimum cost solution by optimizing combinatorial minimization problem has been adopted by P. Briest et al. [5]. The authors start by analyzing a single-price algorithm for one follower. The algorithm proceeds by first finding $c_{p_k^l}$, the cheapest (shortest) path (subnetwork) containing only fixed price edges which meet the follower's demands. Then, the single-price algorithm assigns prices to all tariffed edges and determines returned revenue for each assignment. It returns the pricing that maximizes possible revenue. This concept is extended to a multiple-followers scenario by applying the single-price algorithms for all users. The authors analyze performance of their approaches and prove upper and lower bounds as well as tightness of the approximations.

Stackelberg network pricing games could be adopted to implement DSS with a single PU acting as the leader who dynamically adjusts prices of available re-

sources depending on existing demand and with the remaining SUs being followers. In TC settings with one central entity possessing global network topology information, SG can be used to allow the leader to influence optimal configuration among followers. Though, as in the case of AG, the inherent centralization of SGs may limit possible real life applications of outlined solutions.

3.4.2 Games of Incomplete Information

Games with incomplete information of opponents' characteristics (types) are referred as Bayesian games (BGs). In a BG, we define a strategy for each player as a complete plan of action for all possible combinations of her opponents' types (and strategies she believes they may be). This uncertainty about the opponents is what makes it an incomplete information game. The player assigns a probability to each possible type of opponent and computes her optimal response with respect to the *expected opponents type*. Furthermore, the Bayes-Nash equilibrium (BNE) in BGs is defined as a strategy profile that maximizes a player's expected payoff, given her beliefs about the other players.

S. Adlakha et al. [1] model the competition between wireless nodes with incomplete information as a BG where each wireless device adjusts its power profile to maximize data rate taking into account expected interferences from the other users. The authors define the normal-form BG for two users without knowledge about the channel gain, where both users have the same strategy and where there are only two channels. They demonstrate that a unique symmetric BNE in which both users spread their power equally over the entire available spec-

trum exists. As a result, for the two channel system the equilibrium is achieved when each user propagates information with half of its maximal power on each of the channels. Then, S. Adlakha et al. analyze a more realistic scenario for a normal-form BG in which an user is aware of its own propagation channel gain and the incident channel gain but not of the other users' channel gains. In order to simplify the model, each user is limited to maximum power propagation on one of the channels or can spread the propagation power evenly among all existing channels. It is shown in [1] that BNE exists and is achievable when all users spread their transmission power evenly among all the channels.

S. Adlakha et al. [1] also analyze finite extensive-form BGs. They start with a two stage BG modeled by two players (the PU and the SU) and two channels. The incomplete information game is a result of the primary user knowing her outgoing and incoming channels' gains but the secondary user being aware only of her incoming channel gain. Figure 8 (reprinted from [1]) represents a game tree for extensive game with player 1 as the PU and player 2 as the SU, where *SH* donates a strategy of concentrating power in one channel and *SP* stands for a strategy of equally spreading user's maximum available power among all channels.

The BNE is found by the process of backward induction. The best second player's (P_2's) response to a first player (P_1) playing *SP* in the first stage, would be to play *SP*. If P_1 plays *SH* as her first move, the best response for P_2 depends on the gain of P_1's propagation on the channel and should be *SH* if it is greater than half and *SP* otherwise. Consequently, if the gain on the channel is less than half, P_1 will never choose *SH* in the first stage and if it is greater than half, she

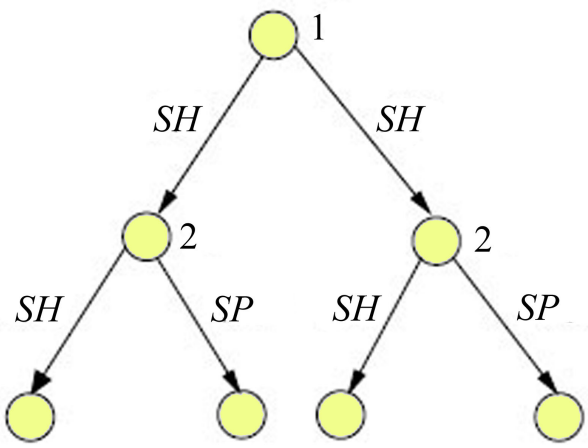


Figure 8: The 2-stage BG (reprinted from [1]).

can compare her payoffs between runs $\{SP, SP\}$ and $\{SH, SH\}$. As a result, a BNE exists in this example and denotes the arrangement where both players play the same strategy.

The game introduced by S. Adlakha et al. is extended by its authors to incorporate the possibility of a P_2 potentially entering the game, depending on incentives and the *cost of power* as an additional constraint on P_2's decision. The new game can be seen in Fig. 9, in which *X* represents the case in which P_2 decides not to enter the system and *N* denotes P_2 entering the game.

Optimal routing in a wireless ad hoc network is one of the most important design and implementation objectives as it affects spectrum utilization and communication delays, thus influencing an overall network performance. Precisely predetermining the traffic load of a wireless network is often impossible due to its unpredictable and dynamic character. Consequently, the path choosing al-

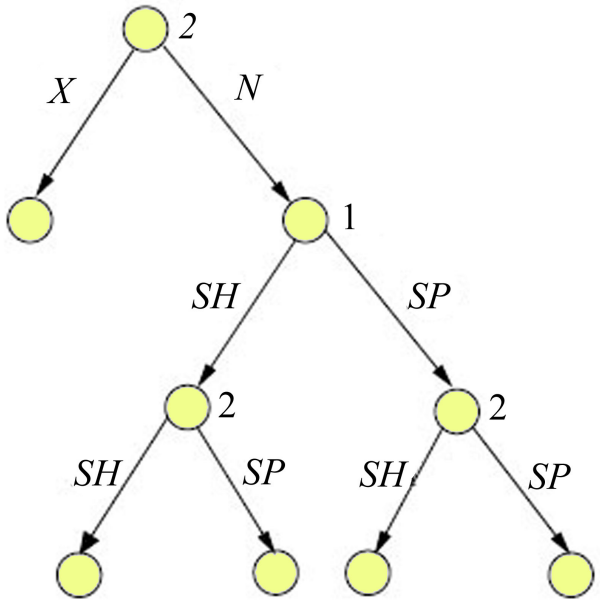


Figure 9: The game tree representing the SU's option to join the game (reprinted from [1]).

gorithm implemented at each node should be dynamic and able to consider the possible actions of other users. Successful applications of BG to assist routing protocols in the network models where nodes can only attain limited knowledge of existing traffic load or a set of active nodes at particular time is presented by [15, 3].

In [15], M. Gairing et al. introduce a BG where selfish users wish to transmit on one of the existing links without knowing the other players' traffic. A player conceives possible opponent types with an appropriate probability assigned to each of them. This model of a game of incomplete information is then transformed to the game of imperfect information by applying the Harsanyi transfor-

mation in which the evaluation of possible scenarios is limited to the comparison of expected payoffs instead of exact payoffs, as in the case of perfect information games. The game proposed by the authors is shown to yield BNE(s) in pure and mixed strategy versions of it in a polynomial time.

I. Ashlagi et al. [3] analyze Byesian routing games where each player can split her traffic among paths in a given network and where the subset of active users at the certain time is unknown. The authors propose a relation between the number of active and potentially active nodes under which *ignorance* is beneficial. They defined an index for the value of ignorance as the difference between the socially optimal cost of routing for k active players and equilibrium in a relative game of incomplete information. Their model indicates that for an index value greater or equal to 0 that ignorance is beneficial.

Bayesian games can correctly predict and enforce an optimal outcome for the game where players do not possess full information about exact opponent types. As such, they are applicable to many wireless ad hoc network problems in which accumulating complete information about network structure and other node objectives is prohibitively expensive or impossible. This characteristics suggest that the role of BG in solving many issues of wireless ad hoc networks will grow in the future.

3.5 Evolutionary Game Theory

EGT originated as an attempt to understand evolutionary processes by means of traditional GT. However, subsequent developments in EGT and broader understanding of its analytical potential provided insights into various non-evolutionary subjects. Some of the EGT contributions to traditional GT are: *(a)* alleviation of the rationality assumption, *(b)* refinement of traditional GT solution concepts, and *(c)* introduction of a fully dynamic game model. Consequently, EGT evolved as a scheme to predict equilibrium solution(s) and to create more realistic models of real-life strategic interactions among agents. Because EGT eases many difficult-to-justify assumptions, which are often necessary conditions for deriving a stable solution by traditional GT approaches, it also become an important tool for designing and evaluating MANETs.

The first formalization of EGT can be traced back to R. C. Lewontin, who, in 1961, suggested that the fitness of a population member is measured by its probability of survival [42]. Subsequent introduction of an evolutionary stable strategy (ESS) by Smith and Price [64] and a formalization by Taylor and Jonker [66] of the replicator dynamics (i.e., replicator dynamics is an explicit model of the process by which the percentage of each individual type in the population changes from generation to generation) led to increased interest in this area.

In EGT, players represent a given population of organisms and the set of strategies for each organism contains all possible *phenotypes* that the player can manifest. Evolutionary game theory focuses on a distribution of strategies in

the population rather than on actions of an individual player. In EGT, changes in a population are understood as an evolution over time resulting from genetic mechanisms favoring one phenotype (strategy) over the other(s). Individuals in EGT are often not modeled explicitly and the fitness of an organism shows how well its type does in a given environment. A very large population size and repeated interactions among randomly drawn organisms are often among initial EGT assumptions. In this framework, the probability that a player encounters the same opponent twice is negligible and each individual encounter can be treated independently from the game history (i.e., each individual match can be analyzed as an independent game).

An ESS is a strategy that cannot be gradually invaded by any other strategy in the population. Let $u(s^*, s')$ denote the payoff for a player playing strategy s^* against an opponent's strategy s'. Then s^* is ESS if either one of the following conditions hold

$$u(s^*, s^*) > u(s', s^*)\,; \tag{4}$$

$$\big(u(s^*, s^*) = u(s', s^*)\big) \wedge \big(u(s^*, s') > u(s', s')\big)\ , \tag{5}$$

where $\wedge$ represents the logical AND operation. The ESS is a NE refinement that does not require an assumption of players' rationality and perfect reasoning ability.

The evolutionary game model where each player has an equal probability of being matched against any of the remaining population members may be inappropriate for analyzing many realistic applications. Nowak and May [50] recognized that organisms often interact only with the population members in their

proximity and proposed a group of spatial games in which members of the population are arranged on a two-dimensional lattice with one player occupying each cell. In the model presented by Nowak and May [50], each player participates in a simple base game with each of its closely located neighbors at every stage of the game, and combines her payoffs from all these matches to evaluate her current and possible future strategies.

Spatial games can be extended to model node movement in MANETs where the decisions of agents are based only on their local information. This approach is presented by us in Chapters 5 and 6.

3.6 Genetic Algorithm Concepts

Genetic algorithms (GAs)are computational methods which emulate evolutionary theory such that the better adapted individuals have greater chances of survival in a given environmental niche and therefore of passing their genetic materials to their offspring. Hence, beneficial characteristics of inhabitants improve from generation to generation and enhance well-being of the entire population.

GAs are used to solve complex problems where the solution spaces are either too large to be searched using brute-force methods, or other approaches are computationally too expensive or cannot provide an adequate approximation of the solution. In contrast, GAs are adaptive search techniques that are intrinsically parallel and can efficiently explore the entire problem space. GAs are commonly applied to a large variety of optimization and search problems, such as schedul-

ing, software engineering, network topology control, and financial system emulators.

Since GAs attempt to mimic evolutionary processes in nature, biological terms have been embraced to designate their elements and processes. In GAs, a *chromosome* represents a possible solution and a considered set of solutions is called a *population*. A *selection* mechanism is used to choose *parent* chromosomes from a given population for a *crossover* (reproduction) operation that yields one or more *offspring*. The chromosomes usually *mutate* (i.e., their genetic representations are altered) with a small probability. The population evolves from one generation to the next through selection, crossover, and mutation mechanisms. In GAs, we use a mathematical *fitness* function to evaluate the quality of a chromosome. We present a general form of GA in Fig. 10 where an initial population is represented by a randomly created group of individuals (e.g., chromosomes or possible solutions).

In basic GA (see Holland [21]), chromosomes are represented by binary strings and are selected for crossover operation based on their proportional representations in the population. Selected chromosomes are then paired to perform a single-point crossover that swaps their respective parts. The binary representations of offspring are then mutated with a small uniform probability. The population evolves for a predefined number of generations or until a satisfactory new solution is found.

The selection, crossover and mutation operators greatly influence the GA performance. The main objective of the selection is to promote better-fitted chro-

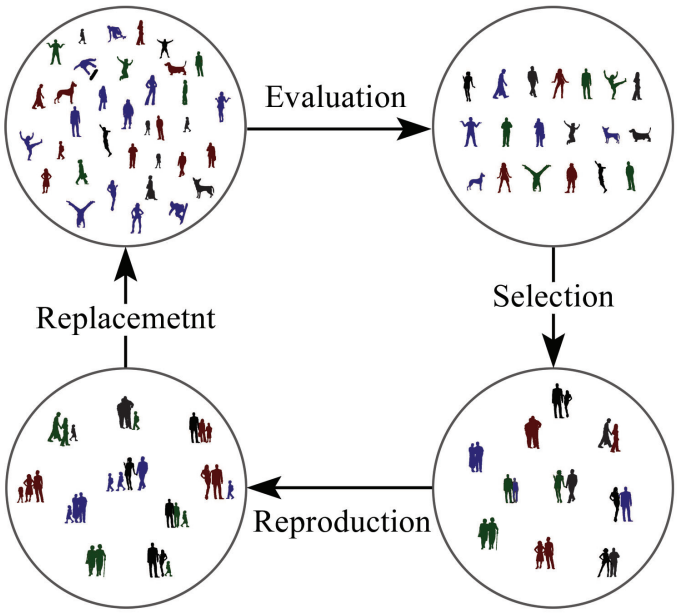


Figure 10: Evaluation, selection, reproduction, and replacement operations of a basic form GA (reprinted from [67]). Depicted in this figure processes represent a single generation of GA.

mosomes for the reproductive processes. Among the selection mechanisms are roulette wheel, elitism, random, and tournament—the effectiveness of which is problem-dependent.

Crossover of selected chromosomes attempts to combine the genetic material of two or more parents to produce one or more offspring. The best probability with which the selected individuals are reproduced can be determined empirically or adjusted dynamically during an evolution of a population. A low crossover probability helps the best-fitted parents to advance to the next generation without any alternations while high crossover probability promotes greater change between generations.

A mutation operation has been proposed to prevent a population from getting stuck within a local optimal solution by introducing chromosomes that may not be reached through selection and crossover operations. A high mutation probability stimulates faster exploration of a search space compared to a lower value. Consequently, a high mutation rate is desired at the beginning of GA iterations, when quick examination of the entire search space is crucial, while a low mutation rate is beneficial towards the end of the evolutionary cycle, since finding the final best solution is a matter of fine-tuning already achieved results. A mutation rate is usually set to a small value (e.g., 0.001 or lower) to ensure that accomplishments of other evolutionary processes are not wiped out at the end of each population cycle but are large enough to allow the population to explore possible new solutions.

3.7 Performance-evaluation Techniques for Gauging Node Self-positioning Methods

In our game-theoretic bio-inspired techniques for self-positioning autonomous mobile agents, the goals of each node are to: *(a)* place itself over an unknown geographical terrain in order to obtain a uniform network distribution, *(b)* reduce a total distance traveled by it before overall network objectives are reached, and *(c)* assume a position that ensures a high coverage of the area by all nodes while preserving the network connectivity. In this section, we introduce quantitative methods for assessing performances of our approaches with respect to these ob-

jectives. The following sections describe each of these schemes in details.

3.7.1 The Uniformity Measures

Equally distanced connected mobile nodes are critical to achieve many network goals. Uniform distribution may simplify high-level network communication and routing operations as well as provide adequate area coverage for environment-sensing purposes. Furthermore, since the lifespan of a MANET under limited-power conditions often depends on the continuous operation of all nodes, it is important to ensure that the nodes deplete their energy resources evenly and no single node stops functioning prematurely. In uniformly distributed networks, where each node has the same sensing area and distance to its neighbors, power utilized by every mobile agent to perform its tasks is similar and, consequently, prolonged uninterrupted operation of a MANET in its full capacity can be accomplished.

In order to gauge a performance of networks with respect to uniform node distribution, we define two measures based on various quantities associated with the Voronoi tessellation of the deployment terrain. Voronoi tessellation associates with each node n_i a Voronoi region V_i such that all locations that are closer to n_i than to any of their neighbors are part of its V_i. We formally define a Voronoi region for each MANET node as

$$V_i = \{\omega \in \Omega : d(n_i, \omega) < d(n_j, \omega), \forall_{n_j \in I \setminus \{n_i\}}\}\,, \tag{6}$$

where Ω represents the set of all positions in the deployment area and $d(i, \omega)$

stands for the Euclidean distance between node n_i (i.e., (x_i, y_i)) and (x_ω, y_ω). Voronoi tessellation of a deployment terrain is a collection of all nodes' Voronoi regions.

Let us define the area of V_i as $A_{\mathrm{v},i}$ and let C_i be the center of mass of region V_i. Figure 11 presents a tessellation of the rectangular constant density terrain depicting basic quantities associated with each Voronoi region. In Fig. 11, the gray rectangle marks $A_{\mathrm{v},i}$ of node n_i's Voronoi cell V_i and $d(n_i, C_i)$ denotes the distance to the center of mass of generated by n_i region. For clarity of presentation, Fig. 11 does not depict any values associated with the other three nodes and their symbols are given to visualize the way V_i was generated.

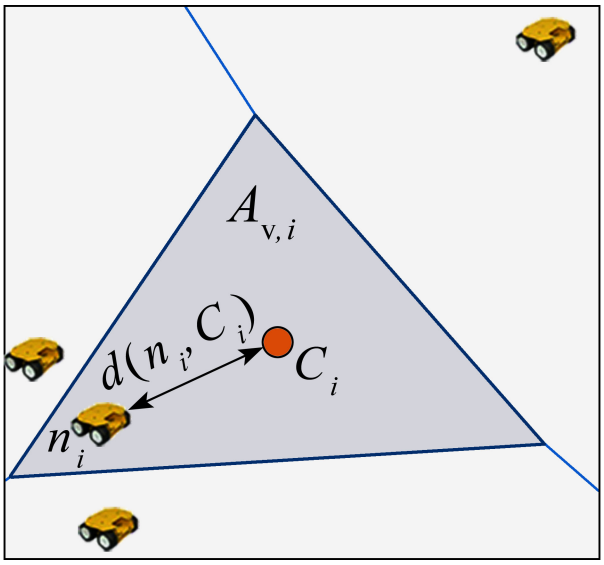


Figure 11: The Voronoi tessellation of a rectangular area. The Voronoi region generated by node n_i (i.e., V_i represented by the dark gray triangle) has area $A_{\mathrm{v},i}$ and center of mass C_i, which is marked by the small circle. The distance between the current location of n_i and corresponding C_i is represented by $d(n_i, C_i)$.

We propose two methods for measuring uniform distribution of MANET nodes over the deployment terrain. One of them, referred as $\mathscr{U}_A$, exploits differences in areas of Voronoi regions generated by the nodes. We define $\mathscr{U}_A$ as

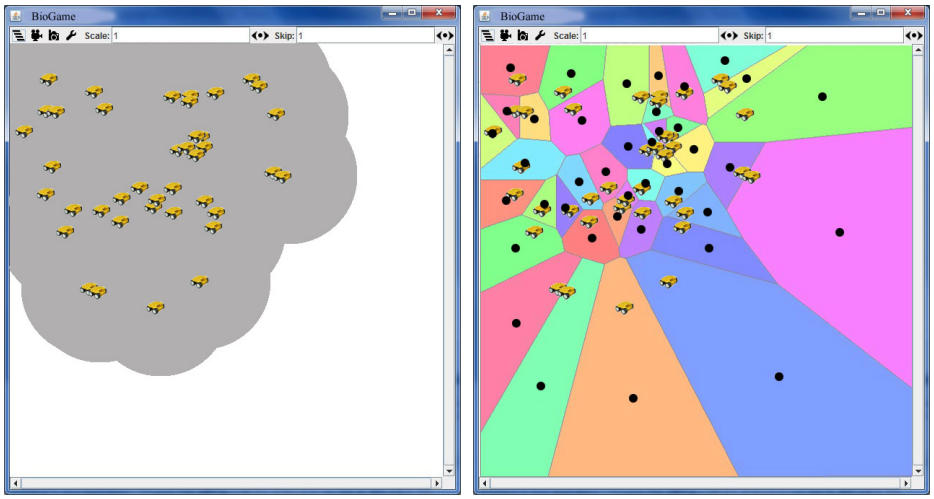
$$\mathscr{U}_A = \frac{1}{\bar{A}_{\mathrm{v}}} \sqrt{\frac{1}{|I|} \sum_{n_i \in I} (A_{\mathrm{v},i} - \bar{A}_{\mathrm{v}})^2} \,, \tag{7}$$

where $\bar{A}_{\mathrm{v}}$ is the arithmetic mean of $A_{\mathrm{v},i}$ for all $n_i \in I$ and $|I|$ denotes a total number of nodes in the network.

In our node self-positioning approaches, if mobile agents are perfectly distributed, areas of Voronoi cell for each node located in the interior of the deployment terrain is the same and slight variations in Voronoi regions can be found only near the boundaries, thus a tessellation of the area that closely resembles collection of congruent regular hexagons reflects a superior node distribution. As a result, measure $\mathscr{U}_A$ approaches zero as autonomous mobile agents improve their locations towards the the stable uniform network distribution.

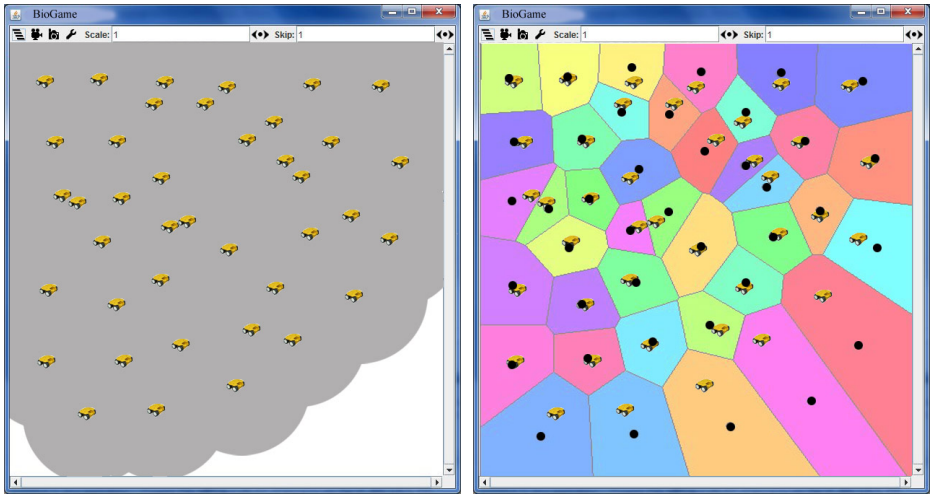
Figures 12–14 show three sample node distributions achieved by our BioGame (Chapter 6) and associated with them Voronoi tessellations. Center of mass of each Voronoi region is marked by a black dot. Measures $\mathscr{U}_A$ for topologies depicted in Figs. 12–14 are 1.6, 0.6, and 0.2, respectively, which is consistent with improvement achieved by the network at these steps.

Our second quality measure for network uniformity is based on the distance between location of each node and the center of mass of its Voronoi region. In a given topology, center of mass C_i indicates the preferred location for node n_i for best controlling its surroundings and to perform, for example, reconnaissance



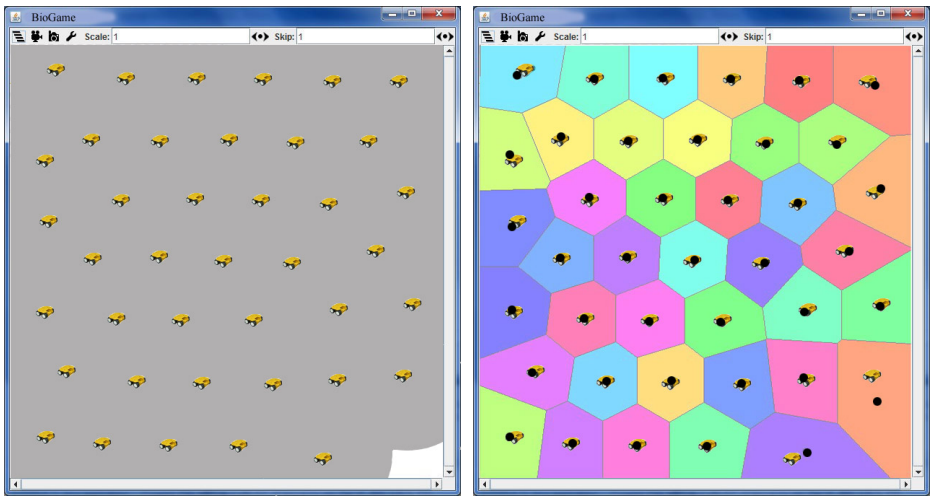
(a) (b)

Figure 12: Example of (a) node distribution and (b) Voronoi tessellation obtained by our BioGame (Chapter 6) at step $t = 5$.



(a) (b)

Figure 13: Example of (a) node distribution and (b) Voronoi tessellation obtained by our BioGame (Chapter 6) at step $t = 15$.



(a) (b)

Figure 14: Example of (a) node distribution and (b) Voronoi tessellation obtained by our BioGame (Chapter 6) at step $t = 50$.

or environment-sensing tasks. Thus, a distance between n_i and C_i indicates how good its present position is to perform a various mission critical tasks. The uniformity measure $\mathscr{U}_C$ is defined as

$$\mathscr{U}_C = \frac{1}{|I|} \sum_{n_i \in I} d(n_i, C_i) \,, \tag{8}$$

where $d(n_i, C_i)$ is the Euclidean distance between node n_i present position and generated by it Voronoi region center of mass C_i, as depicted in Fig. 11.

When network converges to a stable distribution, the separation among neighboring nodes equalizes and for all $n_i \in I$, distance $d(n_i, C_i)$ approaches zero. For example, uniformity measures $\mathscr{U}_C$ for topologies depicted in Figs. 12, 13, and 14, are 7.8, 3.1, and 0.9, respectively. Consequently, in both proposed techniques for assessing uniformity of the network distribution (i.e., $\mathscr{U}_A$ and $\mathscr{U}_C$),

the smaller value achieved by the network indicates better placement of nodes over the area of deployment.

3.7.2 The Average Distance Traveled

Another important metric assessing performances of node self-spreading algorithms is the average distance traveled (ADT) by a mobile node before desired network topology is reached. Since movement is a highly power-consuming operation, reducing the traveling distance for each node is an important task that can significantly extend the lifespan of a network. Let s_i^t represent a strategy of player n_i at time t that corresponds to the spatial coordinates of n_i (i.e., $s_i^t = (x_i^t, y_i^t)$). Furthermore, for all $t \geq 0$, let $d(n_i^0, n_i^t)$ denote the total distance traveled by n_i up to time t. We define $\text{ADT}(t)$ as the average distance traveled by a node until time t as

$$\text{ADT}(t) = \frac{1}{|I|} \sum_{n_i \in I} d(n_i^0, n_i^t) \, . \tag{9}$$

When our simulation experiment progresses and t increases, ADT may rise as well and its value never reduces (i.e., ADT is monotonically increasing function). In simulation experiments, ADT is a monotonically increasing function. When the network reaches a stable distribution, where nodes are not able to improve their positions any further, the derivative of ADT approaches to 0. Note that the rate at which ADT grows indicates how dynamic the network is. When the network reaches a steady distribution with nodes unable to improve their positions any further, the derivative of ADT equals 0, which can be used as an another

indication of the network stability.

When illustrating changes in ADT for our experiments, the vertical axis represents the average total distance traveled by a node up to the time characterized in the horizontal axis.

3.7.3 The Network Area Coverage

The network area coverage (NAC) is a technique for evaluating the effectiveness of self-spreading algorithms. NAC is defined as the ratio of the coverage achieved by the communication areas of all agents to the total deployment terrain. If the region is covered by more than one agent, overlapped area is included in NAC calculations only once. Also, if the communication range of a node extends beyond deployment borders or is blocked by an obstacle, only the part of its coverage that is unobstructed counts towards NAC metrics. Let $A_{\mathrm{C},\,i}$ denote the area covered by node n_i and A_{D} be the size of the area of deployment. We formally define NAC as follows

$$\text{NAC}^{\,t} = \frac{\bigcup_{n_i \in I} A_{\mathrm{C},\,i}}{A_{\mathrm{D}}}\,, \tag{10}$$

where $\bigcup$ represents the union of all locations comprised of the coverage areas of subscribed nodes. A NAC value of 1 implies that the entire area is fully covered while NAC < 1 indicates that some of the deployment terrain may not be adequately monitored. Hence, obtaining the highest possible NAC by mobile agents is one of the goals for our game-theoretic and bio-inspired techniques.

In all of our figures depicting NAC improvements, the vertical axis represents

the ratio of the total deployment terrain covered by at least one node and the horizontal axis represents the duration of the experiments.

4 Node-spreading Potential Game

This chapter presents a distributed game for obtaining a uniform node dispersal among the MANET nodes over a given geographical territory. The proposed node spreading potential game, called NSPG, and the implementation based on it is fully distributed where each node concurrently computes its next preferable location without requiring information on the global network (except for near neighboring nodes) at any node and in any phase of the game. Consequently, each node needs only information about its near surroundings and synchronizes only with its nearest neighbors; as such, it presents an adaptable and easily scalable solution for managing the topology of MANETs in hostile environments.

This part of the thesis is based on studies of NSPG published by us in [33, 34, 37, 38]. Section 4.1 formally introduces NSPG and we analyze its properties and implementation in Sect. 4.2. The results of NSPG simulations are presented in Sect. 4.3. Section 4.4 discusses properties and performance of Rel-NSPG (a resilient NSPG [35]) in situations where the number of nodes in the network change dynamically either due to equipment malfunction or hostile activity.

4.1 Introduction of NSPG

In our NSPG, for each player n_i, a circle centered at n_i with a radius of R_C defines its communication area. A sensing area for node n_i is defined as a circle centered at n_i with a radius of R_S and it describes an area in which n_i is aware of all its neighboring nodes' relative locations. For simplicity and without loss

of generality, our implementation considers a monomorphic population where each player has the same R_C and R_S. We postpone analysis of the relationship between R_C and R_S until additional concepts of our game are spelled out; we revisit it in Section 4.2.

Our NSPG is a distributed game where player n_i makes her decisions based on information about her neighbors using only limited local synchronization. Since it is fully distributed and does not require the entire network to be synchronized, NSPG is also easily scalable. To simplify the implementation of NSPG, we assumed that: *(a)* initially, the network is connected, (i.e., the network is not partitioned into multiple isolated subnetworks), *(b)* each player can determine the relative locations of all neighbors in its sensing area, and *(c)* if a player is moving, no other node in its R_S range can change its location during this time.

Let s_i^t represent a strategy of player n_i at time t, which corresponds to the spatial coordinates of n_i at this time identified as $s_i^t = (x_i^t, y_i^t)$. Let $\boldsymbol{s}^t$ be a strategy profile containing strategies for all $n_i \in I$, which fully determines the network's topology at time t. We define D_i^t to be a set of all neighbors of n_i in its communication area: $D_i^t = \{j \in I : d_{i,j}(\boldsymbol{s}^t) = (0, R_C]\}$, where $d_{i,j}(\boldsymbol{s}^t)$ is the Euclidean distance between nodes n_i and n_j at time t. A set $\bar{D}_i^t = \{n_j \in I : d_{i,j}(\boldsymbol{s}^t) = (0, R_S]\}$ denotes sensing area neighbors of node n_i and, consequently, $D_i^t \subseteq \bar{D}_i^t$. We define $d_{i,min}(\boldsymbol{s}^t)$ to be the distance to the closest neighbor within

communication area of node n_i as

$$d_{i,min}(\boldsymbol{s}^t) = \begin{cases} \min_{n_j \in D_i^t} d_{i,j}(\boldsymbol{s}^t) & \text{if } D_i^t \neq \emptyset \, ; \\ 0 & \text{otherwise .} \end{cases} \tag{11}$$

It is important to note that the strategy profile in $d_{i,j}(\boldsymbol{s}^t)$ refers only the sensing or communication area of node n_i at time t. Note also that $\forall_{n_i}, \forall_{n_j \in D_i^t} \, d_{i,j} > 0$ (i.e., no two nodes may occupy the same physical location) and consequently $d_{i,min}(\boldsymbol{s}^t) = 0$ in Eq. (11) uniquely defines a case in which n_i is disconnected from all other players.

We can now introduce our node-spreading ordinal potential game, NSPG, defined by its payoff function as follows

$$u_i(\boldsymbol{s}^t) = d_{i,min}(\boldsymbol{s}^t) \, . \tag{12}$$

In our NSPG, each $n_i \in I$ moving at time $(t-1)$ has to obey the following condition $\mathscr{C}$

$$\mathscr{C} = \begin{cases} true & \text{if } \left(d_{i,min}(\boldsymbol{s}^t) - d_{i,min}(\boldsymbol{s}^{t-1})\right) > \\ & \quad \sum_{n_j \in \bar{W}} \left(d_{i,j}(\boldsymbol{s}^{t-1}) - d_{i,j}(\boldsymbol{s}^t)\right) \, ; \\ false & \text{otherwise ,} \end{cases} \tag{13}$$

where $\bar{W} = \{n_j \in D_i^t : d_{i,j}(\boldsymbol{s}^t) < d_{i,j}(\boldsymbol{s}^{t-1})\}$ represents n_i's neighbors within its communication area at time t for which the distance decreases as a result of this move. At the beginning of the NSPG (i.e., $t = 1$), $D_i^t \neq \emptyset$ for all n_i since

we assume that the MANET is connected when the nodes are deployed (e.g., when the nodes enter a task area from a common entry point) and, consequently, $\forall_{n_i \in I}\ u_i(\boldsymbol{s}^1) > 0$.

The relative ordinal potential function for our NSPG is defined as

$$\Phi(\boldsymbol{s}^t) = \sum_{n_i \in I} u_i(\boldsymbol{s}^t)\ . \tag{14}$$

Our recursive payoff function defined in Eqs. (12) and (13) is based on the individual objective of increasing the minimal distance to the neighbor(s). However, it prohibits a player from improving her own spatial position by a greater factor than eventual losses she can inflict on adjacent nodes. Although Eq. (12) allows for the value of $u_i(\boldsymbol{s}^t)$ to fluctuate (e.g., as a result of some $n_j \in \bar{D}_i^{t-1}$ moving into the neighborhood of n_i), it never decreases by a factor greater than the improvement of n_i's neighbor that was moving at time $(t-1)$. Our payoff function defined in Eq. (12) encourages an individual player's actions that result in improvement of own payoff, which translates into the objective of finding a location that increases the minimal distance to her neighbor(s). However, the constraint of Eq. (13) prohibits a player from improving her own spatial position by a greater factor than eventual losses inflicted on adjacent nodes. Although Eq. (12) allows for the value of $u_i(\boldsymbol{s}^t)$ to fluctuate over time (e.g., if some $n_j \in \bar{D}_i^{t-1}$ moves into the neighborhood of n_i resulting with $u_i(\boldsymbol{s}^t) < u_i(\boldsymbol{s}^{t-1})$), it never decreases by a factor greater than improvement of n_i's neighbor that was moving at time $(t-1)$, ensuring an overall increase of the potential function defined in Eq. (14). The following theorem formalizes this notion.

Theorem 1. NSPG *is an ordinal potential game and* $\Phi(\boldsymbol{s}^t)$ *is its ordinal potential function, where* $\Phi(\boldsymbol{s}^t)$ *is defined in Eqs. (12), (13), and (14).*

Proof. For all $t > 1$ and all $n_i \in I$, let us define

$$\Delta u_i^t = u_i(\boldsymbol{s}^t) - u_i(\boldsymbol{s}^{t-1}) \tag{15}$$

and

$$\Delta \Phi^t = \Phi(\boldsymbol{s}^t) - \Phi(\boldsymbol{s}^{t-1})\,. \tag{16}$$

Note that Eq. (15) denotes the difference in the individual player's payoff, while Eq. (16) reflects the changes for the entire population. Suppose, for simplicity and without loss of generality, that at any $t > 1$ there is only one node i changing its location. Then, we can rewrite Eq. (2) in terms of NSPG as

$$\forall_{t>1}\ \ (u_i(\boldsymbol{s}_i^t) - u_i(\boldsymbol{s}_i^{t-1})) > 0 \ \text{ iff } \ (\Phi(\boldsymbol{s}^t) - \Phi(\boldsymbol{s}^{t-1})) > 0\,. \tag{17}$$

Also, we define auxiliary variables $\alpha^t = \sum_{n_j \in I \setminus \bar{W}} \Delta u_j^t$ and $\beta^t = \sum_{k \in \bar{W}} \Delta u_k^t$, where the $\bar{W}$ is defined as in Eq. (13), β^t denotes the sum of all utility functions for which distance decreased at time t, and α^t is the sum of increases for remaining players' utility functions (i.e., $\forall_{n_j \in I \setminus \bar{W}}\ u_j(\boldsymbol{s}^t) \geq u_j(\boldsymbol{s}^{t-1})$).

Let us first show that $\Delta u_i^t > 0 \rightarrow \Delta \Phi^t > 0$. If $\Delta u_i^t > 0$, then $n_i \in I \setminus \bar{W}$ and either: player n_i was moving at time $(t-1)$, in which case $\Delta u_i^t > |\beta^t|$ by Eq. (13), or player $n_j \neq n_i$ in $I \setminus \bar{W}$ was moving at time $(t-1)$ with $\Delta u_j^t > |\beta^t|$ and n_i's utility increased as a result of n_j's move. In both cases, at least one player's contribution to the value of α is greater than $|\beta^t|$ and yields $\alpha^t > |\beta^t|$.

We can write $\Phi(\boldsymbol{s}^t)$ as

$$\Phi(\boldsymbol{s}^t) = \sum_{n_j \in I \setminus \bar{W}} (u_j^t) + \sum_{n_k \in \bar{W}} (u_k^t) \tag{18}$$

and, as a result,

$$\begin{aligned}
\Delta\Phi^t &= \left(\sum_{n_j \in I \setminus \bar{W}} (u_j^t) + \sum_{n_k \in \bar{W}} (u_k^t) \right) - \left(\sum_{n_j \in I \setminus \bar{W}} (u_j^{t-1}) + \sum_{n_k \in \bar{W}} (u_k^{t-1}) \right) \\
&= \left(\sum_{n_j \in I \setminus \bar{W}} (u_j^t) - \sum_{n_j \in I \setminus \bar{W}} (u_j^{t-1}) \right) + \left(\sum_{n_k \in \bar{W}} (u_k^t) - \sum_{n_k \in \bar{W}} (u_k^{t-1}) \right) \\
&= \alpha^t + \beta^t \,. \qquad (19)
\end{aligned}$$

Since $\alpha^t > |\beta^t|$ implies $\alpha^t + \beta^t > 0$, we have $\Delta\Phi^t > 0$.

To show that $\Delta\Phi^t > 0 \rightarrow \Delta u_i^t > 0$, we observe that $\Delta\Phi^t > 0$ implies $\boldsymbol{s}^{t-1} \neq \boldsymbol{s}^t$ and there existed player n_i changing her location at time $(t-1)$. Once more, based on Eq. (13), it must be a case that $u_i(\boldsymbol{s}^t) > u_i(\boldsymbol{s}^{t-1})$ and, consequently, $u_i(\boldsymbol{s}^t) - u_i(\boldsymbol{s}^{t-1}) > 0$. □

Theorem 1 shows that our NSPG belongs to the class of ordinal potential games and, as such, is Nash equilibrium-solvable.

4.2 Properties and Implementation of NSPG

In NSPG, the goal for each player n_i is to spread uniformly over an unknown geographical terrain such that the total coverage of the area, defined as the area

covered by communication ranges of all nodes (excluding the overlapping coverage by multiple nodes), is maximized while the network remains connected. Our approach is designed in such a way that each player n_i moves to a new location found by locally running her own GA only if this move will result in an increase of her payoff function value.

4.2.1 Game-theoretic Properties of NSPG

In this section, we prove and justify basic properties, necessary conditions, and optimality of selected parameters of our of NSPG. The following lemma establishes the connection between a player's payoff, her distance to the closest neighbor, and the total coverage area.

Lemma 1. *For any n_i in* NSPG*, an increase in n_i's payoff results in the increase of the total area covered.*

Proof. In a cluster containing only two nodes n_i and n_j, $d_{i,min}(\boldsymbol{s}) = d_{i,j}(\boldsymbol{s})$. An increase of the total area A_T, covered by the communication areas of n_i and n_j, is equivalent to the decrease of an overlapping region $A_O = R_C^2(\theta - sin\theta)$ (colored dark in Fig. 15). We can observe from Fig. 15 that when $d_{i,j}(\boldsymbol{s})$ increases, K and θ decrease (i.e., $d_{i,j}(\boldsymbol{s}^t) > d_{i,j}(\boldsymbol{s}^{t-1})$ iff $K^t < K^{t-1}$ and $\theta^t < \theta^{t-1}$). As $(\theta - sin\theta)$ is strictly increasing on the interval $[0, \pi)$, $\theta < \theta'$ implies $R_C^2(\theta - sin\theta) < R_C^2(\theta' - sin\theta')$ and, consequently, $A_T > A'_T$.

Now suppose that there exist $n_i \in I$ which moves at time $(t-1)$, resulting in $D_i^t > 1$ (i.e., at time t, n_i has more than one neighbor). We know from Eqs. (12)

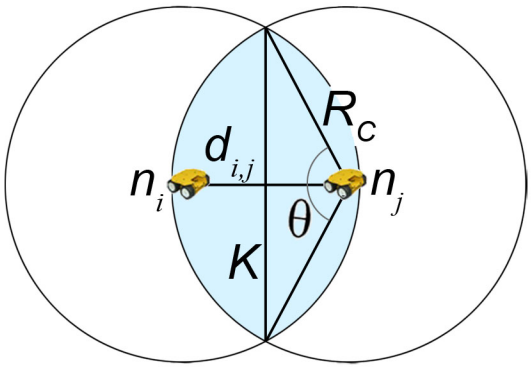


Figure 15: The total communication area coverage achieved by players n_i and n_j.

and (13) that its minimum distance to the closest neighbor at time t compared to time $(t-1)$ has to improve by more than the the sum of eventual decreases in distance with respect to its neighbors. As a result, the sum of distances to all its neighbors increases. Since no other node in n_i's sensing area was moving at time $(t-1)$, an increase in the sum of distances to all neighbors of n_i decreases A_O and increases A_T. □

In the following two lemmas, we state the necessary conditions imposed on our NSPG. First, we show what the relation between values of R_C and R_S should be in order for our NSPG to have a meaningful game structure. Second, we establish a maximum-move distance that a node can relocate at one time unit. This condition is necessary to assure that Eq. (13) is satisfiable for games where nodes further apart than R_S move simultaneously.

Lemma 2. *For each player n_i in* NSPG*, the relation between its sensing and communication areas should obey $(R_C < R_S)$.*

Proof. Let us first consider the case where $R_S < R_C$ for player n_i, implying that her sensing area is smaller than her communication area. Without loss of generality, we can also suppose that, at time t, there exists a node n_j such that $R_S < d_{i,j}(\boldsymbol{s}^t) < R_C$ (i.e., this case would indicate that player n_i can communicate with n_j but have no knowledge of n_j's location). However, this setting represents a situation where n_i is unable to evaluate its own position. Consequently, the relation $R_S < R_C$ is unsuitable for any GT-based implementation, since it represents a game of incomplete information where n_i cannot assess her own payoff.

Let us now consider the case where $R_S = R_C$ (i.e., player i's sensing area is the same as its communication area). Then, there may exist a player $n_j \notin \bar{D}_i^t$ such that $d_{i,j}(\boldsymbol{s}^t) = R_S + \epsilon$ for some small positive ϵ. In this case, n_i and n_j are not aware of each other's existence and any relocation of n_i towards n_j by a distance greater than ϵ may result in a reduction of its payoff. Therefore, player n_i cannot make any prediction about its next location, except for the trivial decision of staying immobile.

As a result, the only valid relation between sensing and communication ranges is when $R_C < R_S$, representing the situation in which player n_i's knowledge of its surrounding area is greater than its communication area. □

Lemma 3. *For each player n_i with its R_C and R_S restricted by Lemma 2, the maximum possible move distance M_{max} at one time is required to be not greater than $(R_S - R_C)/2$.*

Proof. Assume to the contrary that M_{max} is greater than $(R_S - R_C)/2$ by a value of $\gamma > 0$. Also, suppose that at time t there is player n_j located at distance $d_{i,j} = R_C + \epsilon$ from player n_i, where $0 < \epsilon < 2\gamma$. Therefore, players n_i and n_j are unaware of each other's existence and may move simultaneously. If, based on their respective GA results, both n_i and n_j are moving at time t towards each other by $((R_S - R_C)/2 + \gamma)$, then, at $t+1$ the distance $d_{i,j}(\boldsymbol{s}^{t+1}) = R_C + \epsilon - 2\gamma$ and, consequently, $d_{i,j}(\boldsymbol{s}^{t+1}) < R_C$. As an effect of this move, the distance between n_i and n_j may become worse than n_i's minimum distance at time t (i.e., $d_{i,j}(\boldsymbol{s}^{t+1}) < d_{i,min}(\boldsymbol{s}^t)$), which represents an unpredictable non-profitable move for player n_i that may invalidate condition $\mathscr{C}$ in Eq. (13).

If, on the other hand, we define the maximum possible movement distance as $M_{max} = (R_S - R_C)/2$, any simultaneous move of two players n_i and n_j unaware of each other existence at time t cannot result in $d_{i,j}(\boldsymbol{s}^{t+1}) \leq R_C$ and has no influence on payoff functions for both n_i and n_j. Therefore, the maximum safe distance that a node can relocate in our game is defined as $M_{max} = (R_S - R_C)/2$. □

In Lemmas 2 and 3, we established the minimum possible value of R_S with respect to a node's R_C and confined the node's maximum move distance not to exceed half the difference between R_S and R_C. From Lemmas 2 and 3, we can see that increasing R_S with respect to R_C results in an increase in M_{max}, which suggests that, in our game, at each step a greater improvement of each individual payoff function is possible, yielding an efficient convergence towards maximum

coverage. However, we should observe that increasing R_S will have negative implications for the applicability of our game to MANETs: *(a)* R_S defines an area that requires full synchronization and knowledge of all neighbors' locations within moving node sensing radius; a larger R_S may lead towards a fully synchronized game and *(b)* using larger R_S will imply more power consumption for each player, violating one of the major principles of MANETs. Therefore, there is a trade-off in the choice of R_S.

In order to ensure efficient performance of our NSPG with reduced synchronization among the closest neighbors, each node's sensing radius is set up to be twice its communication area radius. Lemmas 2 and 3 together with the following proposition, which formalizes the most appropriate relation between R_S and R_C, establish all the necessary parameters for movement of a node that may affect NSPG performance.

Proposition 1. *We choose* $(R_S/R_C) = 2$ *as the best value for* NSPG *implementation.*

Proof (sketch). According to Lemmas 2 and 3, increasing R_S/R_C ratio leads to an increase of M_{max} and, consequently, faster network stabilization. However, with the increase of R_S, the sensing area for a player grows exponentially while M_{max} grows only linearly. As R_S defines an area in which i has to synchronize its actions with all other nodes within it, increasing R_S leads rather quickly to the fully synchronized network in which each node has complete topology information of the network. Consequently, it is impractical to choose $R_S >> R_C$,

as the benefit of increased M_{max} will be impaired by the overhead to maintain a large synchronization area. On the other hand, as R_S/R_C approaches 1, M_{max} gets close to 0, and hence the number of steps required to converge towards a maximum coverage approaches infinity.

In the light of the above arguments, we balanced the effectiveness of a large M_{max} and convenience of a small R_S by selecting $(R_S/R_C) = 2$. This choice for R_S/R_C also coincides with research results that use 2-hop knowledge which is a common minimum requirement for efficient routing algorithms in MANETs [62, 65, 75]. □

In our approach, the key objective is to increase the total area covered by all nodes through a uniform node distribution with maximized distances among neighboring nodes. However, another important attribute that determines overall MANET performance is its nodes' ability to communicate with neighbors to exchange collected information or participate in aggregated computing. Consequently, even though a disconnected node increases total terrain coverage, as can be inferred from Lemma 1, an isolated node cannot enhance overall network performance in any way. Also note that a single disconnected node contradicts our objective to provide uniform node distribution over deployment terrain. One of the direct consequences of our NSPG setup in Eqs. (12) and (13) guarantees that no node makes a move that results in a single node's being disconnected. In the following lemma, we formalize this basic connectivity property of NSPG.

Lemma 4. *In* NSPG*, no node makes a move that results in a single node's being*

disconnected.

Proof. Recall that in our model, all nodes are initially connected and no two nodes may occupy the same physical location simultaneously. Suppose that at time $\bar{t} > 1$ node n_j becomes isolated, then $\forall_{t<\bar{t}}, \forall_{n_i \in I}\ D_i^t \neq \emptyset$ and $\exists_{n_j}$ at time $\bar{t}$ such that $D_j^{\bar{t}} = \emptyset$. Node n_j could become isolated at time $\bar{t}$ as a result of its own move at $(\bar{t}-1)$, or as a consequence of someone else's leaving it isolated. If n_j were moving at $(\bar{t}-1)$, then $u_j(\boldsymbol{s}^{\bar{t}-1}) > 0$, but $u_j(\boldsymbol{s}^{\bar{t}}) = 0$ and this move would contradict the NSPG definition in Eqs. (12) and (13).

If, on the other hand, n_j's neighbor n_k is moving at time $(\bar{t}-1)$, leaving node n_j isolated, then, from Lemma 3 and Proposition 1, the distance between n_k and n_j would have to obey $R_C/2 < d_{k,j}(\boldsymbol{s}^{\bar{t}-1}) \leq R_C$. Also, $0 < \Delta u_k^{\bar{t}} \leq R_C/2$ follows from Eq. (13) and Lemma 3 (where $\Delta u_k^{\bar{t}}$ is defined as in the proof to Theorem 1). By combining these two inequalities, we see that $\Delta u_k^{\bar{t}} \leq R_C/2 < d_{k,j}(\boldsymbol{s}^{\bar{t}-1})$. However, $\Delta u_k^{\bar{t}} < d_{k,j}(\boldsymbol{s}^{\bar{t}-1})$ violates the condition of the NSPG payoff function given in Eq. (13) and as such could not be performed.

Since all the cases contradict our initial assumption that there exists time $\bar{t}$ such that $D_j^{\bar{t}} = \emptyset$ for any $n_j \in I$ connected at time $(\bar{t}-1)$, we may conclude that in NSPG any move resulting in a disconnected node cannot be performed. □

The following theorem establishes connection among our potential function defined in Eq. (14), NE of our NSPG, and the total area coverage.

Theorem 2. *In* NSPG*, there exists* $v = \Phi(\boldsymbol{s}^{\bar{t}})$*, where* v *is the maximum value of our ordinal potential function and* $\boldsymbol{s}^{\bar{t}}$ *is* NE *of* NSPG*, such that the total area*

covered by the nodes playing strategy profile $\boldsymbol{s}^{\bar{t}}$ cannot be further improved.

Proof (sketch). From Theorem 1, we know that an improvement in each player's payoff is reflected as an increase in our game's ordinal potential function value (Eq. 14). Each node will seek to increase its distance to its neighbors until this distance is equal to R_C (or less than R_C for overcrowded networks such that the distance cannot be further increased).

Let the strategy profile in which no player can further improve her payoff be achieved at some step $\bar{t}$. At this point, all improvements to $\Phi(\boldsymbol{s})$ occurred until $(t < \bar{t})$ and for all $\forall_{t \geq \bar{t}}\ \Phi(\boldsymbol{s}^t)$ can be represented as $\Phi(\boldsymbol{s}^{t+1})$. Since $v = \Phi(\boldsymbol{s}^{\bar{t}})$ is the largest value that NSPG can achieve, $\boldsymbol{s}^{\bar{t}}$ is NE of this game.

Furthermore, Lemma 1 shows that increase the value of an individual player's payoff function is correlated with the increase in an area coverage. Therefore, we can conclude that after $\Phi(\boldsymbol{s}^{\bar{t}})$ reaches its maximum value v, which happens when strategy profile $\boldsymbol{s}^{\bar{t}}$ is NE, the total area covered represented by $\boldsymbol{s}^{\bar{t}}$ has also been maximized for this game. □

As a direct result of Lemma 1 and Theorems 1 and 2, we can state the following corollary:

Corollary 1. NSPG *converges to a stable point (i.e.,* NE*) at which no further improvement for the network coverage is possible.*

4.2.2 Determining a New Location by GA

In our approach, a GA is used by each mobile node to find the most beneficial new position with a low overhead [56, 68]. This GA generates a solution for the *fittest* location to which a node can move in the next time unit. In general, a GA requires two elements to be defined: (*a*) chromosomes representing the problem space, and (*b*) a fitness function in order to evaluate the quality of each candidate-chromosome with respect to the given goal.

The problem space for player n_i at t is its set of possible solutions (chromosomes) at this time (i.e., the set of locations within n_i's $R_C/2$ radius). A chromosome is encoded as binary strings and represents feasible node locations. In our implementation of NSPG the length of a chromosome is 18 bits. The fitness function, defined as in Eq. (12), prescribes the optimality of a solution so that a particular chromosome is ranked against all the other chromosomes. Each mobile node runs GA for 40 generations or until an optimal new location is found. A population consists of 20 chromosomes (new candidate-positions that meet Eq. (13) requirements). During reproduction, single-point crossover is applied to a randomly selected couple of chromosomes to generate two new offspring for the next generation. The selection mechanism is the roulette wheel selection. The mutation operator is applied over all genes in a population to introduce genetic diversity (new chromosomes) and prevents the GAs from getting stuck at a local optimum. In our implementation, genes in a population may arbitrarily change with a probability rate of 0.01.

Each autonomous mobile node gathers information about its neighboring environment including mobile nodes and obstacles within its sensing range of R_S. Our GA generates new chromosomes representing candidate solutions for the next generation at each iteration (i.e., generation). These candidate solutions are ordered from the lowest fitness score to the highest. The lowest (i.e., the best) fitness value corresponds to the solution representing the least amount of virtual force applied to the corresponding mobile node. After 40 generations, the best candidate solution having the lowest fitness score is adopted as the new speed and movement direction such that the total force on the moving node in a new location will be lowered.

In NSPG, player n_i collects location information, advertises its intention to its neighbors, and then finds a new location to move using the GA. After moving to a new location, player n_i advertises that it has completed the operation. The NSPG stabilizes when there are no nodes that can improve their own payoff functions as assured by Theorem 2.

4.2.3 NSPG Implementation

An implementation for NSPG is given in Algorithm 1. In `Update Neighbor Information` player n_i, who wants to change her current location, gathers the locations of her neighbors within her R_S radius. In `Advertise To Move`, player n_i advertises her willingness to moving to all the nodes within her R_S range (note that if there is a collision among multiple nodes trying to move, several methods can be employed such as binary exponential back-off, etc.).

Once player n_i assumes that she is the only one authorized to move within

Algorithm 1 NSPG Implementation

```
for all players n_i ∈ I that can improve u_i do
  Update Neighbor Information;
  Advertise To Move;
  if sucess then
    Find New Location;
    Move To New Location;
  else
    Wait for another turn;
  end if
end for
```

its sensing area, `Find New Location` decides the most beneficial position to move, which is a location within its M_{max} range that increases value of its payoff. We find the most beneficial new position with a genetic algorithm. After moving to a new location, player n_i advertises the completion of its operation. Algorithm 1 stabilizes when there are no nodes that can improve their own payoff function.

Figure 16 shows different distance relations between two nodes n_i and n_j.

In Figure 16(a), $d_{i,j}(\boldsymbol{s}^t) < R_C$, and players n_i and n_j are within their communication areas and one of them can move. If n_i wants to move, it has to advertise its intention to n_j and eventually to other nodes in its R_S range (not shown in Fig. 16(a)). If n_j is the only neighbor of n_i, the new location is no further than R_C away from n_j, as assured by Lemma 4. Also, the new location is chosen to guarantee improvement of $u_i(\boldsymbol{s}^{t+1})$ according to Eqs. (12) and (13).

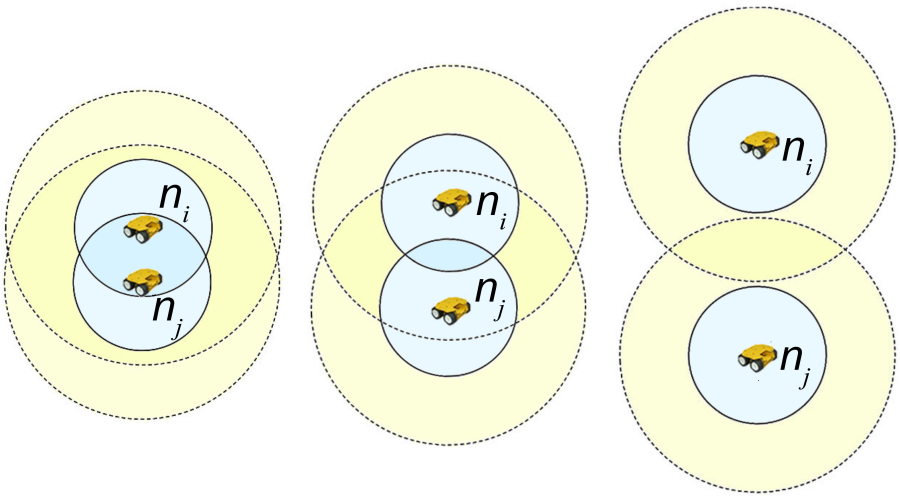


Figure 16: Relative positions of two players n_i and n_j in NSPG.

In Fig. 16(b), $R_C < d_{i,j}(\boldsymbol{s}^t) < R_S$ and n_j is in n_i's sensing, but not communication, area. In this example, n_i and n_j are aware of each other's locations and only one of them can move at a time. Let us assume that i is intending to move. If $d_{i,j}(\boldsymbol{s}^t) < R_C + M_{max}$ and n_i is moving towards n_j by the maximum possible distance of M_{max}, then at time $(t+1)$ $d_{i,j}(\boldsymbol{s}^{t+1}) < R_C$ and, consequently, $n_j \in \bar{W}$ and $|d_{i,j}(\boldsymbol{s}^{t+1}) - d_{i,j}(\boldsymbol{s}^t)|$ must be less than $(d_{i,min}(\boldsymbol{s}^{t+1}) - d_{i,min}(\boldsymbol{s}^t))$ in order to satisfy the condition in Eq. (13). If $d_{i,j}(\boldsymbol{s}^t) > R_C + M_{max}$ or n_i is not moving towards n_j, then n_i is free to move without regard to n_j's location. Note that if there are any other nodes present in n_i's R_S range, they have to be taken into consideration by n_i at the time of selecting a new location. In Fig. 16(c), $R_S < d_{i,j}(\boldsymbol{s}^t)$ and both n_i and n_j are not aware of each other's existence and, consequently, both are able to perform relocations at the same time.

4.3 Simulation Experiments for NSPG

Our software implementation of NSPG consists of nearly 4,000 lines of algorithmic Java code that utilizes MASON libraries [44]. It provides a realistic NSPG simulation tool with a real-time visualization of ongoing network dynamics as an experiment run progresses. Similar to the modeling platform for BioGame (Sect. 6.2 in Chapter 6), our software to evaluate performance of NSPG has the ability to run experiments with a previously used initial mobile node distribution and initial conditions (i.e., the initial data for each mobile node includes its starting coordinate) for generating comparable experiment results. In our study, each experiment was repeated 30 times to eliminate noise in the collected data and obtain accurate stochastic behavior of our NSPG.

For each experiment, the area of deployment was set to 100×100 square units. Initially, the nodes were placed at the center of the deployment area and do not have knowledge of their underlying terrain and locations of neighbors. This initial distribution represents realistic situations where the nodes enter a terrain from a common entry point (e.g., starting node deployment into an earthquake area from a single location) compared to random or other types of initial distributions we see in the literature. The simulation experiments were performed by deploying nodes into two distinct terrains: one containing several strategically placed obstacles, which prevent nodes from moving and communicating through them, and the second with no obstacles. In the unobstructed terrain, autonomous node movement was limited only by the borders confining the area of

the experiment. We ran experiments using networks with $10, 20, \ldots, 60$ nodes and communication ranges of $R_C = 10, 15,$ and 20.

The snapshot in Fig. 17 shows a typical initial distribution of 30 autonomous nodes in the deployment area with obstacles. The black dots and circles around

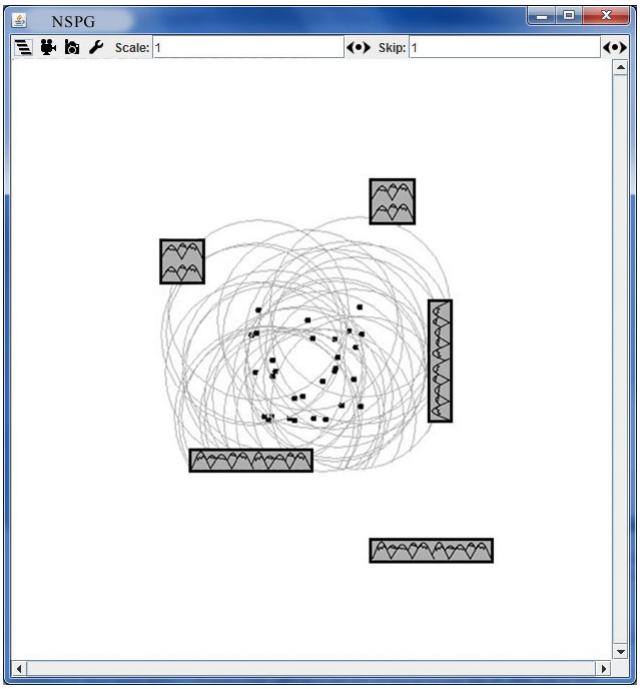


Figure 17: An initial deployment of 30 nodes with $R_C = 15$.

them represent the node locations and their communication areas, respectively. The gray rectangular shapes denote obstacles that restricts nodes' communication and movement. For clarity, the snapshots demonstrated in this paper do not depict nodes' sensing areas.

A typically obtained final distribution of 30 nodes in the unobstructed terrain is shown in Fig. 18, while Fig. 19 shows a final distribution of the same popula-

tion deployed into the area with obstacles. We can observe in Figs. 18 and 19

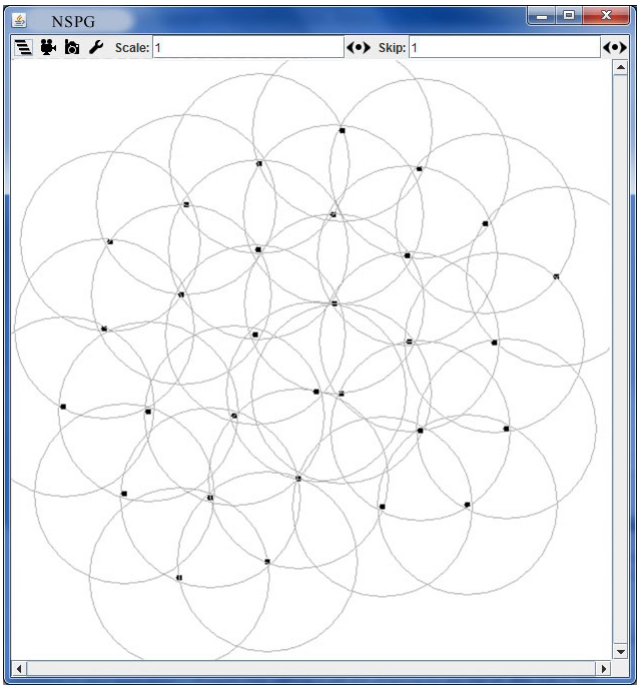


Figure 18: A typical final network topology for 30 nodes with $R_C = 15$ distributed over the unobstructed terrain.

that the nodes can achieve an almost uniform area coverage while maintaining network connectivity. Another observation that can be noted from Fig. 19 is that even in the presence of obstacles, which limit an individual node's coverage area as well as its movement, the achieved total area covered by 30 nodes is comparable to the area covered by the same number of nodes deployed into the unobstructed terrain.

We present in Figs. 20–22 NAC improvement for networks with different

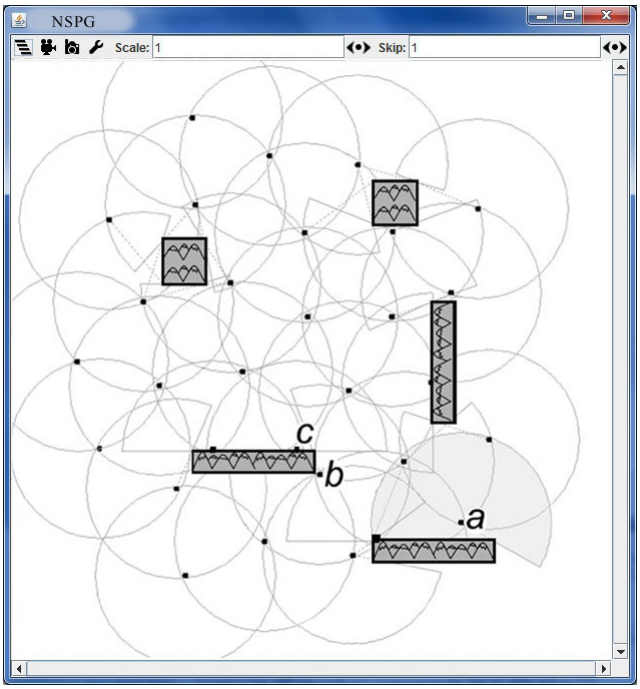


Figure 19: A typical final network topology for 30 nodes with $R_C = 15$ distributed over the the terrain with strategically placed obstacles.

numbers of nodes and varying R_C that are deployed into unobstructed and obstructed geographical territories. In these figures, one time unit (step) stands for the interval needed by a node to evaluate its current position, process GA, and move from one location to another. Note that at each step, more than one node can move and their actions need not be coordinated as long as they are more than R_S distance away from each other.

Figure 20 shows the NAC for networks with 20 nodes and R_C values of 10, 15, and 20. We see in this figure that the area covered by nodes increases with the increase in an individual node's R_C, as the total possible area covered by nodes is proportional to the their communication areas. The maximum area coverage

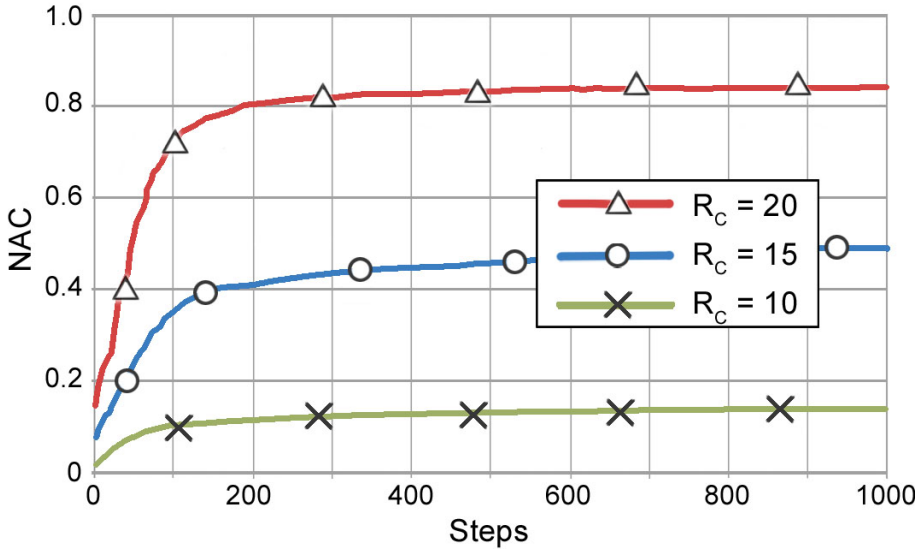


Figure 20: NAC for networks with 20 nodes and $R_C = 10, 15, 20$.

achieved by networks with 20 nodes and $R_C = 10, 15, 20$ are 0.23, 0.54, and 0.85, respectively.

We also observe in Fig. 20 that although every step improves NAC, the most significant improvements occur during the initial iterations of NSPG, demonstrating the effectiveness of GA in finding the fittest new location, especially in the early stages. For example, the network with 20 nodes and $R_C = 20$, which has the maximum NAC value of 0.86, reaches NAC of 0.75 at time $t = 100$ and NAC of 0.83 at time $t = 200$. This observation shows that during the last 800 steps of our simulation, there was only miniscule improvement in NAC for this example and nodes were able to disperse far from their original locations in the early stages of the experiment.

Figure 21 shows the improvement in NAC for networks with different num-

bers of mobile agents, $R_C = 15$, and a deployment area with strategically placed obstacles. We can observe that networks of larger population require more time

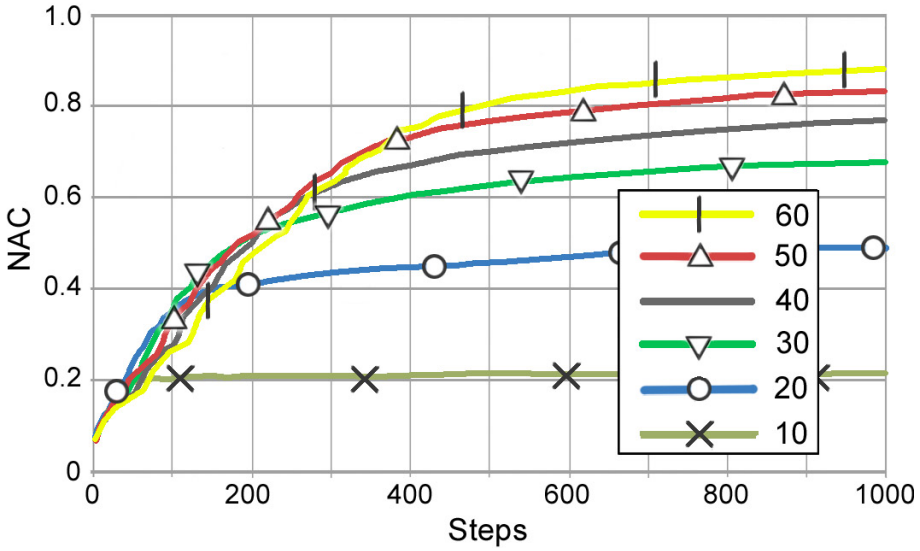


Figure 21: NAC improvement for $R_C = 15$ and networks consisting of $10, 20, ..., 60$ nodes.

to achieve their highest coverage, as more nodes need to disperse in densely populated networks. However, NAC values increase notably as the number of nodes deployed in the same geographical area increases. Another observation is that, for all networks used in these experiments, the nodes were able to effectively navigate around existing obstacles in order to improve area coverage while maintaining network connectivity. We can see in Fig. 21 that networks with populations of 10, 30, and 60 nodes achieved relative NAC values of 0.30, 0.71, and 0.90. We executed the same set of experiments for nodes deployed into the unobstructed region from which similar conclusions on the performance of our NSPG were reached.

In Fig. 22, we show NAC values with respect to the number of steps and various communication areas achieved by nodes deployed into unobstructed terrain (denoted by the gray lines) and terrain with obstacles (represented by black lines). We can see that in all three cases (i.e., $R_C = 10, 15,$ and 20), the total area

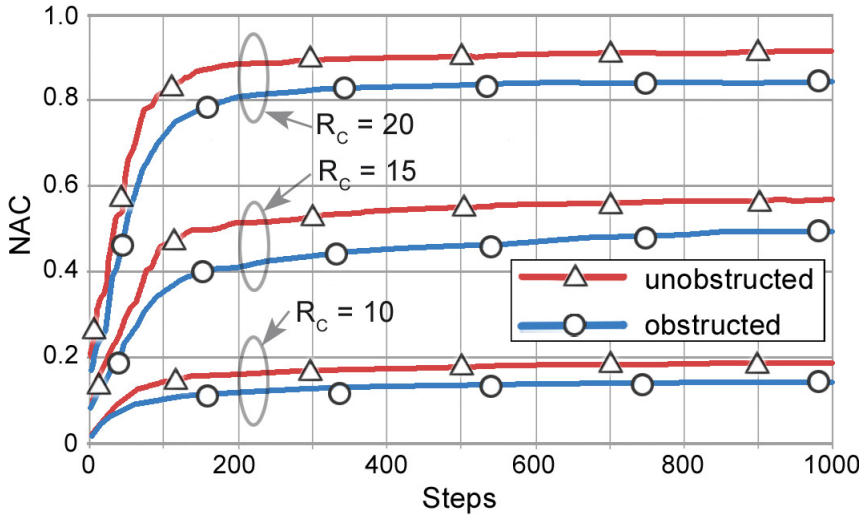


Figure 22: NAC for networks with 20 nodes with $R_C = 10, 15, 20$ distributing themselves over unobstructed terrain (denoted by lines marked $\triangle$) and terrain with strategically placed obstacles (denoted by lines marked $\bigcirc$).

covered by the same number of nodes is greater in the terrain without obstacles than in the territory with strategically placed barriers. Less efficient coverage of obstructed terrain results from limitations in an individual node's communication area coverage due to the nearby obstacle (e.g., node a in Fig. 19) as well as nodes' inability to maximize the distance to their neighbors located behind the obstacle (e.g., the distance between nodes b and c in Fig. 19). Since the nodes deployed into the unobstructed terrain have more freedom of movement, higher

area coverage is achieved sooner in open areas than when they have to navigate obstacles. For example, in the unobstructed terrain, 20 nodes with $R_C = 15$ can reach 0.9 of their maximum achievable area coverage after approximately 140 steps while the same population needs approximately 250 steps in a terrain with obstacles.

We can conclude from Fig. 19 that even when the terrain contains obstacles at unknown locations, the appropriate coverage achieved by the NSPG follows the same pattern as in the case of unobstructed terrain, proving the effectiveness of NSPG to maximize area coverage in a broad range of possible situations.

Our simulation results show that NSPG combined with GA can be effective in providing a satisfactory level of area coverage with near-uniform node distribution while utilizing only local information by each autonomous agent. The nodes are able to position themselves in an unknown geographical terrain to maximize area coverage without global coordination and *a priori* knowledge of a deployment terrain.

4.4 Resilient NSPG

Building on our NSPG (see Sects. 4.1 – 4.3) we present a resilient node-spreading potential game, named Rel-NSPG, for uniform distribution of MANET nodes over an unknown territory. We extend NSPG analysis to deployment areas with mobile agents experiencing hostile activity, resulting in nodes' occupying an area under attack to stop working, and the malfunction of randomly selected agents.

4.4.1 Characteristics of Rel-NSPG

We proved in Theorem 2 the existence of a stable state in NSPG. However, in the presence of hostile attacks or disabled nodes, even if the network reaches the stable state $\boldsymbol{s}^{\bar{t}}$, its equilibrium can be disturbed. In this section, we study how nodes running Rel-NSPG behave and to what degree the dynamical changing size of the network population may affect NAC and network connectivity.

Let $\boldsymbol{s}_{max}$ denote a strategy profile in which each player's distance to its closest neighbor is R_C (i.e., $\forall_{n_i \in I}\ d_{i,min}(\boldsymbol{s}_{max}) = R_C$) and let NAC$(\boldsymbol{s})$ represent NAC determined by a strategy profile $\boldsymbol{s}$. The relation between a strategy profile $\boldsymbol{s}_{max}$ and NE is stated by the following lemma.

Lemma 5. *A strategy profile $\boldsymbol{s}_{max}$ is* NE *of Rel-*NSPG.

Proof (sketch). Since $\forall_{n_i \in I}\ u_i(\boldsymbol{s}_{max})$ represent the maximum possible payoff that player i can receive in Rel-NSPG, no player has an incentive in a unilateral change of her strategy. Therefore, strategy profile $\boldsymbol{s}_{max}$ corresponds to NE of Rel-NSPG and, as a consequence of Theorem 2, no further improvement for NAC$(\boldsymbol{s}_{max})$ is possible. □

In Lemma 5, we show that if the total deployment area, which is denoted as A_{D}, is greater than the maximum possible coverage obtainable by $|I|$ nodes, $\boldsymbol{s}_{max}$ is the NE of our Rel-NSPG. As a direct result of Lemma 5, we can present the following corollary for the relationship between A_{D}, over which the nodes are spreading, and NAC$(\boldsymbol{s}_{max})$.

Corollary 2. *The total deployment area* $A_D \geq \text{NAC}(\boldsymbol{s}_{max})$.

Let us now consider the case where A_D is smaller than the maximum possible coverage that $|I|$ nodes can provide. This situation corresponds to densely populated networks with more nodes than needed to cover the area. The following lemma states that Rel-NSPG can still stabilize with some strategy profile $\boldsymbol{s}^* \neq \boldsymbol{s}_{max}$ as NE of densely populated networks.

Lemma 6. *For a given* $|I|$ *and* A_D *in a densely populated network, there exists a strategy profile* $\boldsymbol{s}^*$ *that is* NE *of Rel-*NSPG *with* $\text{NAC}(\boldsymbol{s}^*) < \text{NAC}(\boldsymbol{s}_{max})$*, where* $\boldsymbol{s}_{max}$ *is the strategy profile for the maximum obtainable coverage.*

Proof (sketch). Suppose the simplest case in a densely populated network with more nodes than needed to cover the area, where each node has three neighbors. In this setting, $\exists_{s_i^* \in S_i}$ for player n_i such that $d_{i,min}(s_i^*)$ represents an equidistance to all her neighbors and such that s_i^* cannot be further improved. By iterating this argument upon the neighbors of n_i's neighbors, we can construct the strategy profile $\boldsymbol{s}^* = \{s_1^*, s_2^*, \ldots, s_n^*\}$ in which no player can improve her location in future iterations of the game. Therefore, $\boldsymbol{s}^*$ is NE of Rel-NSPG, $\text{NAC}(\boldsymbol{s}^*) < \text{NAC}(\boldsymbol{s}_{max})$, and $d_{i,min}(\boldsymbol{s}^*) < R_C$ for all $n_i \in I$. □

We can generalize the results of Lemma 6 in the following corollary.

Corollary 3. *The number of nodes* $|I|$ *represented in* NE *strategy profile* $\boldsymbol{s}^*$ *must be greater than the optimal population size needed to cover* A_D.

The next theorem proves that Rel-NSPG is capable of reaching NE strategy even after losing nodes as the result of hostile activity, resulting in losses concentrated in a particular region.

Theorem 3. *In Rel-*NSPG*, if one or more nodes become disabled at a time $t_A \geq \bar{t}$, where $\bar{t}$ is defined as in Theorem 2, Rel-*NSPG *will reach a stable state representing* NE *for the remaining nodes.*

Proof (sketch). In sparsely populated networks, strategy profile $\boldsymbol{s}^{\bar{t}}$ defines the NE described in Lemma 5. At ($t \geq t_A$), for all n_i that remained active after the loss of nodes, $d_{i,min}(\boldsymbol{s}^t)$ = R_C and, as such, represents NE for the reduced number of nodes.

For densely populated networks, strategy profile $\boldsymbol{s}^{\bar{t}}$ is NE of the type described in Lemma 6. In this case, the area where losses occurred are not adequately covered by nodes and $\exists_{n_i \in I}$'s being able to increase its payoff functions in the future. As a result, these node(s) start participating in Rel-NSPG to improve their payoffs. Consequently, it follows from Theorem 2 that the potential function improves at each step until reaching its maximum value at time after t_A, hence its NE for the reduced number of nodes. □

4.4.2 Rel-NSPG Simulation Experiments

Similar to simulations of our NSPG, the area of deployment for Rel-NSPG was set to 100×100 square units and the initial population was placed randomly in the center of the deployment terrain. We ran experiments for networks with 40

nodes, each node with R_C = 20. Each simulation experiment was run for 1,000 steps and repeated 20 times and the results were averaged in order to reduce experimental noise. At time $t = 200$, we implemented the hostile attack that disabled all nodes occupying 20% of the deployment area located in the southeast corner of the deployment terrain. After the attack, the Rel-NSPG runs for 400

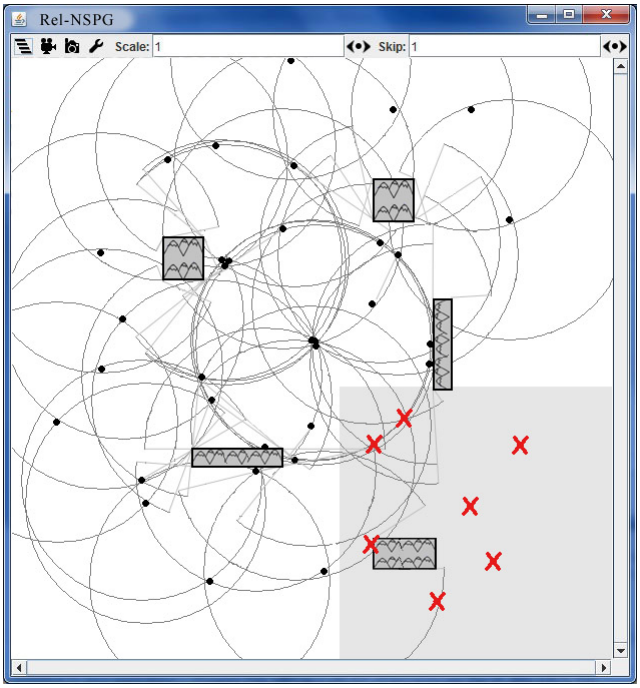


Figure 23: Topology snapshots of the network experiencing the hostile attack ($t = 200$).

steps without any disturbance and then, at $t = 600$, 10 additional randomly selected nodes became disabled representing equipment malfunction. The disabled nodes remain inactive for the rest of the simulation.

Figure 23 shows a typical node distribution after the attack at $t = 200$. We

can observe in Fig. 23 that the attack destroyed seven nodes, eliminating the area covered by them. Between $t = 200$ and $t = 600$, the remaining nodes were able to compensate for losses in area coverage by finding new locations. The network topology shortly before the random node malfunctions can be seen in Fig. 24.

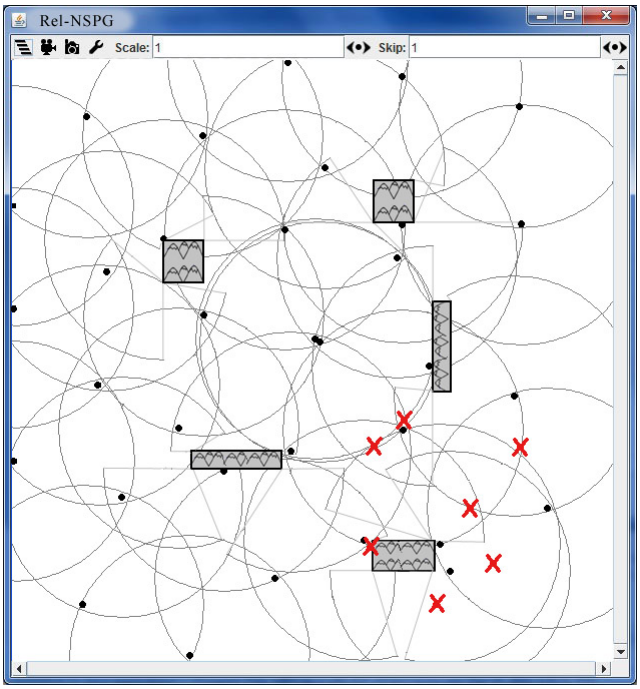


Figure 24: Topology snapshots of the network before the node malfunctions ($t = 599$).

Figure 25 shows the network topology after 10 randomly selected nodes became disabled (i.e., $t = 600$). The final mobile node distribution after running Rel-NSPG for 1,000 steps is presented in Fig. 26, where we can observe that remaining nodes readjusted their positions to compensate for the missing agents.

Figure 27 shows NAC improvement when the nodes undergo hostile attack at

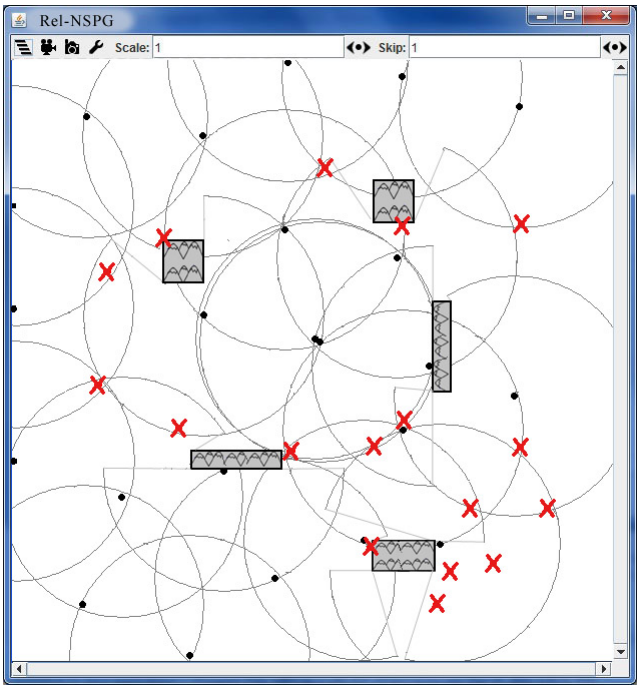


Figure 25: Topology snapshot of the network at $t = 600$ after malfunction of randomly selected nodes.

time $t = 200$ and the random malfunctions at time $t = 600$. We can observe that the mobile nodes achieve coverage of 0.93 shortly before the attack. After the attack (i.e., $t = 200$), there is a NAC drop to 0.83 due to the loss of seven nodes. NAC recovers, reaching the value of approximately 0.92. A less significant NAC loss occurs at time $t = 600$, when additional 10 randomly chosen nodes became disabled. The main recovery task for the remaining nodes after time $t = 600$ is to uniformly cover the area with remaining resources. We see in Fig. 26 that our Rel-NSPG is able to recover after additional nodes became inactive, converging towards uniform area coverage.

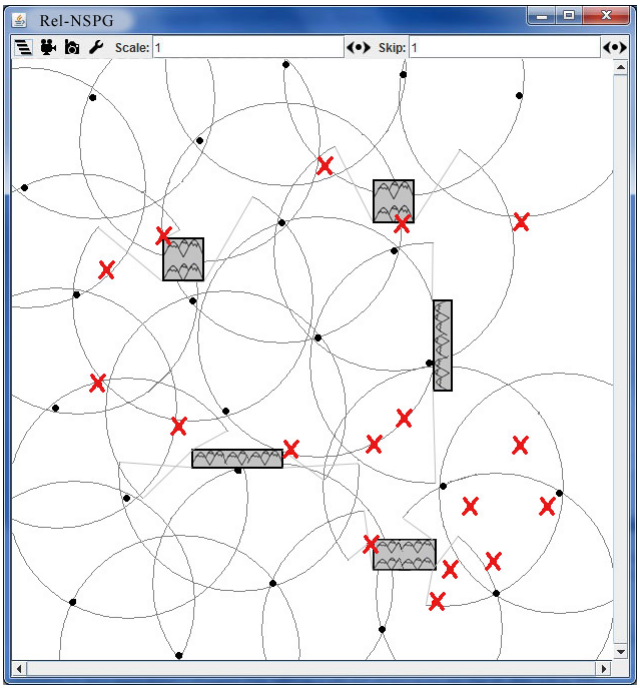


Figure 26: Topology snapshot of the network at $t = 1000$ after recovering from malfunction of randomly selected nodes.

4.5 Observations

NSPG is a distributed potential game that depends only on local information to maximize the area coverage achieved by all nodes. In our approach, a GA is used to determine the next best location to move by evaluating qualities of candidate positions with the node's payoff function. We proved that the total area covered by nodes improves in each step until a stable state (i.e., NE) is reached, which represents a stable solution for the network. In our experiments, the nodes were deployed in an area with and without strategically placed obstacles limiting their communication and movement abilities and suffering from node losses

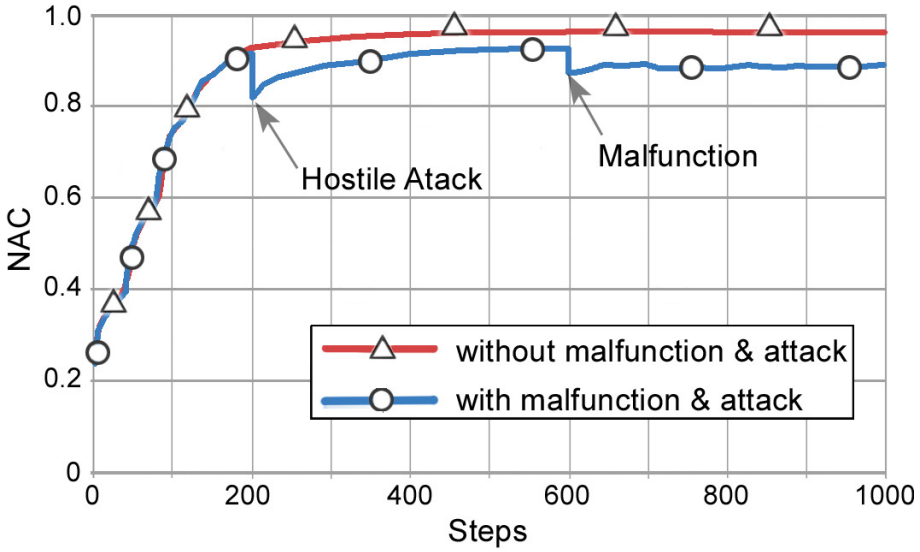


Figure 27: NAC improvement for 40 node network with $R_C = 20$ experiencing hostile attack at step 200 and malfunction of randomly selected nodes at step 600.

due to hostile attacks and malfunction of randomly selected mobile agents. The simulation experiments show that NSPG performs effectively and efficiently for various network densities, node communication ranges, and terrain characteristics such as obstacles.

5 Node-spreading Evolutionary Game

In this chapter, we present a distributed and scalable evolutionary game played by MANET nodes to place themselves uniformly over a dynamically changing environment without a centralized controller. A node-spreading evolutionary game, called NSEG, runs at each mobile node autonomously, making movement decisions based on localized data evaluated by a spatial game while the movement probabilities of possible next locations are assigned by our FGA. NSEG was first presented in [31] and its formal analysis extended in [36].

We formally introduce NSEG in Sect. 5.1. Details of a process by which each player evaluates her current position is given in Sect. 5.2, and Sect. 5.3 describes our spatial game as run by each node. We analyze convergence of NSEG and prove that the optimal network topology is evolutionary stable and once reached, guarantees network stability in Sect. 5.4. Section 5.5 demonstrates that NSEG performs well with respect to network area coverage, uniform distribution of mobile nodes, and convergence speed.

5.1 Introduction of NSEG

In NSEG, the goal for each node is to distribute itself over an unknown geographical terrain in order to obtain high coverage of the area by nodes and to achieve uniform node distribution while keeping the network connected. Initially, nodes are placed in a small subsection of a deployment territory simulating a common entry point in the terrain. This initial distribution represents realistic situations

(e.g., starting node deployment into an earthquake area from a single entry point) compared to random or any other types of initial distributions we see in the literature. In order to model our game in a discrete domain with a finite number of possible strategies, we transpose the nodes' physical locations onto a two-dimensional square lattice. Consequently, even though the physical location of each node is distinct, each logical cell may contain more than one node.

For each node, the set of neighboring cells is defined with respect to its location and its communication radius (R_C). For example, $R_C = 1$ indicates that each node can communicate with all nodes in the same cell as well as nodes located in its adjacent 8 cells (i.e., all cells within a Chebyshev distance smaller or equal to 1) resulting in the set of 9 neighboring cells. In our NSEG, for simplicity and without loss of generality, the communication radius is selected as $R_C = 1$ for all nodes; each player is able to move to any location within its R_C.

Figure 28 shows an area divided into 5 x 5 logical cells with 22 nodes. A node located in a cell (x, y) can communicate with the nodes in a cell (w, z) where $w = x - 1,\ x,\ x + 1$ and $z = y - 1,\ y,\ y + 1$. For example, in Fig. 28, n_1 and n_7 can communicate. On the other hand, n_1 is not able to communicate with node n_9 or any other node located in cells farther than one Chebyshev distance from cell $(2, 2)$ (e.g., in Fig. 28, n_1 cannot communicate with n_9).

In our model, each individual player asynchronously runs NSEG to make an autonomous decision about its next location to move. Each node is aware of its own location and can determine the relative locations of its neighbors in R_C. This information is used to assess the goodness of its own position.

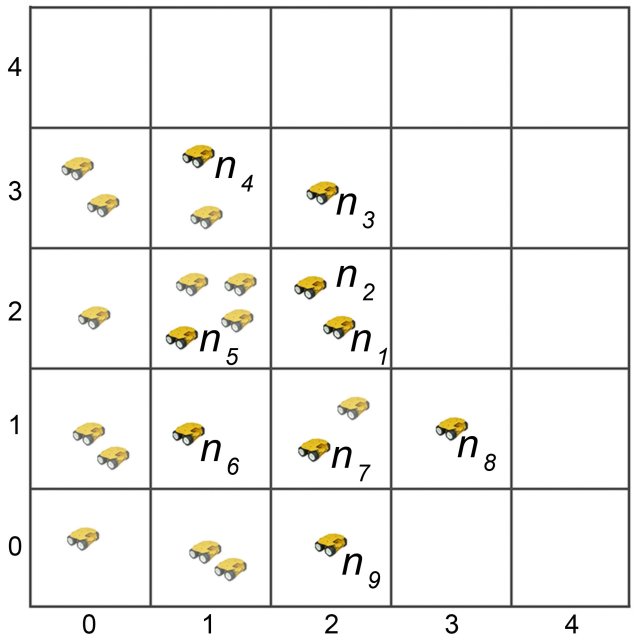


Figure 28: Example of a 5 x 5 logical square lattice populated with 22 nodes. Nodes n_1 and n_7 can communicate, but n_1 cannot establish a wireless link with n_9, as the distance between them is too great.

In NSEG, a set I of m players represents all active nodes in the network. For all $n_i \in I$, a set of strategies $S_i = \{NW, N, NE, W, U, E, SW, S, SE\}$ stands for all possible next cells to which n_i can move. The definitions of NSEG strategies are shown in Table 2.

For example, NW is a new location in the adjacent cell *North-West* of n_i's current location and U is the same *unchanged* location that n_i now inhabits. In Fig. 28, node n_1's strategy s_0 corresponds to a location within cell $(1, 3)$ and s_1 points to a location within cell $(2, 3)$.

We define $f^0_{i,j}$ as a virtual force inflicted on n_i by node n_j located within the

Table 2: Definition of strategies.

Strategy	Location	Movement
s_0	NW	*North-West* of the current location
s_1	N	*North* of the current location
s_2	NE	*North-East* of the current location
s_3	W	*West* of the current location
s_4	U	The same *unchanged* location
s_5	E	*East* of the current location
s_6	SW	*South-West* of the current location
s_7	S	*South* of the current location
s_8	SE	*South-East* of the current location

same cell (e.g., in Fig. 28, a force on node n_1 caused by node n_2). Similarly, $f^1_{i,k}$ is defined as the virtual force inflicted on n_i by node n_k located in a cell one Chebyshev distance away from it (e.g., in Fig. 28, a force inflicted on node n_1 by node n_3). A node n_i is unaware of other agents more than R_C away from it and hence their presence has no effect on node n_i's actions. Let us define $f^0_{i,j}$ as follows

$$f^0_{i,j} = \mathscr{F}_0 \text{ for } 0 < d_{ij} \leq d_{cel} \,, \tag{20}$$

where d_{ij} is the Euclidean distance between n_i and n_j which are in the same logical cell, d_{cel} is the dimension of the logical cell, and $\mathscr{F}_0$ is a large force value between n_i and n_j as defined below.

Now we define the total virtual force on n_i exerted by the neighboring nodes located in the same cell as

$$\sum_{n_j \in D^0_i} f^0_{i,j} = \sum_{n_j \in D^0_i} \mathscr{F}_0 \,, \tag{21}$$

where D_i^0 is a set of all nodes located in the same cell.

Similarly, $f_{i,k}^1$ can be defined as

$$f_{i,k}^1 = \varphi_i(d_{cel} - d_{ik}) \text{ for } d_{cel} < d_{ik} < R_C \ , \tag{22}$$

where d_{ik} is the Euclidean distance between n_i and its neighbor n_k (one Chebyshev distance away), φ_i is the expected node degree which is a function of mean node degree, as presented in Urrea et al. [68], and the total number of neighbors of n_i to obtain the highest area coverage in a given terrain.

Let us now define the total force on n_i exerted by its neighbors one Chebyshev distance away from it as

$$\sum_{n_k \in D_i^1} f_{i,k}^1 = \sum_{n_k \in D_i^1} \varphi_i \left(d_{cel} - d_{ik}\right) \ , \tag{23}$$

where D_i^1 is the set of nodes occupying cells one Chebyshev distance away from n_i's current location.

To encourage the dispersion of nodes, we assign a larger value to the force from those neighbors located in D_i^0 (i.e., F_0 in Eq. (20)) than that of the total force exerted by neighbors in D_i^1 (i.e., $f_{i,k}^1$ from Eq. (22)) as

$$\mathscr{F}_0 > \sum_{n_k \in D_i^1} f_{i,k}^1 \, . \tag{24}$$

In NSEG, player n_i's payoff function $u_i(\boldsymbol{s})$ is defined as the total forces inflicted on n_i by the nodes located in her neighborhood as follows

$$u_i(\boldsymbol{s}) = \begin{cases} \sum_{n_j \in D_i^0} \mathscr{F}_0 + \sum_{n_k \in D_i^1} f_{i,k}^1 & \text{if } D_i^0 \cup D_i^1 \neq \emptyset \, ; \\ \mathscr{F}_{max} & \text{otherwise} \ , \end{cases} \tag{25}$$

where $\mathscr{F}_{max}$ represents a large penalty cost for a disconnected node defined as

$$\mathscr{F}_{max} = |I| \times F_0 \, . \tag{26}$$

The main objective for each node is to minimize the total force inflicted by its neighbors, which implies minimizing the value of the payoff function expressed in Eq. (25).

Now we can introduce our NSEG as a two-step process:

- Evaluation of player's current location
- Spatial game setup.

Let us study each step in detail in the following sections.

5.2 Evaluation of NSEG Player's Current Location

After moving to a new location, n_i computes $u_i(\boldsymbol{s})$ defined in Eq. (25) to quantify the goodness of its current location. Then, it runs FGA to determine a set of possible *good* next locations L_i into which it can move. This is achieved by running FGA over a continuous space in n_i's proximity. Computation of L_i is based only on the local neighborhood information of n_i. Note that n_i can acquire this information by various means (e.g., the use of directional antennas and received signal strength) without requiring any information exchange with its neighbors.

We generate discrete locations from L_i by mapping them into a stochastic vector σ_i with probabilities assigned to each cell into which player n_i can move.

Consequently, n_i's mixed strategy profile is defined as

$$\sigma_i = (\sigma_i(s_0), \sigma_i(s_1), \ldots, \sigma_i(s_8)) \,, \tag{27}$$

where $\sigma_i(s_l)$ represents a probability of strategy l's being played. The mixed strategy profile σ_i reflects n_i's preferences over its next possible locations by assigning positive probability only to these locations that may improve its payoff. Figure 29 shows the probability state transition diagram for a node in state s_4. In this figure, the probability of each transition is assigned by the FGA locally run by this node.

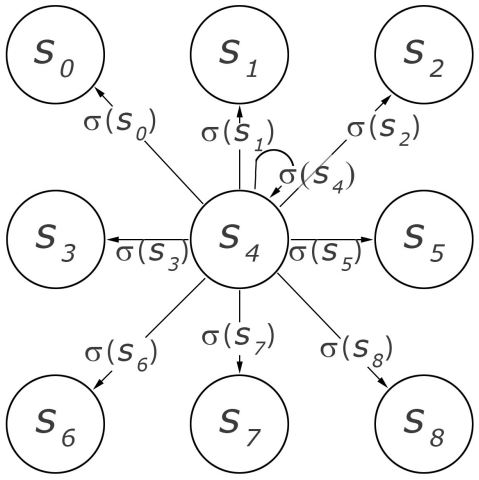


Figure 29: The probability state transition diagram derived from a stochastic vector σ_i.

Player n_i determines if it should move to a new location by evaluating $\sigma_i(s_4)$ as

$$\sigma_i(s_4) > (1 - \epsilon) \,, \tag{28}$$

where ϵ is a small positive number.

If Eq. (28) holds, n_i stays in its current location. Otherwise, it moves to a new location that results in an improvement of its payoff.

In our NSEG, multiple nodes can occupy one logical cell. All nodes located in the same logical cell will generate the same payoff values and similar mixed strategy profiles resulting from running the FGA in the same environment. Therefore, to reduce computational complexity, one player can represent the behavior of all other players located in the same logical cell. Consequently, without loss of generality, instead of referring to u_j and σ_j for player n_j, we will refer to $\bar{u}$ and $\bar{\sigma}$ for each player located in the logical cell in which n_j is located. As a result, the set of each spatial game players $\bar{I} \subset I$ consist of up to nine members, $\bar{u}_j$ reflects the total forces inflicted on n_i's neighboring cell j, and $\bar{\sigma}_j \in \bar{\boldsymbol{\sigma}}$ denotes a stochastic vector with probabilities assigned to each possible location that player(s) occupying cell j may move to at the next step.

5.3 Spatial Game Setup

If player n_i decides to move to a new location using Eq. (28), she gathers $\bar{u}_j$ and $\bar{\sigma}_j$ for all $n_j \in \bar{I}$. Node n_i constructs its payoff matrix M_i with an entry for each possible strategy profile $\boldsymbol{s}$ that can arise among members of $\bar{I}$. Each element of M_i reflects the goodness of n_i's next location over possible combinations of all other players' strategies. After that, n_i computes its expected payoff for this game as

$$u_i(\bar{\boldsymbol{\sigma}}) = \sum_{\boldsymbol{s} \in S} \left(\prod_{n_j \in \bar{I}} \sigma_j(s_j) \right) u_i(\boldsymbol{s}) \, . \tag{29}$$

Expected payoff $u_i(\bar{\boldsymbol{\sigma}})$ is an estimation of what the total forces inflicted on player n_i will be if she plays her mixed strategy profile $\bar{\sigma}_i$ against her opponents' strategy profiles $\bar{\boldsymbol{\sigma}}_{i-1}$. As such, $u_i(\bar{\boldsymbol{\sigma}})$ is an indication of n_i's possible improvement resulting from the mixed strategy profile obtained by FGA.

Our FGA only takes into account the current positions of the neighboring nodes to find the next locations to move. However, our NSEG, combining FGA with game theory, can find even better locations since it uses additional information regarding the payoffs of the neighbors as defined in Eq. (25). We formalize this notion in the lemma below.

Lemma 7. *Player n_i's mixed strategy profile σ_i obtained from* FGA *may not reflect the best new location(s) for player n_i.*

Proof. Let us consider a case where set D_i^1 (Eq. (23)) consists of equally distanced neighbors from n_i. Suppose also that there is a node n_k in the same cell as n_i. Consequently, our FGA will decide that n_i should move into one of its neighboring cells because of n_k. In this setting, our FGA will result in $\sigma_i(s_4) = 0$ (i.e., the probability of staying in the same location is 0). This decision is based on the fact that FGA only takes into account those forces inflicted on a player by her neighbors (Eqs. (21) and (23)).

It is clear that FGA cannot distinguish the optimal choice among possible positions to move within its neighboring cells since the forces applied from each direction are equal by the above assumption. Hence, it is possible that our FGA assigns a probability of 1 to a strategy l (i.e., $\sigma_i(s_l) = 1$) while a better strategy

j exists (requiring to move to cell j) with $u_j(\boldsymbol{s}) < u_l(\boldsymbol{s})$ (Eq. 25). □

Lemma 7 shows that player n_i's mixed strategy profile may not be the most profitable strategy in her proximity. Therefore, player n_i should utilize additional information about her neighbors' payoffs and mixed strategy profiles (Eqs. (25) and (27)) to determine if locations obtained from FGA are indeed the best and what her next location should be. Hence, player n_i sets up a spatial game among herself and all other members of $\bar{I}$ to compute her expected payoff from this interaction using Eq. (29).

Let us consider the neighboring cells for player n_i. Recall that each neighboring cell j will have forces, called $\bar{u}_j$, applied on it by its local neighbors. Let $\mathscr{C}_{min} = \min\{\bar{u}_0, \bar{u}_1, \ldots \bar{u}_8\}$ denote player n_i's neighboring cell such that the forces inflicted on it is the minimum.

To make her movement decision, player n_i evaluates possible improvement reflected in $u_i(\bar{\boldsymbol{\sigma}})$ against $\mathscr{C}_{min}$ using the following equation

$$\mathscr{C}_{min} + \alpha < u_i(\bar{\boldsymbol{\sigma}}) \,, \tag{30}$$

where α represents the value by which the total force on the logical cell $\mathscr{C}_{min}$ would have changed if player n_i have moved there. In this case, if there exists a logical cell $\mathscr{C}_{min}$ in player n_i's neighborhood that guarantees her better improvement than location(s) returned by FGA, she should move into $\mathscr{C}_{min}$.

Therefore, as the direct result of Lemma 7 and Eq. 30, we can state the following corollaries which govern decisions of our NSEG.

Corollary 4. *If the expected improvement for player n_i resulting from moving into a location obtained by* FGA *is worse than moving into $\mathscr{C}_{min}$ (Eq. (30)), player n_i's next position should be $\mathscr{C}_{min}$.*

Corollary 5. *If the expected improvement for player i obtained from* FGA *is better than (or the same as) moving into $\mathscr{C}_{min}$ (Eq. (30)), player i selects her next location according to her mixed strategy profile σ_i.*

5.4 Analysis of NSEG Convergence

In NSEG, a movement decision for node n_i is based on the outcome of the locally run FGA and the spatial game set up among n_i and the nodes in its neighborhood. Each node pursues its own goal of reducing the total force inflicted on it by effectively positioning itself in one of the neighboring cells. However, our ultimate goal is to evolve the entire system towards a uniform node distribution as the result of each individual node's selfish actions. In order to analyze the performance of a system, we define the optimal solution for each node and its effect on the entire node population.

The worst possible state for player n_i is to become isolated from the other nodes, in which case $u_i = \mathscr{F}_{max}$ and player n_i cannot interact with other nodes to improve her payoff. From an entire network perspective, the disconnected node adds little to network performance and can be considered a lost resource. Eq. (25) guarantees that no individual node chooses a new location that will result in its becoming disconnected.

Since an additional node located in the same cell as player n_i (i.e., $D_i^0 = 1$) affects n_i's payoff adversely to a greater degree than that of distant neighbors (i.e., members of D_i^1), player n_i prefers to be the only occupant of its current logical cell. Multiple nodes in a single cell is also undesirable from a network perspective, as area coverage could be improved by *transferring* the additional node into a new empty cell where possible. Therefore, given a large enough terrain, a preferred network topology would have each cell occupied by at most one node without any disconnected nodes, which is precisely the goal of each player in our NSEG.

Let s_i^* be a strategy for a non-isolated player n_i who is the sole occupant of her cell. Let s_{opt}^* be an optimal strategy, representing a permutation of neighbor locations and mixed strategy profiles s_i^*. Suppose, at some point in time, all nodes evolve their positions such that each node plays its own optimal strategy of s_{opt}^*. Then a strategy profile $\boldsymbol{s}^* = (s_1^*, s_2^*, \ldots, s_n^*)$ represents a network topology in which each node is a single occupant in its cell and there are no disconnected nodes.

In our NSEG, the main objective for each node is to minimize the total force inflicted on it, which translates into the goal of minimizing the value of the payoff functions defined in Eqs. (25) and (29).

Let an invading sub-optimal strategy $s_j' \neq s_{opt}^*$ be played by player n_j. Then s_{opt}^* is ESS if the following condition holds

$$u(s_{opt}^*, s_{opt}^*) < u(s_j', s_{opt}^*)\,, \tag{31}$$

where an optimal strategy s^*_{opt} can be played by any $n_i \in I \backslash \{n_j\}$.

The following lemma shows that a strategy s^*_{opt} is evolutionary stable and hence no other strategy can invade a population playing $\boldsymbol{s}^*$.

Lemma 8. *A strategy s^*_{opt} is evolutionary stable.*

Proof. There are two cases in which player $n - j$'s strategy s'_j may differ from s^*_{opt}. In one, strategy s'_j represents a case where player j is disconnected and, as stated in Eq. (25), receives payoff $\mathscr{F}_{max}$, which is strictly greater than any possible $u(s^*_{opt}, s^*_{opt})$. If, on the other hand, strategy s'_j stands for player j's location in the cell already occupied by some other node, then, according to Eq. (24), $u(s^*_{opt}, s^*_{opt}) < u(s'_j, s^*_{opt})$. Consequently, in both cases in which strategy $s'_j \neq s^*_{opt}$ invades a population playing strategy s^*_{opt} (i.e., a population playing a mixed strategy profile $\boldsymbol{s}^*$), the first condition of ESS (Eq. (31)) holds, establishing that s^*_{opt} is an ESS. □

Lemma 8 shows that when entire population plays the strategy in which each individual node is a single occupant of its cell and is connected to at least one other node, no other strategy can successfully invade this topology configuration. We can generalize the results of Lemma 8 in the following corollary.

Corollary 6. *A strategy $\boldsymbol{s}^*$ represents a stable network topology that will maintain its stability since no node has any incentive to change its current position.*

5.5 NSEG Experimental Results

We implemented NSEG using Java programming language. For each simulation experiment, the area of deployment was set to 100×100 unit squares. Initially, the nodes were placed in the lower-left corner of the deployment area, and have no knowledge of the underlining terrain and neighbors' locations. Each simulation experiment was repeated 10 to 15 times and the results were averaged to reduce the noise in the observations.

The snapshot in Fig. 30 shows a typical initial node distribution before NSEG is run autonomously by each node. The total deployment area is divided into 10×10 logical cells (each 10×10 unit squares). The four cells located in the lower-left corner of the deployment terrain are occupied by a population of 80 autonomous mobile agents. This type of initial deployment simulates a realistic scenario in which mobile nodes enter the geographical area from one common access point, as anticipated in many rescue or reconnaissance missions. The shaded area around the nodes indicates the portion of the terrain cumulatively covered by the communication ranges of the nodes.

The snapshot of the node positions after running NSEG 10 steps is shown in Fig. 31. We can observe that even in the early stages of the experiment, the nodes are able to disperse far from their original locations and provide significant improvement of the area coverage while keeping network connected. However, since it is very early in the experiment, there is still notable node concentration in the area of initial deployment.

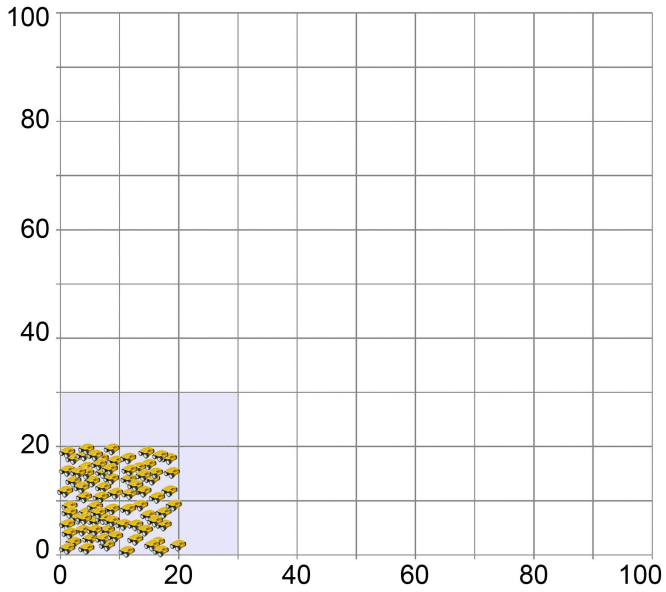


Figure 30: An initial deployment of 80 nodes in a 100 $\times$ 100 unit square area.

A stable node distribution after running NSEG for 60 time units is shown in Fig. 32. At this time, no cell is occupied by more than one node and the entire terrain is covered by the nodes' communication ranges. The snapshot in Fig. 32 represents the stable state for this population. As presented in Lemma 8 and Corollary 6, after this stable topology is reached, no node has an incentive to change its location in the future. After step 60, this stable network topology for this example remains unchanged in all consecutive iterations of our NSEG, which verifies the conclusions of Lemma 8 and Corollary 6.

Figure 33 shows the improvement of NAC and total number of cells that are occupied at each step of the simulation as NSEG progresses. We can observe that the entire area becomes covered by mobile nodes' communication areas

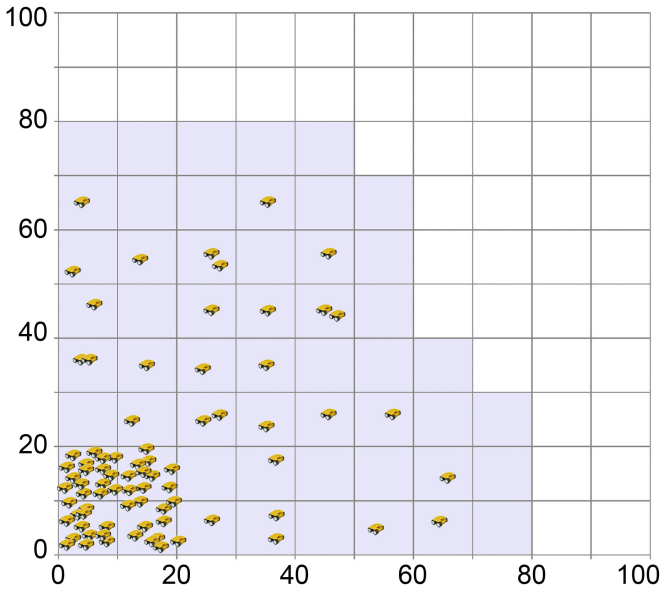


Figure 31: Network distribution obtained by 80 autonomous nodes running NSEG for 10 steps.

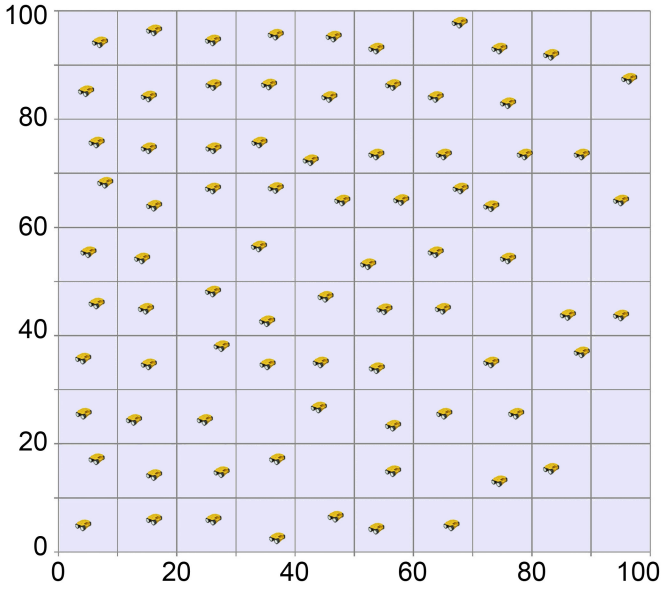


Figure 32: Stable node distribution obtained by 80 autonomous nodes after running NSEG for 60 steps.

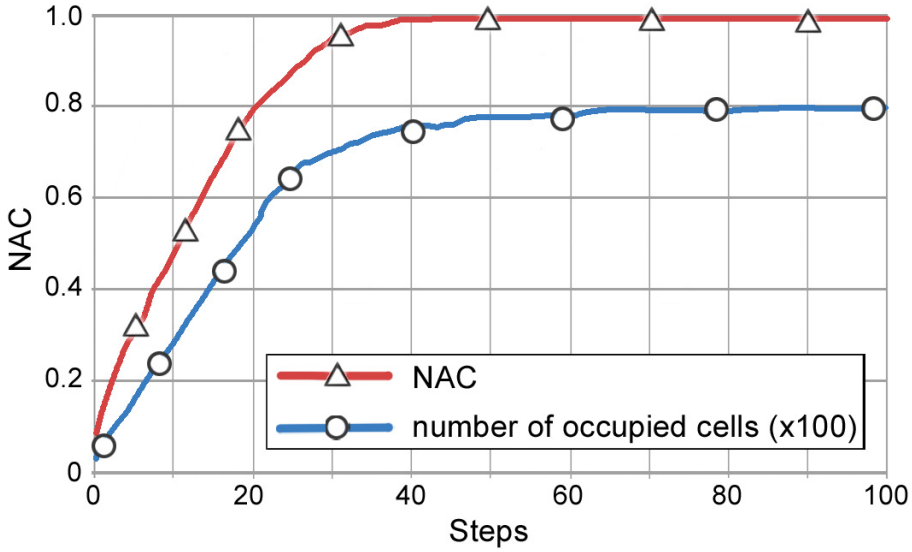


Figure 33: NAC and the number of occupied logical cells obtained by 80 autonomous nodes running NSEG.

(i.e., NAC = 1) after approximately 40 iterations of NSEG. However, the number of occupied cells increases for another 20 steps to a point where each cell becomes occupied by at most one node. We can derive two conclusions from this observation: *(a)* for the deployment area of 100×100 unit square area divided into 10×10 logical cells, 80 nodes are sufficient to achieve NAC = 1, and *(b)* even when the goal of total area coverage is achieved, the network topology does not stabilize until the optimal strategy profile $\boldsymbol{s}^*$ is realized by the entire network nodes.

Figure 34 shows the improvement in NAC for networks with different number of mobile nodes. In Fig. 34, for a larger number of nodes, the network requires more time to achieve its maximal area coverage since there are more nodes to disperse in larger networks. However, maximal NAC achieved by NSEG

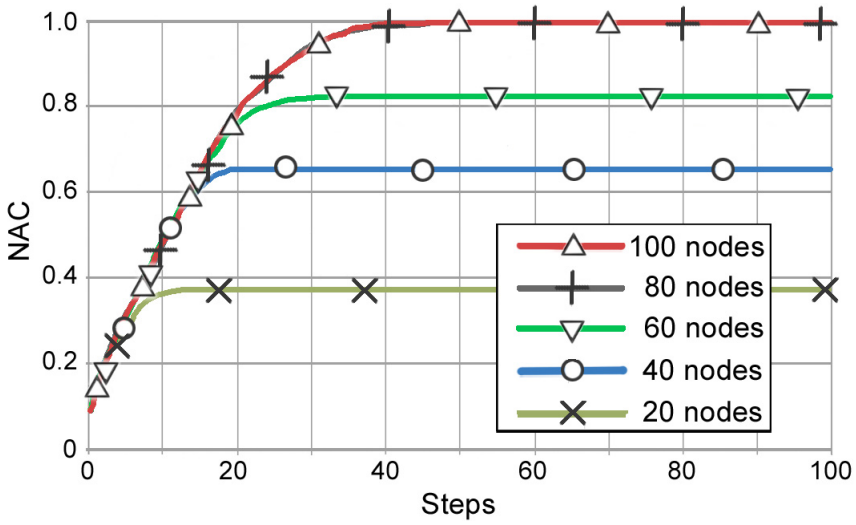


Figure 34: Improvement of NAC by NSEG in different network sizes ($n = 20$ to 100).

increases notably as the number of nodes deployed in the same geographical area increases. It can also be seen in Fig. 34 that the rate at which networks increase their NACs is independent of the number of nodes (up to the point where the maximum coverage areas of relative populations are reached). In Fig. 34, it is clear that a network with 60 nodes is not sufficient to cover the entire area, whereas a 100-node network does not further improve NAC compared to an 80-node network. This observation justifies our network size selection for the experiment shown in Figs. 30–33.

5.6 Observations

NSEG combines FGA, traditional GT, and EGT. In NSEG, each node pursues its own selfish goal of reducing the total virtual force inflicted on it by effectively

positioning itself in one of the neighboring cells. Nevertheless, each node's selfish actions lead the entire system towards a uniform and stable node distribution. We showed a formal analysis of our NSEG and prove that ESS is its convergence point. Our simulation results demonstrate that NSEG performs well with respect to network area coverage, uniform distribution of mobile nodes, and convergence speed.

6 Node-spreading Bio-inspired Game

A novel mechanism for autonomous MANET nodes dispersal over the unknown geographical territory called BioGame is presented in this chapter. A preliminary version of BioGame was introduced in [32, 40] and is presented here with its extended formal description and comprehensive performance analyses.

BioGame combines FGA and GT to achieve a uniform distribution of mobile nodes in the area of deployment. In BioGame each individual player makes an autonomous decision about its next location to move. A node wishing to change its position first runs our FGA to determine a set of favored new locations and assigns a stochastic vector of probabilities to each element of this set. The stochastic vector of probabilities reflects preferences over the next possible locations for the moving node. Using FGA in this step significantly decreases the search cost for finding the preferred next position since it reduces the computational space for the spatial game which will be run next. Our FGA takes into account only positions of neighboring nodes to find a set of plausible next locations, which is significantly smaller than the initial search space, allowing our spatial game to perform more refined calculations. In order to determine the best choice for the next position, a moving node computes expected payoffs from a spatial game set up among itself and its neighbors that takes into account the preferred next positions of neighbors and goodness of their current locations. This step replaces a *roulette wheel* or *elitism* used in classic genetic algorithms to make a decision for a node movement. In BioGame, our spatial game effec-

tively and efficiently utilizes information about the neighbors in order to enhance FGA movement decisions. Because BioGame is partially based on game theory, we will refer to a node as a player or a mobile agent, interchangeably, and its locations, represented by a chromosomes, will be often identified as strategies. Following various of the publications in the area of GT, we refer to a player in the feminine form without any clandestine intention.

We formally introduce BioGame and present proofs of its basic properties in Sect. 6.1. Section 6.2 demonstrates that our BioGame performs well with respect to network area coverage, uniform distribution of mobile nodes, and convergence speed and shows that it outperforms RW and approaches running FGA in terms of both convergence speed and area coverage.

6.1 Introduction of BioGame

In this section, we introduce our BioGame model and explain in details how FGA and our spatial game are merged to achieve a uniform distribution of mobile nodes with maximized total area coverage over an unknown geographical terrain without requiring global network information or a synchronization among the nodes.

In BioGame, a set I of m players represents all active nodes in the network and for all $n_i \in I$, a set of strategies S_i stands for possible locations into which player n_i can move. The deployment terrain in our model is represented as a two-dimensional hexagonal lattice, which closely emulates the idealized circular

propagation pattern of an antenna and can provide a compact coverage of the underlining terrain. We identify each logical cell by its coordinate (x, y) in the hexagonal grid. The distance $\Delta_{i,j}$ between two nodes n_i and n_j located in logical cells (x_i, y_i) and (x_j, y_j), respectively, is defined as

$$\Delta_{i,j} = \begin{cases} max(|\Delta_x|, |\Delta_y|) & \text{if } \textit{signs} \text{ of } \Delta_x \text{ and } \Delta_y \text{ are the same ;} \\ |\Delta_x| + |\Delta_y| & \text{otherwise ,} \end{cases} \tag{32}$$

where $\Delta_x = (x_j - x_i)$, $\Delta_y = (y_j - y_i)$, and vertical bars denote the absolute value of enclosed expression.

We define $R_C \in \mathbb{Z}^+ \cup \{0\}$ as the communication range of a node and let the circle with radius R_C circumscribe the area covered by it. As a result, in a hexagonal grid nodes n_i and n_j can establish a wireless link if $\Delta_{i,j} \leq R_C$ and the coverage area of node n_i extends over all all logical cells within R_C distance from it. For simplicity and without loss of generality, our implementation considers a monomorphic population where each node has the same R_C and the center-to-center distance between two adjacent cells equals one.

Figure 35 shows an example of hexagonal lattice with 68 logical cells and five mobile nodes. If we let $R_C = 3$, in Fig. 35, node n_1 can communicate with nodes n_2, n_3, and n_4 but cannot establish a wireless connection with node n_5 nor has any knowledge of its existence.

We define N_i as a set of neighbors for player n_i that are positioned somewhere in its coverage area (i.e., $N_i = \{\forall_{n_j \in I \setminus \{n_i\}} : \Delta_{i,j} \leq R_C\}$) and assume that

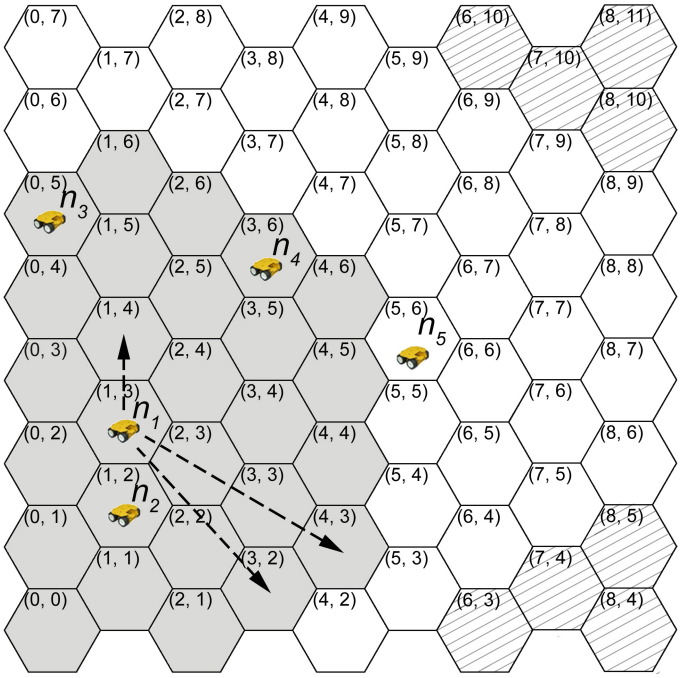


Figure 35: Five nodes distributed within a hexagonal lattice of 68 logical cells. Shaded region represents the coverage area of node n_1 with three arrows pointing to its preferred new locations. Slanted lines mark the region that is not covered by communication ranges of any of the nodes in this figure.

each node can determine the relative locations of all neighbors in its communication range. Note that a player may discover positions of her neighbors without requiring exchange of information by employing one of the existing methods reported in the literature. For example, a mobile agent can use directional antennas (see P. Rong et al. [53] or S. Winfree [73]) and received the signal strength indicator (see X. Li et al. [43]) for estimating locations of its neighbors. Since set N_i contains all nodes with which n_i can establish wireless links, we refer to $|N_i|$ as the *degree* of node n_i.

In our BioGame, each node autonomously determines a new position within R_C distance from its current coordinate. The shaded region in Fig. 35 denotes the coverage area of node n_1 and all the possible cells into which it can move in one step. Figure 35 also depicts the area that is not covered by the communication ranges of any node in this example by slanted lines.

The following sections present our BioGame in detail. In Sect. 6.1.1, a node that can improve its spatial location determines a set of next preferred positions using FGA. Section 6.1.2 explains a spatial game played between a moving node and its neighbors to determine the best new location to move. We describe implementation of our BioGame and analyze its basic properties in Sect. 6.1.3.

6.1.1 Finding Next Preferred Locations Using FGA

In our model, each individual player asynchronously runs BioGame to make an autonomous decision about its next location. When node n_i plans to change its position, it first carries out FGA computation over a continuous space in its proximity to find a set of preferred new locations and a stochastic representation for each element of this set that reflects its preferences over these future locations.

The fitness function used by our FGA is based on the virtual forces envisioned to be inflicted on a mobile node by its neighbors. The virtual force $F_{i,j}$ exerted on node n_i by node $n_j \in N_i$ is calculated according to the following equation

$$F_{ij} = \begin{cases} \gamma_i \left(R_c - d_{ij}\right) & \text{if } 0 < d_{ij} < d_{th} \ ; \\ \epsilon & \text{if } d_{th} \leq d_{ij} \leq R_C \ , \end{cases} \tag{33}$$

where d_{ij} is the distance between mobile nodes n_i and n_j, d_{th} is the threshold value to define the best separation among nodes, ϵ is a positive number ($\epsilon < R_c - d_{th}$), and γ_i is the force scaling factor to promote an optimal node degree.

The scaling factor γ_i is a function of desired node degree $\mu \in \mathbb{Z}^+$ and its existing degree $|N_i|$. We define γ_i as

$$\gamma_i = \frac{(|N_i| - \mu)^2 + 1}{|N_i|} \,. \tag{34}$$

A small value of μ promotes sparsely connected network topologies in which the nodes have limited number of neighbors and reduced overlapping communication areas. On the other hand, a large value of μ supports densely packed networks in which each agent has multiple neighbors and redundant communication channels among nodes can be easily established. Urrea et al. [68] proposed μ as a mean node degree determined by the total number of nodes and deployment area. For our simulation experiments, we selected $\mu = 6$ to optimally utilize resources and to achieve resilient and uniform node distribution in a final network topology. This value of μ provides the highest possible node degree in which each agent may attain R_C distance to all of its neighbors.

The fitness $f_i(\boldsymbol{s})$ of node n_i located in $s_i \in S_i$ exerted by all its neighboring nodes, which positions are conveyed by the deleted strategy profile $\boldsymbol{s}_{-i}$, is calculated as:

$$f_i(s_i, \boldsymbol{s}_{-i}) = \begin{cases} \sum_{n_j \in N_i} F_{ij} & \text{if } N_i \neq \emptyset \,; \\ \mathscr{F}_{max} & \text{otherwise} \,, \end{cases} \tag{35}$$

where $\mathscr{F}_{max}$ represents a large penalty for a disconnected node defined as:

$$\mathscr{F}_{max} > m \times R_C \,. \tag{36}$$

Figure 36 shows the surface plot of the fitness function defined by Eq. (35) for various degrees of node $n_i \in I$ and averaged distances to its neighbors in range $(0, R_c]$. We can observe in Fig. 36 that fitness for node n_i improves when

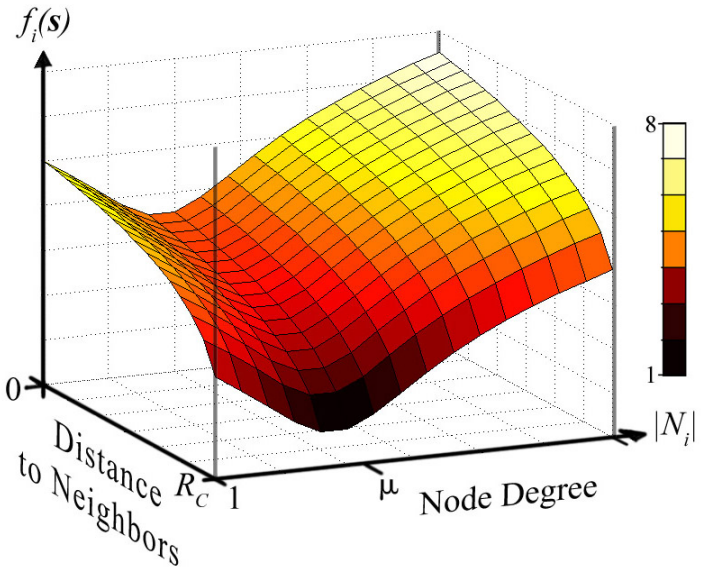


Figure 36: Fitness landscape of our FGA. For clarity of presentation, this figure does not depict the case where $N_i = \emptyset$, which results in $f_i(\boldsymbol{s}) = \mathscr{F}_{max}$ that is much greater than any $f_i(\boldsymbol{s})$ depicted in this graph (see Eq.(36)).

its number of neighbors approaches μ and the distance to them approaches to R_C. At any stage of the game, a player may not have the entire landscape of possible solutions to choose from, as neighbor positions may restrict it, but even a local improvement shifts the node closer to a position with the minimal force

inflicted on it.

In our FGA, if mobile node n_i intends to move, it creates an initial population of randomly generated chromosomes. Each chromosome encodes possible location that is not further than R_C from the current position of n_i. We refer to P_g as the population P at generation g. First, elements of the initial population P_1 are paired to perform selection and single-point crossover operators. The offspring are evaluated using Eq. (35) and added to a pool of candidate solutions for next population P_2. At this stage, the pool of candidate solutions contains both offspring and all elements of P_1. The best performing individuals from the pool of candidate solutions are then selected into the newly created population P_2. To prevent FGA from getting stuck at a locally optimal point, individuals from population P_2 have a small probability of being mutated. In our implementation of FGA, node n_i will repeat this process for a predetermined number of generations or until a set of suitable new locations is found.

Let P_g denote the last population attained by FGA. We generate set of discrete locations from the chromosomes in P_g by mapping their encoded Cartesian plane coordinates into a corresponding logical cells and refer to these locations as L_i. Node n_i forms a set $\bar{S}_i$ of k-preferred new positions by selecting all distinct elements from L_i. Consequently, the number of elements in set $\bar{S}_i$ is weakly smaller than in L_i (i.e., $|\bar{S}_i| \leq |L_i|$) because $\bar{S}_i$ does not contain any duplicate entries. Set $\bar{S}_i$ is described by vector $\langle s_1, s_2, \ldots, s_k \rangle$ which is associated with the stochastic representation of $\sigma_i = \langle \sigma_i(s_1), \sigma_i(s_2), \ldots, \sigma_i(s_k) \rangle$ that assigns a probability to each element of $\bar{S}_i$. An unempty set of preferred new locations is a

subset of all possible next locations for node n_i (i.e., $\bar{S}_i \subset S_i$). A probability for strategy $s_i \in \bar{S}_i$ is denoted by the corresponding element of σ_i, namely $\sigma_i(s_i)$. A likelihood that node n_i will move to location s_i is a result of this location proportional in L_i as well as its fitness. Let us define $w(s_i) = f_i(\hat{s}_i, \boldsymbol{s}_{-i}) \div f_i(s_i, \boldsymbol{s}_{-i})$ and $\mathscr{S}_i = \sum_{s_i \in \bar{S}_i} w(s_i)$, where $\hat{s}_i \in \bar{S}_i$ is a strategy yielding the worst (i.e., the highest) forces inflicted on node n_i. We compute $\sigma_i(s_i)$ as

$$\sigma_i(s_i) = \frac{1}{2}\left(\frac{s_i^n}{|L_i|} + \frac{w(s_i)}{\mathscr{S}_i}\right), \tag{37}$$

where s_i^n denotes a number of times strategy s_i appears in L_i.

For example, suppose that node n_1 runs FGA and obtains possible locations as indicated by the dashed arrows in Fig. 35. After evaluating Eq. (37) for each of these positions, n_1 moves to cell $(1, 4)$ with probability of 0.2, to cell $(3, 2)$ with probability of 0.4, and to cell $(4, 2)$ with remaining probability of 0.4. Consequently, the set of next locations $\bar{S}_i$ for node n_1 is represented by $\langle(1, 4), (3, 2), (4, 3)\rangle$ and its associated stochastic representation $\sigma_i = \langle 0.2, 0.4, 0.4\rangle$. For this example, the resulting probability state transition diagram reflecting the preferences of node n_i over its set of strategies $\bar{S}_i$ is given in Fig. 37.

Each location $s_i \in \bar{S}_i$ would result in force $f_i(s_i, \boldsymbol{s}_{-i})$ (see Eq. (35)) inflicted on node n_i by its neighbors placed according to their strategies expressed by $\boldsymbol{s}_{-i}$. Because $\boldsymbol{s}_{-i}$ represents locations of neighbors in the area around node at the

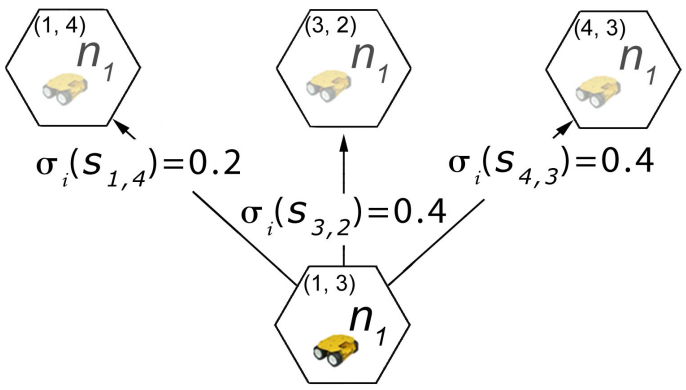


Figure 37: The probability state transition diagram for node n_1 with its set of preferred new locations $\bar{S}_i = \langle (1,4), (3,2), (4,3) \rangle$ and associated stochastic vector $\sigma_i = \langle 0.2, 0.4, 0.4 \rangle$. Node n_1 moves to a new location s_i with the probability $\sigma_i(s_i)$.

time when FGA finds new proffered coordinates, n_i may not distinguish between strategies that are in fact different in their qualities. For example, $s_i, s_i' \in \bar{S}_i$ such that $f_i(s_i, \boldsymbol{s}_{-i}) = f_i(s_i', \boldsymbol{s}_{-i})$) where one location may be more profitable to move to than the other. The theorem below formalizes this notion.

Theorem 4. *Preferences of player n_i over the elements of $\bar{S}_i$ may not reflect a superiority of one strategy over the other, even though such advantage may exist.*

Proof (sketch). Let us consider two strategies $s_i, s_i' \in \bar{S}_i$ at the opposite spectrum of possible new locations for player n_i, so that neighborhoods of s_i and s_i' are independent but both resulting in the same fitness (i.e., $f_i(s_i, \boldsymbol{s}_{-i}) = f_i(s_i', \boldsymbol{s}_{-i})$). Clearly, locations s_i and s_i' appear equally good for player n_i at the time of running its instance of FGA. If we suppose that player n_i is the only moving node at this time and there are nodes $n_j, n_j' \in N_i$ placed in the proxim-

ity of locations s_i and s'_i, respectively, with n'_j having more neighbors than n_j (i.e., $f_j(s_j, \boldsymbol{s}_{-j}) < f_{j'}(s_{j'}, \boldsymbol{s}_{-j'})$), then moving to the location s_i will result in a greater reduction of the total forces imposed on u_i. However, this distinction cannot be recognized by FGA, as it does not consider the total forces applied on its neighbors. □

Each node can distinguish the best location to move by considering additional information about its neighbors. The following lemmas establish advantages for moving player n_i resulting from incorporating information about $f_j(\boldsymbol{s})$ and σ_j for all $n_j \in N_i$.

Lemma 9. *A moving player n_i can improve her decision into which location to move by considering the fitnesses of her neighbors (i.e., $f_j(\boldsymbol{s})$ for all $n_j \in N_i$).*

Proof (sketch). It is a direct consequence of Theorem 4 that player n_i can consider the fitnesses of her neighbors to distinguish locations with smaller concentrations of nodes and thus make a better decision into which position to move. □

Lemma 10. *A moving player n_i can improve her decision about her future position by considering her neighbors' preferences over their next locations (i.e., σ_j for all $n_j \in N_i$).*

Proof (sketch). Let us suppose that there are two strategies $s_i, s'_i \in \bar{S}_i$ such that $f_i(s_i, \boldsymbol{s}_{-i}) = f_i(s'_i, \boldsymbol{s}_{-i}) + \varepsilon$ for a small positive ε resulting in location s_i being better than s'_i. As a result, the FGA run by player u_i will indicate her preferences

for location s_i over s_i'. However, if there there is at least one node $n_j \in N_i$ for which the location indicated by s_i is also the strategy of choice, at the next step nodes n_i and n_j may end up in the same cell, which invalidates the goodness of location s_i for player n_i.

Node n_i can predict future distribution of nodes in its proximity from the possible actions of its neighbors. This information can be used by player n_i to avoid future coordinates that are targeted by nodes in her neighborhood. Consequently, by considering $\sigma_j \ \forall_{n_j \in N_i}$, node n_i can choose a better next position. $\square$

Our FGA run by node n_i finds the best k positions by considering all the locations in a continuous space around it. Although for each player the initial search space of possible new locations is infinite, the set of next k positions computed by FGA is discrete and typically contains only a few elements. Consequently, a moving node can perform more refined calculations to determine its best next position. Player n_i improves FGA performance by setting up a spatial game Γ_i among herself and all nodes in N_i. This step allows u_i to utilize additional information about her neighbors and provides an effective way to choose the best $s_i \in \bar{S}_i$.

6.1.2 Spatial Game $\Gamma_{\mathbf{b}}$

Player n_i assesses the goodness of each $s_i \in \bar{S}_i$ by computing an expected payoff from the spatial game $\Gamma_{\mathrm{b},i}$ set up among herself and opponents in set N_i. We refer to game $\Gamma_{\mathrm{b},i}$ as a *spatial* game since it is limited only to players $(\{n_i\} \cup$

$N_i) \subset I$ and player n_i considers only her personal outcome. Player n_i performs necessary calculations using $\bar{S}_j$ and σ_j for all $n_j \in N_i$. This information can be solicited each time a player moves to a new position, or gathered periodically if she became inactive for a period of time due to achieving an optimal position after her last move.

Let us define $\bar{S}_{-i}$ as a space of possible next positions for all $n_j \in N_i$ (i.e., $\bar{S}_{-i} = \bigtimes_{\forall_{n_j \in N_i}} \bar{S}_j$) and let $\boldsymbol{\sigma}_{-i}$ be a deleted mixed strategy profile that reflects movement preferences for all $n_j \in N_i$. We compute the expected payoff for each $s_i \in \bar{S}_i$ as

$$u_i(s_i, \boldsymbol{\sigma}_{-i}) = \sum_{\boldsymbol{s}_{-i} \in \bar{S}_{-i}} \left(\prod_{n_j \in N_i} \sigma_j(s_j) \right) f_i(s_i, \boldsymbol{s}_{-i}) \,. \tag{38}$$

Player n_i finds the best location s_i^* to move by evaluating all elements of $\bar{S}_i$ using Eq. (38) and selecting one that minimizes possible forces inflicted on it, as stated below.

$$s_i^* \in \underset{s_i \in \bar{S}_i}{\operatorname{argmin}} \; u_i(s_i, \boldsymbol{\sigma}_{-i}) \,. \tag{39}$$

This step replaces the stochastic roulette wheel or deterministic elitism selection mechanisms in making a final decision for the new position of node n_i. However, contrary to *elitism* and the *roulette wheel*, $\Gamma_{\mathrm{b},i}$ utilizes additional information about the neighbors in order to enhance FGA performance, as rationalized by Theorem 4 and Lemma 10. We validate the advantages of BioGame over FGA in the following theorem.

Theorem 5. *In BioGame, the decision based on spatial game $\Gamma_{b,i}$ to determine the next position for player n_i provides the same or better results than a position based only on the outcomes of* FGA.

Proof (sketch). Let us first assume that player n_i is the only node intending to change its location at this time instance and, consequently, no information about eventual actions of the players in N_i provide additional information for n_i. Since $\bar{S}_{-i}$ is a singleton and $\forall_{n_j \in N_i}\ \sigma_i(s_x) = 1$, where s_x represents present location of player n_j, Eq. (38) becomes equivalent to Eq. (35) and, hence, player n_i selects the best new place for her to move, as ensured by Eq. (39).

If, on the other hand, there is at least one other player $n_j \in N_i$ at this time intending to move according to her σ_j, this information can improve u_i selection process. Let $\hat{s}_i$ be the best strategy that FGA can find (either as an outcome of elitism or roulette wheel process), then the expected payoff resulting from moving into $\hat{s}_i$ evaluated by Eq. (38) can be at most as good as the result of Eq. (39) evaluated by our BioGame. Therefore, $u_i(\hat{s}_i, \boldsymbol{\sigma}_{-i}) \geq u_i(s_i^*, \boldsymbol{\sigma}_{-i})$ must hold.

As a result, player n_i can find the next best location to move by evaluating her future positions with respect to possible movements of all $n_j \in N_i$ through evaluating a spatial game $\Gamma_{\mathrm{b},i}$. □

Theorem 5 states that player u_i can improve its performance by executing spatial game $\Gamma_{\mathrm{b},i}$ to determine the best next position to move and speed up the network convergence time. This observation has been further validated by the

results of our simulation experiments presented in Sect. 6.2.

6.1.3 Implementation of our BioGame

In BioGame, the goal for each node is to place itself over an unknown geographical terrain to maximize the area covered by all nodes and to achieve a uniform node distribution while the overall distance traveled by each node is minimized and the network remains connected. Each player independently runs its own copy of BioGame to determine the best next location to move. Our BioGame implementation is given in Algorithm 2.

Algorithm 2 Implementation of BioGame at each $u_i \in I$

```
loop
  if f_i(s) > threshold then
    Run FGA
    Run Γ_{b,i}
    Move to a location determined by Eq. (39)
  else
    Do nothing
  end if
end loop
```

As can be seen in Algorithm 2, a node examines its fitness to monitor possible changes around it. This step allows a mobile agent to detect changes in its proximity, which could be a result of movement, redeployment, or deactivation of other nodes. Hence, the node can respond in real time to the changes in the surrounding area by repositioning itself according to the new environment.

In BioGame, each node can move into any location within R_C distance from

its current coordinates as long as the new position is not obstructed or outside the area of deployment. However, in order to preclude an autonomous mobile agent from considering positions which may result in it being disconnected and to balance network convergence speed with an accuracy of selected coordinates, we limit the FGA search space to the area circumscribed by a circle with radius $R_m \leq R_C$ define as

$$R_m = R_C - d_{i,min} \,, \tag{40}$$

where $d_{i,min}$ is the distance to the nearest neighbor of node n_i. This constraint enable nodes to spread faster shortly after initial deployment, when the concentration of mobile agents in one area is high, and to perform a more detailed search for new locations when a network topology approaches an optimal distribution.

Gundry et al. showed that FGA converges to a stationary distribution with high aggregate fitness [16]. We restate here their theorem describing this fact

Theorem 6. ***(from [16])*** *If the transition matrix Q for a Markov chain of our* FGA *topology control algorithm is ergodic, then Q will converge to a stationary distribution.*

Proof. See Gundry et al. [16] for the proof. □

Since our FGA converges (see Theorem 6) and BioGame performs at least as well as FGA (see Theorem 5), we can state the following corollary for the convergence of our BioGame

Corollary 7. *A* MANET *with a stable number of autonomous mobile nodes, each running BioGame, converges to a stationary distribution where no node can further improve its position.*

6.2 BioGame Experimental Results

We have built a modeling platform for our BioGame using multi-agent simulation of networks (MASON), which is a discrete-event simulation library designed by George Mason University [44]. Our software implementation consists of over 3,000 lines of algorithmic Java code and provides a graphical user interface allowing for a real-time visualization of ongoing network dynamics throughout an experiment. While a simulation experiment is running, we are able to collect all crucial data needed for further analysis of BioGame performance. Our software has the ability to run simulations using the same initial deployment conditions for all related cases to generate comparable results.

For each experiment, the area of deployment is set to 100×100 units and the nodes are initially placed in its upper-left corner. Deployed autonomous mobile agents had no *a priori* knowledge of the underlining area and locations of their neighbors. Our initial distribution imitates a realistic situation where the nodes enter a terrain from a common point (e.g., initiating nodes into a post-earthquake zone or a territory occupied by hostile forces) compared to random or other initial allocation schemes often see in the literature. Each of our experiments was repeated 20 times and the results were averaged to reduce the noise in the

collected data.

Figure 38 shows a typical distribution of a network comprising 40 nodes before each autonomous MANET agent runs its copy of BioGame for the first time.

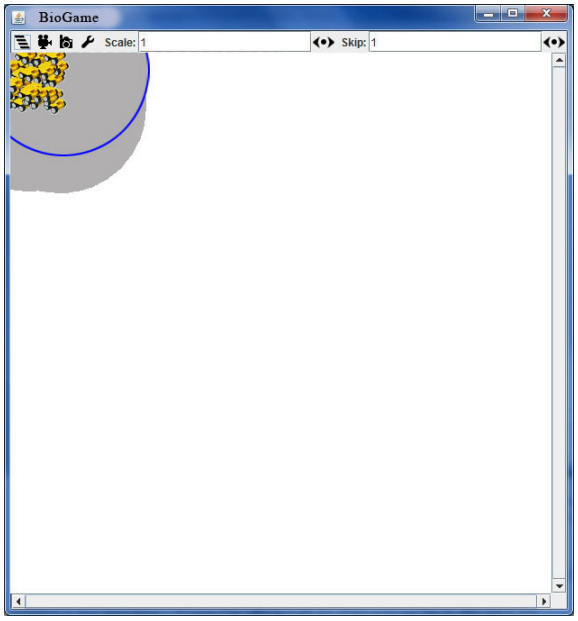


Figure 38: An initial deployment of 40 mobile agents with $R_C = 16$. The total terrain covered by the communication areas of all nodes is colored gray and only the communication range of one node is outlined for simplicity.

In Fig. 38, all nodes are crammed into the 10×10 upper-left corner of the deployment terrain and each node has the area coverage defined by $R_C = 16$. In all snapshots depicting simulated network topologies in this section, shaded area indicates the portion of the terrain cumulatively covered by communication ranges of the nodes. To visualize the communication range of a single node, we outlined in Fig. 38 the coverage area of only one mobile agent. Also, for clarity

of presentation, we do not delineate the hexagonal logical grid in the figures.

A typically stable distribution of 40 nodes with $R_C = 16$ after running BioGame for 100 steps is shown in Fig. 39.

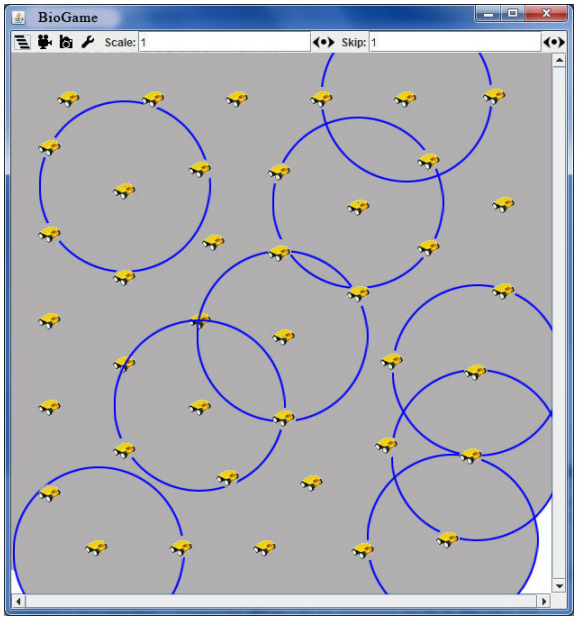


Figure 39: A typical final node distribution for population of 40 nodes with $R_C = 16$. To better visualize our BioGame, we outline only the communication areas of selected nodes.

We observe in Fig. 39 that the nodes can achieve nearly uniform distribution while maintaining network connectivity. In Fig. 39, the distance among all neighboring nodes is close to R_C and almost the entire terrain is covered by the communication ranges of the mobile agents.

In the following sections, we evaluate the performance of BioGame with respect to various criteria. In Sect. 6.2.1, we compare NAC in networks where mobile agents selecting their next locations by means of BioGame, FGA, and

RW. Sections 6.2.2 and 6.2.3 compare BioGame and FGA with respect to ADT and uniformity measures $\mathscr{U}_A$ and $\mathscr{U}_C$, respectively. Finally, Sect. 6.2.4 analyzes BioGame resilience to the reduction of number of nodes during the experiments.

6.2.1 The Network Area Coverage

In order to evaluate the performance of BioGame, we compare it to the network in which autonomous mobile agents determine their future locations using FGA. To establish a basis for our comparisons, we also include the results for nodes moving to the next positions by means of RW. In the RW approach, each node can move into any location constrained by Eq. (40) with equal probability while nodes running FGA use an elitism selection mechanism to choose the best position. In order to meaningfully evaluate BioGame, FGA, and RW based approaches, we gauge their performances using the network of 40 mobile agents that were initially placed in the same starting locations.

Figure 40 shows the improvement of NAC for networks where nodes are running Bio-Game, FGA, and RW methods for 100 steps. We note in Fig. 40 that BioGame combining FGA and GT performs better than the network in which nodes are choosing their next locations by means of FGA or RW. We can also observe that after the initial few steps, BioGame converges faster than FGA alone and that the approach where the nodes chose their future locations randomly (i.e., using RW) results in inferior performance when compared to BioGame and FGA base schemes. Also, it can be seen in Fig. 40 that in the early stages of

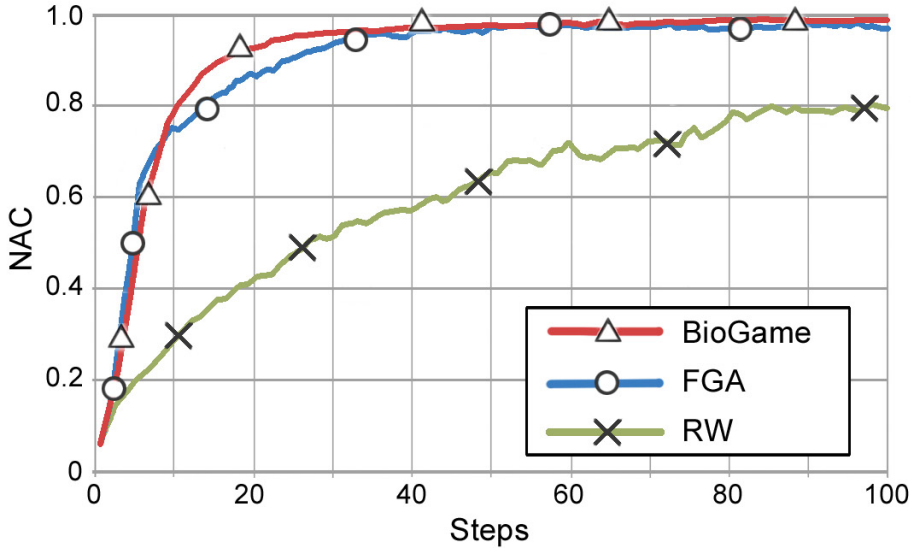


Figure 40: NACs obtained by 40 autonomous nodes with $R_C = 16$ running BioGame (denoted by the line marked △), FGA (denoted by the line marked ◯), and RW (denoted by the line marked ×) methods.

the experiments, the NAC for BioGame and FGA have the highest improvement rate, indicating that nodes are able to disperse far from their initial locations, especially at the beginning of simulations and demonstrating the effectiveness of FGA in finding new locations.

6.2.2 The Average Distance Traveled

Figure 41 presents the comparison of ADT by nodes running BioGame and FGA in a network consisting of 40 mobile agents. In Fig. 41, ADT is displayed as the aggregated value for a node movement. As we observe in Fig. 40, the area covered by mobile agents running FGA approaches the area covered by BioGame network at step 35. However, we can see in Fig. 41 that the average distance

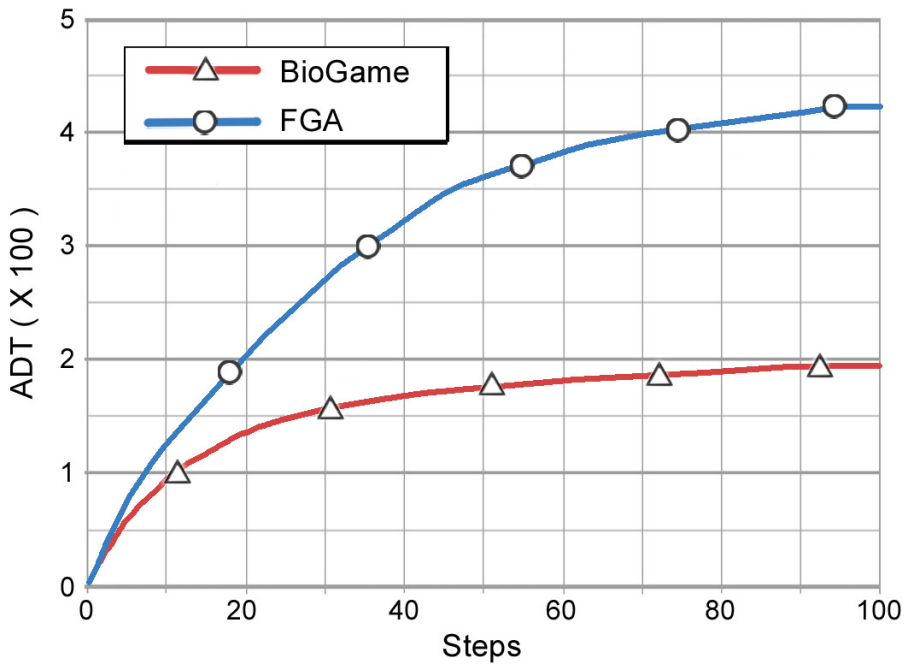


Figure 41: The ADT by a mobile agent in the networks consisting of 40 nodes running BioGame (denoted by the line marked △) and FGA (denoted by the line marked ◯).

traveled by a node running FGA at this point was almost twice that for BioGame. Specifically, Fig. 41 shows that at step 35, ADT by a node running FGA is 297 whereas it is 162 for a node running BioGame. At step 50, when FGA and BioGame networks approach their maximum area coverage for this example (Fig. 40), a node selecting its next location based on FGA traveled more than twice the distance of a node using BioGame (Fig. 41). Conversely, by the time BioGame achieves 98% coverage by traveling a distance of 162, FGA has only achieved 78% area coverage (i.e., Fig. 41, shows that FGA network ADT is 162 is at step 14). The ability of BioGame to significantly reduce the required dis-

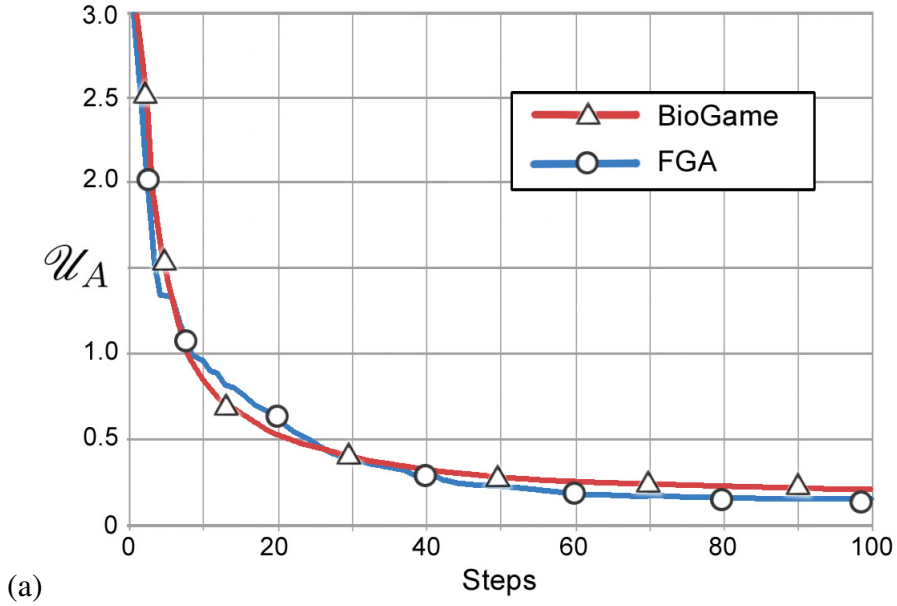
tance that nodes have to travel to accomplish predefined area coverage objectives assures its practical value for all realistic implementations in which power is a scarce resource.

Another observation that we can make from Fig. 41, which is supported by a jittery NAC line in the late iterations of FGA (Fig. 40), is that ADT by a node implementing FGA keeps increasing noticeably until almost the end of the experiments. This observation shows that the nodes running FGA need more time to attain the stable network topology than the BioGame nodes. One reason for FGA's lower performance is that multiple nodes simultaneously move in the same direction, and hence delaying uniform node distribution. These types of inefficient movements are greatly reduced by BioGame since it considers neighbors' future actions.

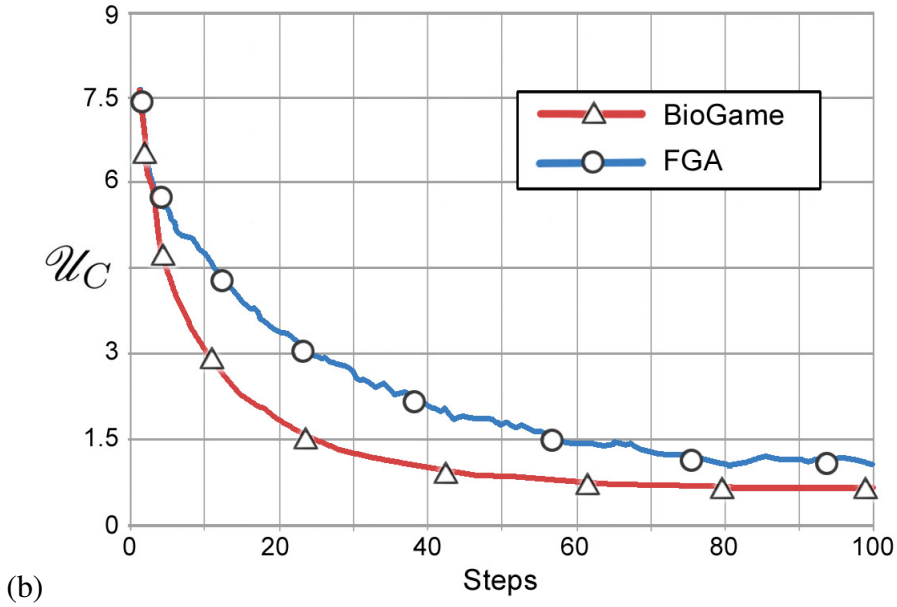
6.2.3 Uniformity

In this section, we demonstrate improvement in network uniformity when our BioGame and FGA evolve towards final distributions. We use techniques given in Sect. 3.7.1 of Chapter 3 that are based on the areas of Voronoi regions generated by nodes (i.e., uniformity measure $\mathscr{U}_A$) and the distance between each node location and the center of mass of generated by its Voronoi region (i.e., uniformity $\mathscr{U}_C$).

Figures 42(a) and 42(b) show improvement of $\mathscr{U}_A$ and $\mathscr{U}_C$ while simulation experiments progress. We can see in Figs. 42(a) and 42(b) that both BioGame and FGA converge rapidly towards the uniform distribution over the area of de-



(a)



(b)

Figure 42: The improvement of uniformities (a) $\mathscr{U}_A$ and (b) $\mathscr{U}_C$ for networks running BioGame (denoted by the line marked △) and FGA (denoted by the line marked ◯).

ployment with results for our BioGame being slightly better. The largest improvement of uniform distribution occurs during the initial iterations of the simulations showing the effectiveness of our approaches at finding remote new locations when space is not constrained. However, as demonstrated in Sect. 6.2.2, BioGame provides a superior method for the spreading of autonomous mobile agents over an unknown terrain since nodes utilizing it have to move much less while providing better results with respect to NAC (Sect. 6.2.1) and appropriate separation among agents.

6.2.4 Simulation of Hostile Attack and Random Node Malfunction

In order to simulate realistic scenarios in which the number of nodes in a network fluctuates due to external factors, we implement a *hostile attack* scenario, that results in the destruction of nodes occupying part of the deployment area, and a *node malfunction* scenario, which makes randomly selected nodes cease to function for the remainder of the experiment.

Figure 43 depicts the progression of a BioGame model during which hostile attack and random node malfunctions reduce the number of MANET nodes. Figure 43(a) shows a snapshot of the network topology attained by 60 nodes running BioGame after the first 10 steps. We can see in Fig. 43(a) that the nodes are able to cover most of the deployment terrain. After step 10, we create a hostile attack that permanently disables all nodes located in the lower-left area of the deployment terrain. The area under attack is outlined by the small square shown in the lower-left corner in Fig. 43(b). Crosses in Fig. 43(b) represent the

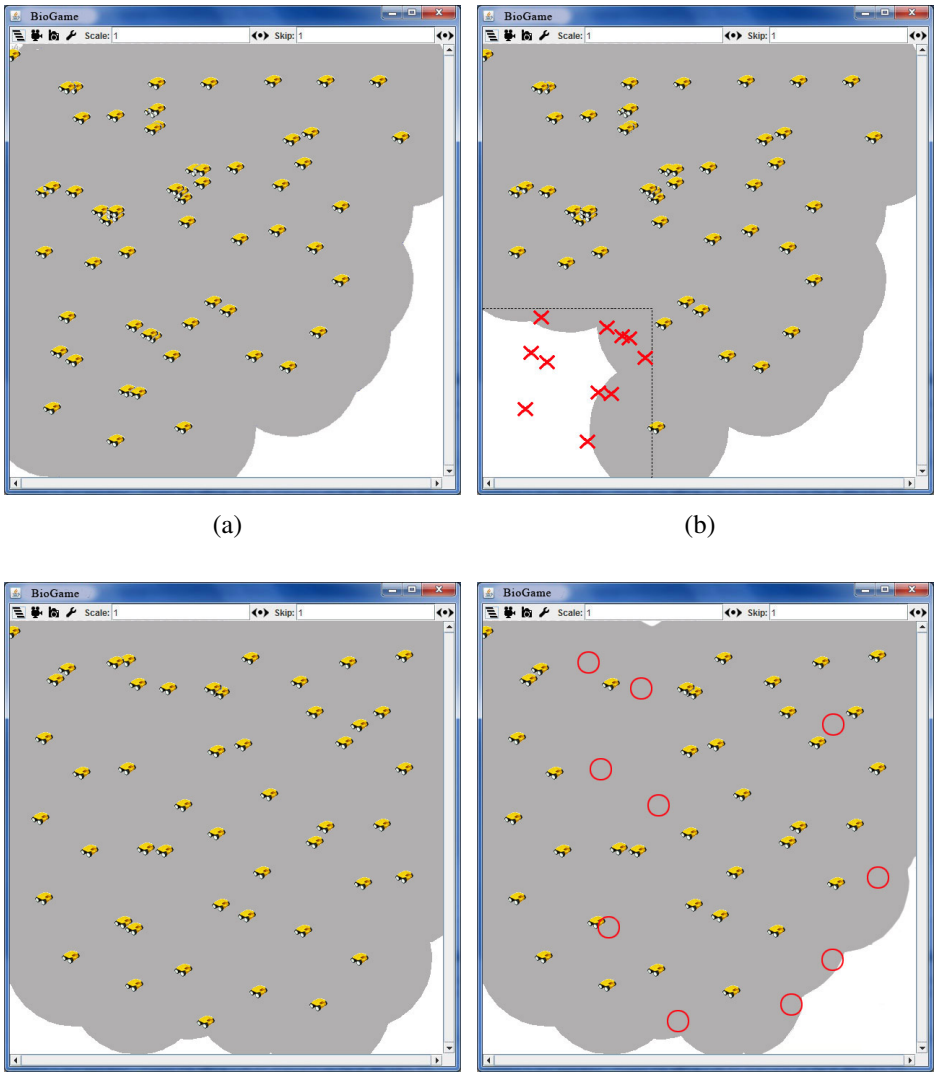


Figure 43: Topology snapshots of the network experiencing the hostile attack and malfunction of randomly selected nodes. Network topologies at: (a) step 10 shortly before the hostile attack; (b) step 11 when the attack disabled nodes placed in 16% of the deployment area; (c) step 20 prior to node malfunction; (d) step 21 after additional 10 randomly selected nodes became disabled.

last known locations of the disabled nodes. Figure 43(b) indicates that the total population of nodes in this example was reduced to 49 nodes.

After the hostile attack, the remaining nodes run BioGame for 10 more steps without disturbance and are able to compensate for losses in the total area coverage by finding new profitable locations. Figure 43(c) shows the population of the remaining 49 mobile nodes shortly before the malfunction of 10 additional randomly selected nodes. Figure 43(c) represents a network topology at step 21 after the node malfunctions with a population of 39 nodes. The last known positions of the disabled nodes are represented by circles.

The final node distribution depicted in Fig. 44 was achieved after running BioGame for 100 steps, where the 60 initially deployed nodes were reduced to 49 mobile agents after step 10 due to hostile attacks destroying all nodes in the lower-left corner of the deployment terrain, and further reduced to 39 nodes after step 20 when 10 additional randomly selected nodes stopped functioning.

We can see in Fig. 44 that the nodes are able to compensate for those lost by repositioning themselves to maximize the distance from their neighbors and, consequently, improve the total area coverage.

Figure 45 shows NAC improvement when the network with 60 nodes undergoes the hostile attack at step 10 and the random malfunctions at step 20.

We can observe in Fig. 45 that the mobile nodes cover almost 90% of the area shortly before the attack. After the attack, NAC drops to 0.78 due to the lost nodes. During the next 10 steps, NAC recovers, reaching the a value of approxi-

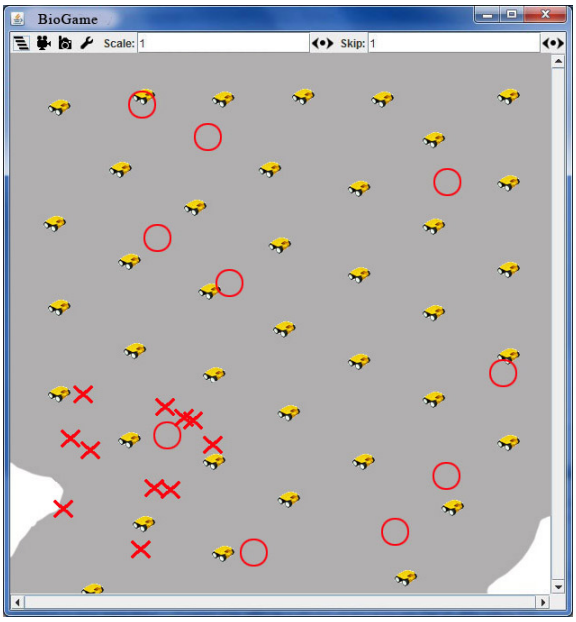


Figure 44: A typical final node distribution for the network with initial population of 60 autonomous agents decremented by 11 due to a hostile attack (the last position of each disabled nodes is marked $\times$) and additional 10 agents due to node malfunctions (the last positions of each malfunctioning node is marked $\bigcirc$).

matively 0.98. A NAC loss of approximately 0.07 occurred at step 20, when 10 additional nodes became disabled. We observe less significant NAC loss at this stage since the 10 malfunctioning nodes were selected randomly and some were placed in densely populated regions with many neighboring nodes providing redundant area coverage for the surrounding terrain. The main network recovery task after this step is to increase area coverage with the remaining resources. It can be seen in Fig. 45 that our BioGame is able to recover after additional nodes became inactive and converge towards a uniform distribution. Figure 45 shows that the network with the remaining nodes is able to provide NAC of ap-

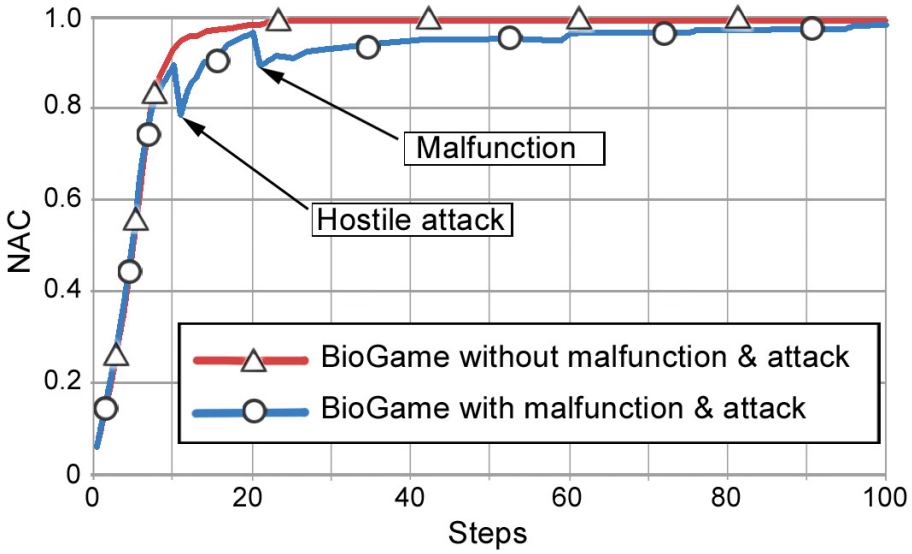


Figure 45: NAC improvement for network with 60 nodes decreased by the hostile attack at step 10 and randomly disabled nodes at step 20. This figure shows also NAC improvement for the same number of nodes running BioGame without any disturbance.

proximately 0.99, which is consistent with our results for similar network sizes (see NAC of the network with 40 nodes running BioGame in Fig. 40). We can see in Fig. 45 that a network composed of 60 nodes running BioGame without disturbance is able to achieve full coverage of the deployment terrain at step 23.

In summary, our simulation results show that BioGame combining NAC with GT can be a power-efficient and resilient technique, providing a promising level of area coverage with near-uniform node distribution while utilizing only local information by each autonomous agent and without synchronization among nodes.

6.3 Observations

BioGame is a novel approach for self-spreading autonomous nodes over an unknown geographical territory by combining our new FGA and GT. Our BioGame runs at each mobile node making independent movement decisions based on the outcome of locally run FGA. FGA takes into account the coordinates of neighboring nodes to find a set of plausible next locations and does not consider possible moves of other mobile agents. However, FGA computes a subset of next possible locations that is significantly smaller than the initial search space, allowing our spatial game to perform more refined calculations. A node makes the final decision of where it should move by evaluating the outcome of the spatial game setup among itself and its neighbors. This step replaces the roulette wheel or elitism selection mechanisms used in classic GAs to make a final movement decision. In this section, we formally define FGA and spatial game parts of our BioGame. We prove that BioGame improves the decision of a node to find the next location and performs at least as well as FGA with respect to network area coverage and convergence speed. Our simulation experiments verify that BioGame outperforms FGA in area coverage, uniform distribution, and convergence time. However, as a direct result of finding accurate new positions, we observe significant savings by BioGame for the total distance traveled by the nodes, which can greatly reduce power requirements of a node. Our BioGame adapts gracefully when the number of MANET nodes decrease due to the equipment malfunction and hostile activities resulting in concentrated losses in a given region. We show that in

BioGame, the spatial game effectively and efficiently utilizes information about possible next positions of neighbors in order to enhance FGA performance.

7 Comparison of Results Obtained by NSPG, NSEG, and BioGame

As described in Chapters 4–6, there are some significance differences in the underlying properties of the three node positioning models that we introduced. In this chapter, we compare their performance using specific metrics. Our self-positioning techniques for MANETs are distinct in their decision-making processes and representations of node movements and area coverages. Continuous, square, and hexagonal space representations are adopted by NSPG, NSEG, and BioGame, respectively. Because of this different grid systems, we could not use the same number of nodes in all three methods for comparing their performance. For example, networks running NSPG and BioGame have 40 nodes with $R_C = 16$ while NSEG is represented by a population of 80 nodes with $R_C = 1$, which, in the model with 10×10 logical cells (each 10×10 unit squares), provides roughly the same coverage area for a node in each compared approach. The choice of 80 nodes in NSEG coincides with our earlier experimental results (Sect. 5.5 in Chapter 5) that showed full coverage found by NSEG occurring with this larger number of nodes. Because NSPG uses a logical square grid to model our game in a discrete domain, the optimal node degree for a mobile agent running NSPG is eight and the total number of nodes to effectively cover the deployment terrain is 80, as has been demonstrated in Fig. 34 of Sect. 5.5 in Chapter 5. Each simulation experiment proceeds 200 steps and is repeated 10 times to reduce the noise in the collected data.

In Fig. 46, we demonstrate the improvement of NAC for networks with nodes locally running NSPG, NSEG, and BioGame.

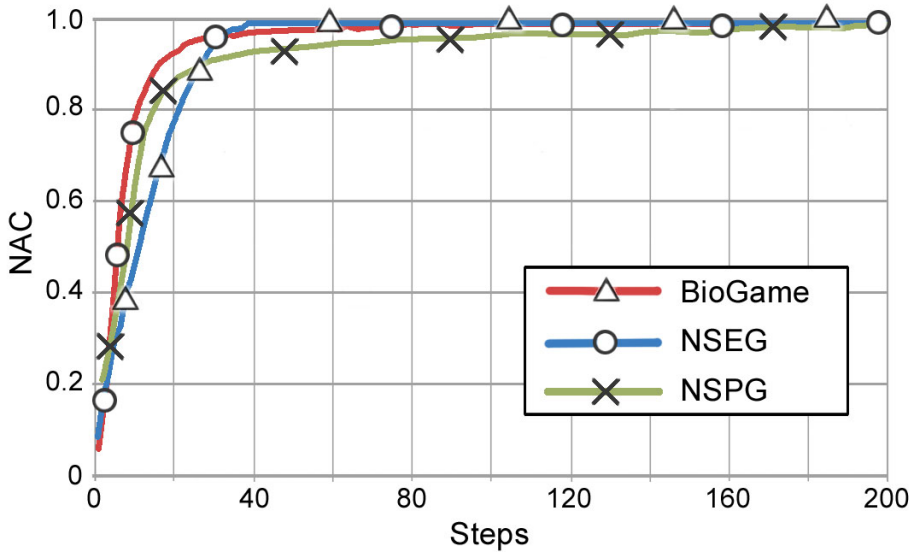


Figure 46: NAC improvement for nodes running NSPG, NSEG, and BioGame.

Figure 46 shows that all our node-spreading techniques provide similar area coverages, which is consistent with our postulate to simulate NSPG, NSEG, and BioGame with parameters providing comparable NACs. We can observe in Fig. 46 that NSEG achieves NAC = 1 after approximately 40 steps while NSPG requires nearly 200 steps to fully cover the territory. However, the most significant NAC improvement for NSPG occurs during the initial 40 iterations of the simulation experiments whereas the following steps bring only minor NAC increases for the network.

Figure 47 illustrates ADTs for networks in which autonomous agents select their locations using NSPG, NSEG, and BioGame. We can see in Fig. 47 that in

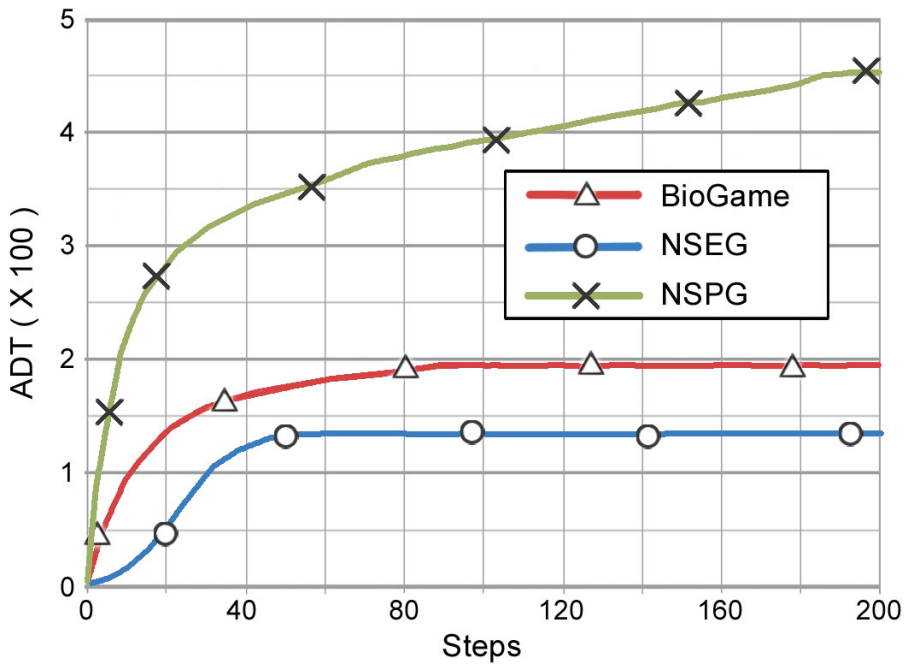
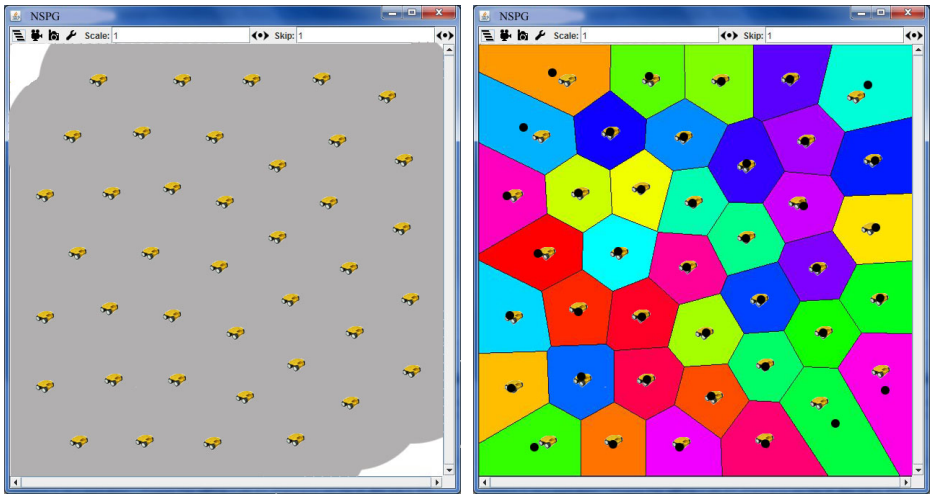


Figure 47: Comparison of ADTs by nodes running NSPG, NSEG, and BioGame.

our simulation experiments a node running NSEG travels less on average than a node finding its new location using BioGame or NSPG. Recall that NSEG experiments use 80 nodes as opposed to 40 for each BioGame and NSPG, could contribute to its dramatic reduction in ADT. The increased number of nodes in NSEG could contribute to its reduction in ADT. Furthermore, Fig. 47 demonstrates that NSPG performs poorer with respect to ADT than both NSEG and BioGame, as NSPG nodes keep adjusting their locations during almost all iterations of the experiments to improve uniformity of the network distribution. These small node movements can be seen in Fig. 47 as the rise in ADT till the end of each simulation experiment. We can also determine from Figs. 46 and 47 that at step 40,

when all compared networks reach more than 90% of the area coverage, ADT for NSEG, BioGame, and NSPG, are approximately 120, 170, and 330, respectively.

Figures 48 and 49 show typical final node distributions achieved by NSPG and NSEG, respectively, and their corresponding Voronoi tessellations of the

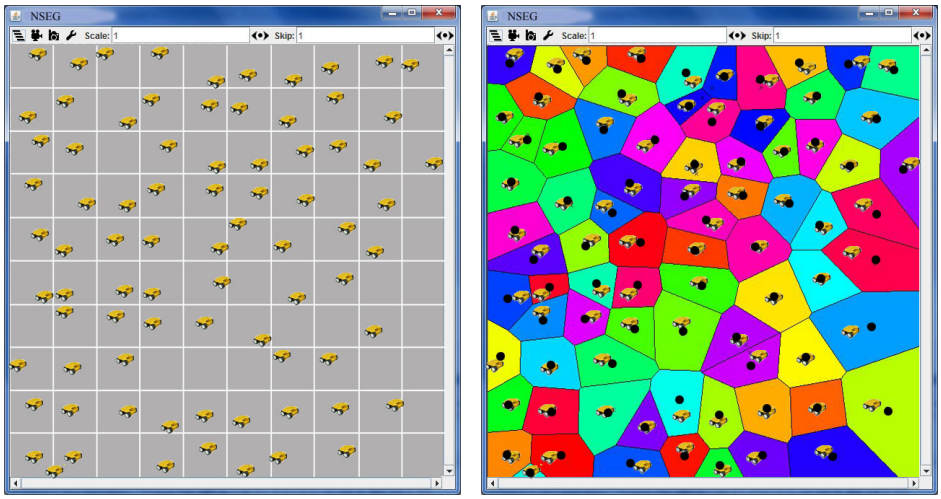


(a) (b)

Figure 48: Example of typical final node distribution (a) and corresponding Voronoi tessellation (b) obtained by our NSPG.

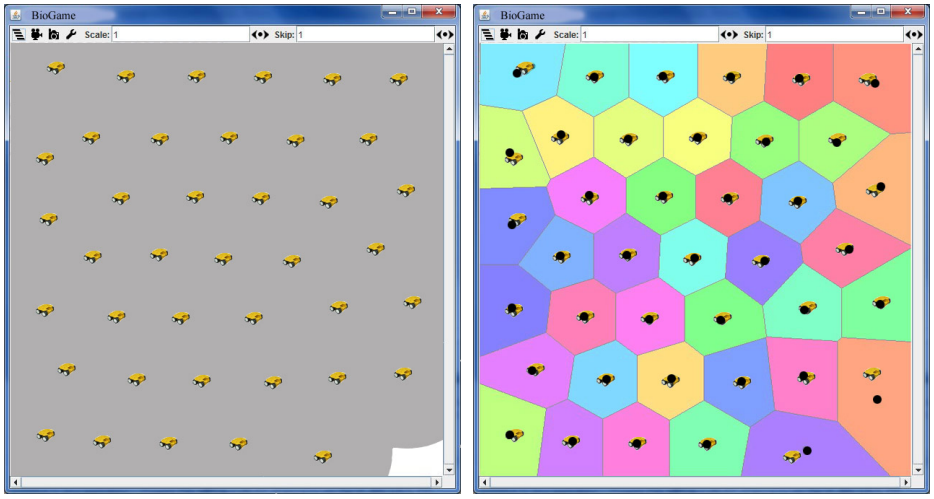
deployment terrain. A final distribution and related Voronoi tessellation attained by our BioGame is presented in Fig. 14 of Sect. 3.7.1 in Chapter 3 but repeated here for comparison in Fig. 50.

As in our earlier illustrations of Voronoi tessellations, center of masses for Voronoi regions are marked by black dots. Following our previous observations and by examining Figs. 48–50, we can determine that, although NSEG provides the maximum area coverage early during the simulation experiments, NSPG and BioGame methods guide nodes towards superior uniform distributions. This



(a) (b)

Figure 49: Example of typical final node distribution (a) and corresponding Voronoi tessellation (b) obtained by our NSEG.



(a) (b)

Figure 50: Example of (a) node distribution and (b) Voronoi tessellation obtained by our BioGame at step $t = 50$.

observation is substantiated by Figs. 51 and 52 where the network uniformity attained by nodes running NSEG is inferior to the uniformities attained by NSPG and BioGame.

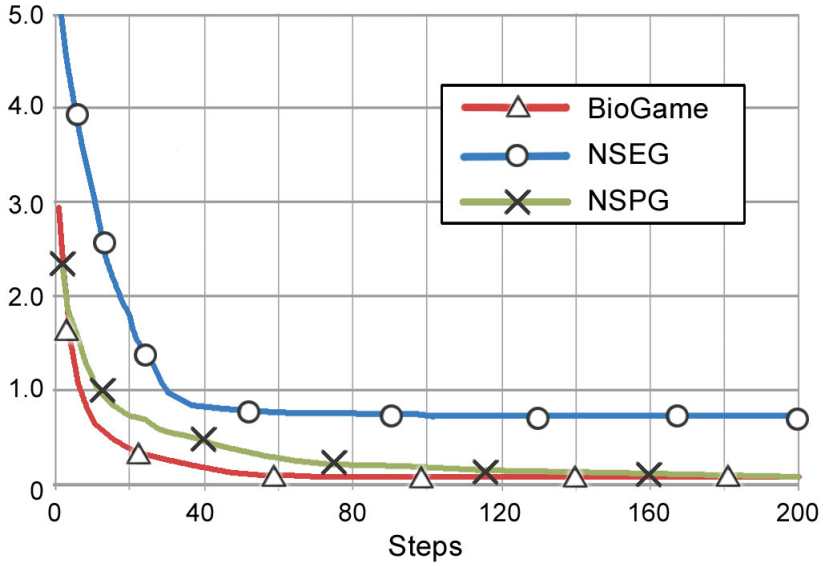


Figure 51: The improvement of uniformity $\mathscr{U}_A$ for networks running NSPG, NSEG, and BioGame.

Figures 51 and 52 indicate that a network running BioGame converges quicker towards a uniform distribution over the area of deployment than NSPG and NSEG can provide and show that NSPG keeps improving network uniformity almost until the end of the simulation experiments, which is reflected by growth of its ADT (Fig. 47).

Comparison of the results for simulation experiments presented in this section demonstrates that the mobile agents running NSEG need to travel a shorter distance before covering the entire area of deployment than the networks using BioGame or NSPG. However, both BioGame and NSPG are superior to NSEG

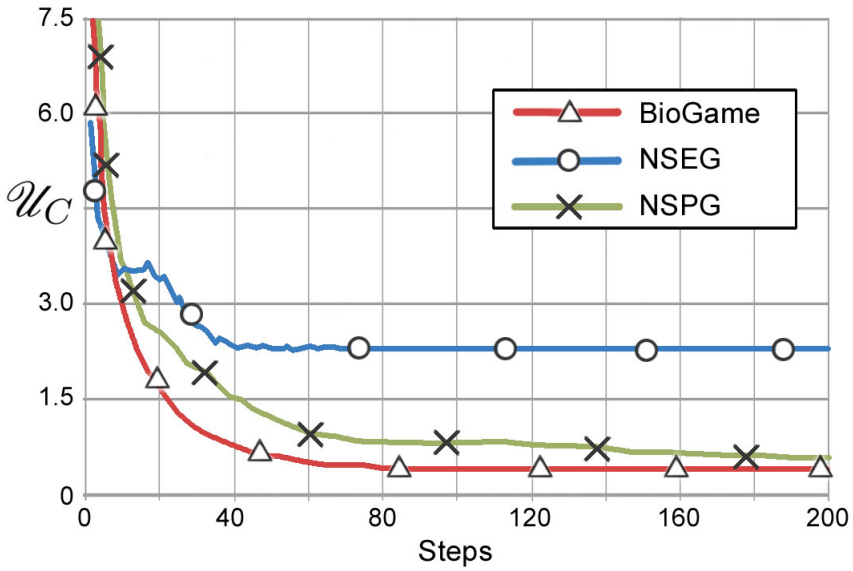


Figure 52: The improvement of uniformity $\mathscr{U}_C$ achieved by networks running NSPG, NSEG, and BioGame.

with respect to uniform distribution of nodes, with BioGame requiring considerable less time than NSPG to reach a stable network topology in which mobile agents cannot further improve their positions. Thus, NSEG provides promising model for applications that require the smallest amount of node movements during the network formation phase. This suggest that NSEG is best suited to applications where power consumption is more important than network coverage. On the other hand, BioGame offers the best overall performance when NAC, ADT, and network uniformity are considered to be equally important objectives. BioGame also delivers the best improvements in both NAC and uniformity metrics in the earlier stages of execution suggesting its usefulness in the situations where quick spreading of mobile nodes is crucial. Our NSPG offer satisfactory

results with respect to uniform node distribution and area coverage. In addition, its resilience against hostile activities and node malfunctions together with its good performance in the areas with strategically placed obstacles (Sect. 4.4 in Chapter 4) suggest that NSPG can be a candidate for deployment into harsh rescue military theatres.

8 Concluding Remarks

In this thesis, we introduce several novel approaches for self-positioning of autonomous nodes in an unknown geographical territory. Our methods combine various concepts of traditional game theory (GT), evolutionary GT (EGT), and genetic algorithms (GAs), to effectively and efficiently guide autonomous MANET nodes in finding the best positions without requiring a centralized controller or a priori information about the deployment terrain and the states of other nodes. In our approaches, myopic actions of each individual node lead the entire network towards a stable and uniform distribution.

We introduce quantitative methods to evaluate the performance of node self-positioning techniques with respect to their area coverage (NAC), average distance traveled by each mobile agent (ADT) before the desired network topology is reached, and uniformity of node distribution. NAC assesses how effective a given population of nodes is in maximizing the area monitored by them. Therefore, achieving a substantial NAC with limited number of nodes is a critical requirement for many MANET topologies. ADT let us estimate a distance traveled by each node before the network reaches its stable arrangement. Since node's movement is one of the most energy-consuming operation that mobile agent performs, ADT indicates power-efficiency of a node self-spreading algorithm in attaining its final topological objective. Uniform distribution of nodes is one of the most desired network topologies. It helps to prolong network's lifespan by ensuring that nodes deplete their energy resources evenly by equally sharing

sensing and communication tasks, and hence, no single node stops functioning prematurely. In order to gauge the performance of a network with respect to its uniform distribution, we introduce two metrics based on the Voronoi tessellation of their deployment terrain.

Our first novel method for self-positioning of autonomous nodes is a node-spreading potential game (NSPG) combined with a genetic algorithm. In NSPG simulation experiments, MANET nodes are deployed in an unknown geographical terrain with strategically placed obstacles. We show that NSPG performs well with respect to convergence speed and adaptability to adverse terrain conditions such us arbitrarily placed obstacles. We also demonstrate that our NSPG degrades gracefully when the number of MANET nodes decrease either due to equipment malfunction or hostile activities. In NSPG, a GA uses potential game's payoff function to evaluate fitness of the next possible positions and determine the best next location to move. We prove important properties of NSPG and verify that the total area covered by the nodes improves in each step until a stable network state is reached.

We formalize another new approach for self-spreading of autonomous nodes over an unknown geographical territory by combining a force-based genetic algorithm (FGA), GT, and EGT. We present a formal analysis of our node-spreading evolutionary game (NSEG) and prove that the evolutionary stable state is its convergence point. Our simulation results demonstrate that NSEG performs well with respect to the network area coverage, the uniform distribution of mobile nodes, and the convergence speed.

We also introduce a bio-inspired game (BioGame) for self-positioning of autonomous nodes, which combines our FGA and GT concepts. In BioGame, each mobile node makes independent movement decisions based on the outcome of a locally run FGA. FGA returns a subset of next possible locations that is significantly smaller than the initial search space, allowing our spatial game to perform more refined calculations. A node makes its final movement decision by evaluating an outcome of the spatial game set up among itself and its neighbors. This step replaces the roulette wheel or elitism selection mechanisms used in classic genetic algorithms to make a final movement decision. We show that in BioGame the spatial game effectively and efficiently utilizes information about possible next positions of neighbors in order to enhance FGA performance. Each node pursues its own selfish goal of reducing the total virtual force inflicted on it by optimally positioning itself in the deployment terrain. Our simulation experiments verify that both BioGame and FGA provide near-uniform node distribution. However, BioGame outperforms FGA in both area coverage and convergence time. We observe significant savings from BioGame for ADT of the nodes, which greatly reduces power consumption.

In this thesis, we also compare the experimental results achieved by NSPG, NSEG, and BioGame with respect to NAC, ADT, and our uniformity metrics. Our observations indicate that, while all proposed node self-positioning techniques provides satisfactory NACs, nodes running NSEG need to travel shorter distance before covering the entire area of deployment than networks employing BioGame or NSPG approaches. However, both BioGame and NSPG are supe-

rior to NSEG regarding uniform distribution of nodes, with BioGame requiring considerable less time than NSPG to stabilize a network topology. Generally, BioGame offers the best overall performance when NAC, ADT, and network uniformity are considered to be equally important objectives.

We developed a Java-based modeling platform to conduct simulation experiments to measure the performances of our approaches for mobile networks. The Java environment allows for real-time visualization of ongoing network dynamics and collection of all critical data. Our analysis and experimental results demonstrate that NSPG, NSEG, and BioGame successfully combine GT, EGT and GA concepts for autonomous node placement and provide useful and resilient methods for optimizing MANET topology.

8.1 Prospect Work and Applications

The immediate extension of our work will include simulation experiments of NSPG, NSEG, and BioGame on more realistic terrain data from military and civilian applications. We will show their capabilities to form different node distribution topologies such as star, ring, or linear formations. We also intend to develop a realistic testbed implementation for our techniques using smartphones running independently copies of our self-positioning algorithms.

9 Our Published Research Results

During the course of this research, we have published our results in various peer-reviewed books, journals, and conference proceedings. In this chapter, we provide a list of our articles and briefly describe each of them. Figure 53 shows a dependency graph of our publications in which an edge incident to a target node indicates that a publication represented by it is based on the results of the paper represented by a source node. In Fig. 53, B_x denotes a book chapter with subscript specifying its chronological order with respect to another chapter. Accordingly, we let J_x and C_x denote journal and conference papers, respectively, with subscripts labeling the order of their authoring.

9.1 Refereed Journal Papers

$\boldsymbol{J_1}$ [34] J. Kusyk, E. Urrea, C.S. Şahin, M.U. Uyar, "Game theory and genetic algorithm based approach for self positioning of autonomous nodes Game theory and genetic algorithm based approach for self positioning of autonomous nodes," (in print). An International Journal of Ad Hoc & Sensor Wireless Networks, (*Old City Publishing*).

In this journal paper, we presented a formal analysis of our distributed NSPG for self-positioning of autonomous nodes. A summary of this publication is given in Chapter 4.

$\boldsymbol{J_2}$ [31] J. Kusyk, C.S. Şahin, M.U. Uyar, E. Urrea, S. Gundry, "Self Organi-

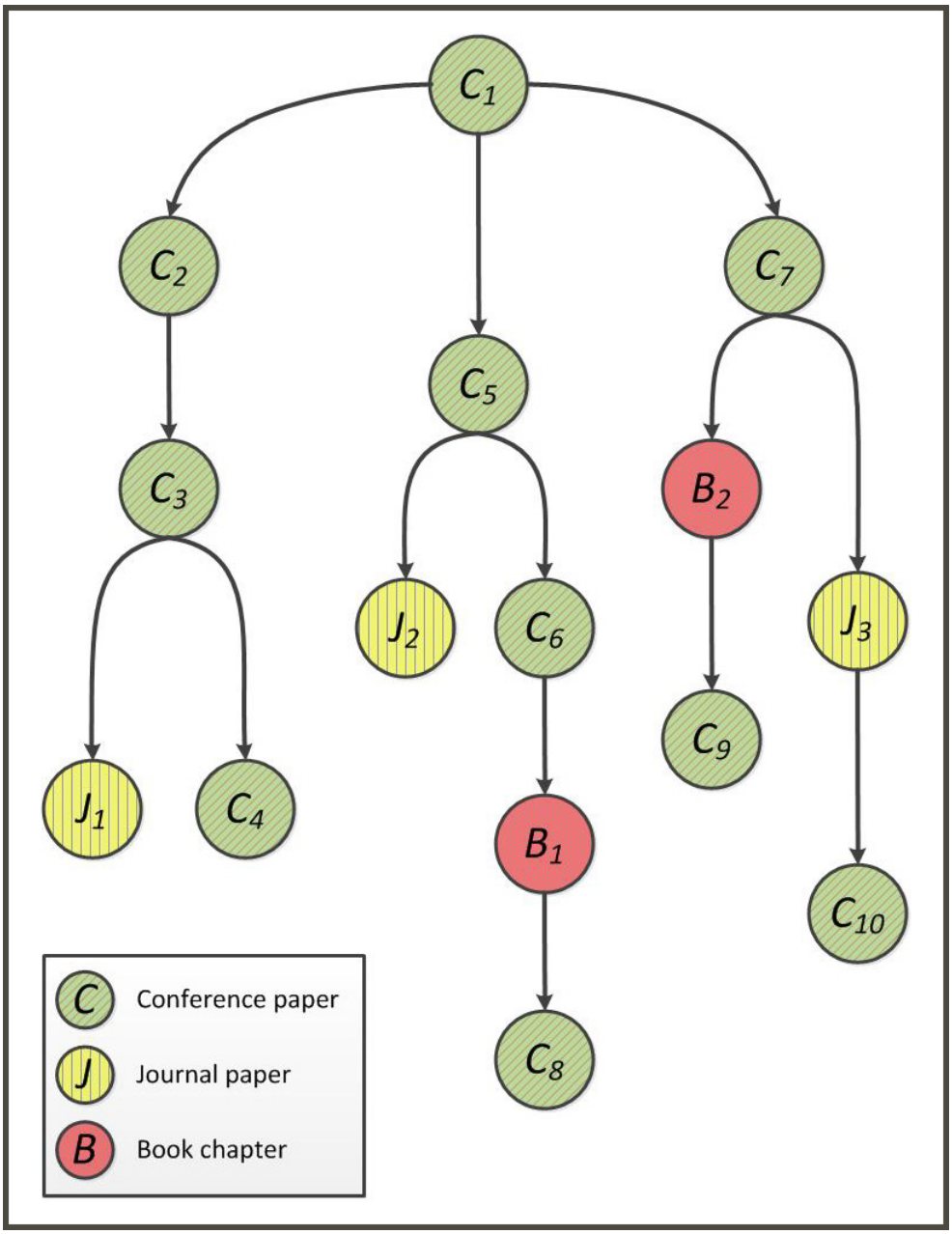


Figure 53: Dependency graph of our publications resulted from GT- and GA-based topology control research.

zation of Nodes in Mobile Ad Hoc Networks Using Evolutionary Games and Genetic Algorithms," (invited paper). The Journal of Advanced Research, special issue on Mobile Ad-Hoc Wireless Networks, (*Elsevier*).

A node-spreading evolutionary game, called NSEG, is introduced in this manuscript. NSEG is based on force-based genetic algorithm and a traditional GT, where each node participates in a spatial normal-form game to determine its actions. Evolutionary game theory framework is used to formally evaluate the network performance. An abridged version of this journal paper is given in Chapter 5.

J_3 [77] J. Zou, S. Gundry, J. Kusyk, M.U. Uyar, and C.S. Şahin, "3D Genetic Algorithms for Underwater Sensor Networks," (in review). International Journal of Ad Hoc and Ubiquitous Computing, special issue on Underwater Sensor Networks: Technology and Theory, (*Interscience Publishers*).

Based on our previous research (C_7), we present in this journal paper 3D-GA for autonomous underwater vehicles (AUVs) in underwater sensor networks (UWSN). Using limited information collected from its own local neighbors, 3D-GA runs autonomously at each AUV and provides guidance for its speed and direction towards a maximum spatial coverage within unknown underwater spaces while maintaining network connectivity. AUVs operating in unknown underwater spaces face more challenges since they often operate with less precise knowledge of each other and are equipped with shorter communication range devices. Imprecise and limited neigh-

borhood knowledge could potentially disrupt convergence towards a uniform and stable spatial coverage. We demonstrate that AUVs running our 3D-GA create a highly resilient network that can adapt to changing conditions such as the addition, loss or malfunction of AUVs. We also show that the ambiguity in detecting neighbors' exact locations (for example due to deteriorated underwater conditions) does not prevent 3D-GA from achieving a uniform spatial coverage but requiring AUVs travel longer distances to stabilize. Our simulation software results verify that 3D-GA can be an effective tool for providing a robust solution for volumetric spatial control of AUVs in UWSN within unknown spaces.

9.2 Book Chapters

B_1 [32] J. Kusyk, C.S. Şahin, J. Zou, S. Gundry, M.U. Uyar, and E. Urrea, "Game Theoretic and Bio-inspired Optimization Approach for Autonomous Movement of MANET Nodes," (in press). 2012 Handbook of Optimization, (*Springer*).

This book chapter presents BioGame that is based on our preliminary work introduced in C_6. In BioGame, we define a new FGA to finds a set of preferred next locations to move. Next, favorable locations identified by FGA are evaluated by the spatial game set up among a moving node and its current neighbors. A formal presentation and analysis of BioGame is given in Chapter 6 of this thesis.

B_2 [19] S. Gundry, J. Zou, E. Urrea, C.S. Şahin, J. Kusyk, and M.U. Uyar, "Analysis of Emergent Behavior for GA-based Topology Control Mechanism for Self-Spreading Nodes in MANETs," (in press). Advances in Intelligent Modeling and Simulation: Artificial Intelligence-based Models and Techniques in Scalable Computing.

In this paper, we introduce a genetic algorithm based MANET topology control mechanism to be used in decision making process of adaptive and autonomic systems at run time. A mobile node adapts its speed and direction using limited information collected from local neighbors operating in an unknown geographical terrain. We represent the genetic operators (i.e., selection, crossover and mutation) as a dynamical system model to describe the behavior of a single nodes decision mechanism. In this dynamical system model each mobile node is viewed as a stochastic variable. We build a homogeneous Markov chain to study the convergent nature of multiple mobile nodes running our algorithm, called FGA. Each state in our chain represents a configuration of the nodes in a MANET for a given instant. The homogeneous Markov chain model of our FGA is shown to be ergodic; its convergence is demonstrated using Dobrushins contraction coeficients. We also observe that the nodes with longer communication ranges utilize more information about their neighborhood to make better decisions, require less movement and converge faster, whereas smaller communication ranges utilize limited information, take more time to es-

cape local optima, and hence consume more energy.

9.3 Refereed Conference Papers

C_1 [37] J. Kusyk, M.U. Uyar, E. Urrea, M.A. Fecko, S. Samtani, "Applications of Game Theory to Mobile Ad Hoc Networks: Node Spreading Potential Game." 32^{nd} IEEE Sarnoff Symposium, Mar. 30, 31 & Apr. 1, 2009, Princeton, NJ, USA.

In this paper, we introduced a node-spreading potential game for uniform distribution of MANET nodes over an unknown geographical territory. We prove that our game is an ordinal potential game and thus guarantees convergence of the proposed algorithm based on it to a stable state. We implemented a proposed model using MASON [44], a discrete-event multi-agent simulation tool developed in Java, to evaluate its performance. Initial results from simulation experiments show that our approach can be an effective mechanism for distributed tasks such as uniform node distribution.

C_2 [38] J. Kusyk, M.U. Uyar, C.S. Şahin, E. Urrea, M.A. Fecko, S. Samtani, "Efficient Node Distribution Techniques in Mobile Ad Hoc Networks Using Game Theory."2009 Military Communication Conference (MILCOM 2009), Oct. 18 - 21, 2009, Boston, MA, USA.

This publication shows that our distributed game introduced in C_1 for MANET nodes to position themselves in an unknown geographical ter-

rain, maximizing the coverage area, combined with a distributed GA to determine the next best location to move provides a near-uniform node spreading. In this paper, we formalized our model's properties and necessary conditions and justify optimality of selected parameters. In our game, each node first finds the most suitable position that it can move using a GA. This step significantly minimizes the computational cost for such problems.

C_3 [33] J. Kusyk, E. Urrea, C.S. Şahin, M.U. Uyar, "Self Spreading Nodes Using Potential Games and Genetic Algorithms." 33^{rd} IEEE Sarnoff Symposium, Apr. 12 - 14, 2010, Princeton, NJ, USA.

Based on our previous research (C_1 and C_2) we presented in this paper a node spreading potential game, for uniform distribution of MANET nodes over an unknown geographical territory. We show the relationship between sensing and communication ranges of a node, and prove that NSPG converges to a stable state. In our simulation experiments, the nodes were deployed in an area with several strategically placed obstacles limiting nodes communication and movement abilities. The experiments were performed for various network densities and communication ranges and showed that NSPG can provide an appropriate coverage of the unknown territory with an acceptable rate.

C_4 [35] J. Kusyk, E. Urrea, C.S. Şahin, M.U. Uyar, G. Bertoli, C. Pizzo, "Resilient Node Self-Positioning Methods for MANEs Based on Game The-

ory and Genetic Algorithms." 2010 Military Communication Conference (MILCOM 2010), Oct. 31 - Nov. 3, 2010, San Jose, CA, USA.

In this conference paper, we presented Rel-NSPG for uniform distribution of autonomous MANET nodes over an unknown geographical territory where nodes' individual actions result in a maximization of the total area coverage. We extended our former game model presented in C_3 to operate in a broad spectrum of geographical terrains with possible hostile activities resulting in nodes occupying an area under attack to stop working and malfunction of randomly selected nodes.

Rel-NSPG, run at each node, autonomously makes movement decisions based on localized data while the best next location to move is selected by a GA. Since it requires only a limited synchronization among the closest neighbors of a player, and does not require *a priori* knowledge of the environment, Rel-NSPG is a good candidate for node-spreading class of applications used in military tasks. The performance of Rel-NSPG degrades gracefully when the number of MANET nodes decrease either due to equipment malfunction or hostile activities. We show that this resilience to loss of nodes is inherent in Rel-NSPG. Simulation experiments demonstrate that, after a subset of the MANET nodes arbitrarily become unavailable, the remaining nodes recover and offset lost nodes. Similarly, when losses are concentrated in a given region, the remaining nodes reconfigure their positions to compensate for the missing area coverage. The simulation

experiments with arbitrarily placed obstacles, in addition to lost assets, produce promising results.

C_5 [36] J. Kusyk, M.U. Uyar, C.S. Şahin, E. Urrea, and S. Gundry, "Game Theory Based Bio-inspired Techniques for Self-positioning Autonomous MANET Nodes." 34^{rd} IEEE Sarnoff Symposium, May. 2 - 4, 2011, Princeton, NJ, USA.

We introduce in this publication a new node-spreading bio-inspired game (NSBG) combining bio-inspired algorithms and traditional game theory to maximize the area covered by autonomous mobile ad hoc network nodes and to achieve a uniform node distribution while keeping the network connected. NSBG is a distributed and scalable game where each nodes selfish actions lead the entire network towards a uniform and stable node distribution without a centralized controller. In NSBG, each mobile node autonomously makes movement decisions based on localized data while the movement probabilities of possible next locations are assigned by our FGA. Because FGA takes only into account the current position of the neighboring nodes, our NSBG, combining FGA with traditional and evolutionary game theory, can find even better locations by setting up spatial games among neighbors. NSBG is a good candidate for the node-spreading class of applications used in both military and commercial applications. We formally introduce our NSBG and analyze results of our simulation experiments.

C_6 [40] J. Kusyk, J. Zou, C.S. Şahin, M.U. Uyar, S. Gundry, and E. Urrea, "A Bio-Inspired Approach Combining Genetic Algorithms and Game Theory for Dispersal of Autonomous MANET Nodes." 2011 Military Communication Conference (MILCOM 2011), Nov. 7 - 9, 2011, Baltimore, MA, USA.

Our BG-Game combining genetic algorithms and traditional game theory was introduce in this publication. BG-Game is fully distributed, scalable, and does not require synchronization among nodes. Each mobile node runs BG-Game autonomously to make movement decisions based solely on localized data. Our simulation experiments demonstrate that BG-Game significantly outperforms FGA and successfully distributes mobile nodes over an unknown geographical terrain without requiring global network information nor any synchronization among the nodes.

C_7 [18] S. Gundry, J. Zou, J. Kusyk, M.U. Uyar, C.S. Şahin, and E. Urrea, "Genetic Algorithms for Self-spreading Autonomous and Holonomic Unmanned Vehicles in a Three-dimensional Space." 2011 Military Communication Conference (MILCOM 2011), Nov. 7 - 9, 2011, Baltimore, MA, USA.

This paper presents a genetic algorithm, called 3D-GA, as a decentralized spatial control mechanism autonomously running in holonomic unmanned vehicles (HUVs) to achieve a uniform distribution in a three dimensional space. In aerial and underwater military theatres this task is difficult due

to dynamic, harsh and bandwidth limited conditions, and lack of a centralized controller. Using only near neighbor information, our 3D-GA guides each HUV to select a velocity vector with a higher fitness among exponentially large number of choices, converging towards a uniform spatial distribution. We demonstrate that the HUVs running our 3D-GA create a highly resilient network that can adapt to changing conditions such as the addition or loss of HUVs due to replenishment, malfunction or destruction. If HUVs are added or removed, the rest of the HUVs will reposition themselves using our 3D-GA to maximize their volumetric coverage of an aerial or underwater space. Our simulation software results verify that our 3D-GA can be an effective tool for providing a robust solution for volumetric spatial control of HUVs in military applications.

C_8 [39] J. Kusyk, J. Zou, S. Gundry, C.S. Şahin, and M.U. Uyar, "Metrics for performance evaluation of self-positioning autonomous MANET nodes," (in review). 35^{rd} IEEE Sarnoff Symposium, Apr. 30 - May. 1, 2012, Princeton, NJ, USA.

In this paper we present new quantitative techniques to assess performance of mobile ad hoc network MANET nodes with respect to uniform distribution, total terrain covered by communication areas of all nodes, and distance traveled by each node before a desired network topology is reached. Our uniformity metrics exploit information of Voronoi tessellation of a deployment territory generated by nodes. Since node movement is a power

consuming task, ADT before the network reaches its final distribution is an important indicator for the performance of MANET nodes. Another performance metric is the NAC achieved by all nodes showing how efficiently the MANET nodes perform. For evaluation of these metrics we use our BioGame presented in Chapter 6. We formally define Voronoi based node uniformity measures, ADT, and NAC in Sect. 3.7 of Chapter 3. Also, our book chapter B_1 presented in Chapter 6 has been extended by the results of these performance evaluation techniques.

C_9 [17] S. Gundry, J. Zou, J. Kusyk, C.S. Şahin, and M.U. Uyar, "Markov chain model for differential evolution based topology control in MANET," (in review). 35^{rd} IEEE Sarnoff Symposium, Apr. 30 - May. 1, 2012, Princeton, NJ, USA.

We introduce in this paper a differential evolution based topological control mechanism for the decision making process of evolutionary and autonomous systems that adaptively reconfigures spatial configuration in mobile ad hoc networks. We also present a formal analysis of the effectiveness of our differential evolution based topology control mechanism and introduce an inhomogeneous Markov chain model to prove its convergence. The experiment results from our simulation software show that our nature-inspired algorithm produces positive results for uniform distribution of mobile nodes over unknown terrains.

C_{10} [76] J. Zou, S. Gundry, J. Kusyk, C.S. Şahin, and M.U. Uyar, "Par-

ticle swarm optimization for autonomous vehicles in a three-dimensional space," (in review). 35^{rd} IEEE Sarnoff Symposium, Apr. 30 - May. 1, 2012, Princeton, NJ, USA.

In this paper, we present a force-based particle swarm optimization (PSO) algorithm as the topology control mechanism running in holonomic unmanned vehicles (HUVs) used in a three dimensional space. Each HUV could adjust its speed and direction to achieve better fitness location using PSO algorithm. We demonstrate that PSO used limited information collected from its own local neighbors to guide HUVs over an unknown 3D space to obtain a uniform spatial distribution while maintaining the whole network connectivity. Simulation experiments show different parameters such as the number of iterations and acceleration coefficients would influence the efficiency of PSO algorithm.

References

[1] S. Adlakha, R. Johari, and A. J. Goldsmith. Competition in wireless systems via Bayesian interference games. *Computing Research Repository*, 0709.0516, September 2007.

[2] C. Ahn and R. S. Ramakrishna. A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation*, 6(6):566–579, December 2002.

[3] I. Ashlagi, D. Monderer, and M. Tennenholtz. Routing games with an unknown set of active players. *6th International Joint Conference on Autonomous Agents and Multiagent Systems*, (195), May 2007.

[4] L. Barolli, A. Koyama, and N. Shiratori. A qos routing method for ad-hoc networks based on genetic algorithm. *Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA)*, pages 175–179, September 2003.

[5] P. Briest, M. Hoefer, and P. Krysta. Stackelberg network pricing games. *Symposium on Theoretical Aspects of Computer Science*, pages 133–142, 2008.

[6] L. Cao and H. Zheng. Distributed spectrum allocation via local bargaining. *IEEE Sensor and Ad Hoc Communications and Networks (SECON)*, pages 475– 486, September 2005.

[7] M. Chen and A. Zalzala. Safety considerations in the optimization of the paths for mobile robots using genetic algorithms. In *Proc. of First Int. Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, 1995.

[8] J. Cortés, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, April 2004.

[9] J. B. Cruz and M. Simaan. Ordinal games and generalized Nash and Stackelberg solutions. *Journal of Optimization Theory and Applications*, 107(2):205–222, November 2000.

[10] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations: applications and algorithms. *Society for Industrial and Applied Mathematics Review*, 41(4):637–676, 1999.

[11] J. Duffy. Introduction to game theory. http://www.pitt.edu/jduffy/econ1200/ Lectures.htm, December 2012.

[12] S. Eidenbenz, V. S. A. Kumar, and S. Zust. Equilibria in topology control games for ad hoc networks. *Mobile Networks and Applications*, 11(2):143–159, April 2006.

[13] S. Fischer and B. Vöcking. Evolutionary game theory with applications to adaptive routing. *European Conference on Complex Systems (ECCS)*, page 104, November 2005.

[14] D. Fudenberg and J. Tirole. *Game theory*. The MIT Press, August 1991.

[15] M. Gairing, B. Monien, and T. Tiemann. Selfish routing with incomplete information. *ACM Symposium on Parallel Algorithms and Architectures*, pages 203 – 212, 2005.

[16] S. Gundry, E. Urrea, C. S. Sahin, J. Zou, J. Kusyk, and M. U. Uyar. Formal convergence analysis for bio-inspired topology control in MANETs. *IEEE Sarnoff Symposium*, pages 1–5, May 2011.

[17] S. Gundry, J. Zou, J. Kusyk, C. S. Sahin, and M. U. Umit. Markov chain model for differential evolution based topology control in MANETs. *IEEE Sarnoff Symposium*, 2012 (in review).

[18] S. Gundry, J. Zou, J. Kusyk, M. U. Uyar, C. S. Sahin, and E. Urrea. Genetic algorithm for self-spreading autonomous and holonomic unmanned vehicles in a three-dimensional space. *IEEE Military Communications Conference (MILCOM)*, pages 1073–1078, November 2011.

[19] S. Gundry, J. Zou, E. Urrea, C. S. Sahin, J. Kusyk, and M. U. Uyar. Analysis of emergent behavior for GA-based topology control mechanism for self-spreading nodes in MANETs. *Advances in Intelligent Modeling and Simulation: Artificial Intelligence-based Models and Techniques in Scalable Computing*, 2012 (in press).

[20] M. M. Halldórsson, J. Y. Halpern, L. E. Li, and V. S. Mirrokni. On spectrum sharing games. *Proceedings of the Twenty-Third annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 107–114, 2004.

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

[22] 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. *Distributed Autonomous Robot Systems 5*, pages 299–308, 2002.

[23] J. Huang, R. A. Berry, and M. L. Honig. Auction mechanisms for distributed spectrum sharing. *Annual Allerton Conference on Communication, Control and Computing*, October 2004.

[24] J. Huang, R. A. Berry, and M. L. Honig. Auction-based spectrum sharing. *ACM/Springer Mobile Networks and Applications*, 11(3):405–418, June 2006.

[25] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[26] Z. Ji and K. J. R. Liu. Belief-assisted pricing for dynamic spectrum allocation in wireless networks with selfish users. *Annual IEEE Communications*

Society on Sensor and Ad Hoc Communications and Networks, 1:119–127, September 2006.

[27] Z. Ji and K. J. R. Liu. Dynamic spectrum sharing: a game theoretical overview. *IEEE Communications Magazine*, 47:88–94, May 2007.

[28] Z. Ji and K. J. R. Liu. Multi-stage pricing game for collusion-resistant dynamic spectrum allocation. *IEEE Journal on Selected Areas in Communication*, 26(1), January 2008.

[29] R. S. Komali and A. B. MacKenzie. Distributed topology control in ad-hoc networks: a game theoretic perspective. *3rd IEEE Conference on Consumer Communications and Networking Conference (CCNC)*, 1:563– 568, January 2006.

[30] R. S. Komali, A. B. MacKenzie, and R. P. Gilles. Effect of selfish node behavior on efficient topology design. *IEEE Transactions on Mobile Computing*, 7(9), September 2008.

[31] J. Kusyk, C. S. Sahin, M. U. Uyar, E. Urrea, and S. Gundry. Self organization of nodes in mobile ad hoc networks using evolutionary games and genetic algorithms. *Journal of Advanced Research. Elsevier*, 2:253–264, July 2011.

[32] J. Kusyk, C. S. Sahin, J. Zou, S. Gundry, E. Urrea, and M. U. Uyar. *Handbook of Optimization*, chapter Game theoretic and bio-inspired optimiza-

tion approach for autonomous movement of MANET nodes. Springer, 2011 (in press).

[33] J. Kusyk, E. Urrea, C. S. Sahin, and M. U. Uyar. Self spreading nodes using potential games and genetic algorithms. *IEEE Sarnoff Symposium*, pages 1–5, April 2010.

[34] J. Kusyk, E. Urrea, C. S. Sahin, and M. U. Uyar. Game theory and genetic algorithm based approach for self positioning of autonomous nodes. *International Journal of Ad Hoc & Sensor Wireless Networks. Old City Publishing*, 2011 (in press).

[35] J. Kusyk, E. Urrea, C. S. Sahin, M. U. Uyar, Giorgio Bertoli, and Christian Pizzo. Resilient node self-positioning methods for MANETs based on game theory and genetic algorithms. *IEEE Military Communications Conference (MILCOM)*, pages 1275–1280, November 2010.

[36] J. Kusyk, M. U. Uyar, C. S. Sahin, E. Urrea, and S. Gundry. Game theory based bio-inspired techniques for self-positioning autonomous MANET nodes. *IEEE Sarnoff Symposium*, pages 1–5, May 2011.

[37] J. Kusyk, M. U. Uyar, E. Urrea, M. Fecko, and S. Samtani. Applications of game theory to mobile ad hoc networks: node spreading potential game. *IEEE Sarnoff Symposium*, pages 1–5, March 2009.

[38] J. Kusyk, M. U. Uyar, E. Urrea, C. S. Sahin, M. Fecko, and S. Samtani. Efficient node distribution techniques in mobile ad hoc networks using game

theory. *IEEE Military Communications Conference (MILCOM)*, pages 1 – 7, October 2009.

[39] J. Kusyk, J. Zou, S. Gundry, C. S. Sahin, and M. U. Umit. Metrics for performance evaluation of self-positioning autonomous MANET nodes. *IEEE Sarnoff Symposium*, 2012 (in review).

[40] J. Kusyk, J. Zou, C. S. Sahinand, M. U. Uyar, S. Gundry, and E. Urrea. A bio-inspired approach combining genetic algorithms and game theory for dispersal of autonomous MANET nodes. *IEEE Military Communications Conference (MILCOM)*, pages 1067–1072, November 2011.

[41] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potential and coordinated control of groups. *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 2968–2973, 2001.

[42] R. C. Lewontin. Evolution and the theory of games. *Journal of Theoretical Biology*, 1:382–403, 1961.

[43] X. Li, H. Shi, and Y. Shang. A sorted rssi quantization based algorithm for sensor network localization. *11th International Conference on Parallel and Distributed Systems*, 1(20-22):557–563, July 2005.

[44] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. MASON: A multiagent simulation environment. *Simulation*, 81(7):517–527, 2005.

[45] A. B. MacKenzie and L. A. DeSilva. *Game theory for wireless engineers*. Morgan and Claypool Publishers, 1 edition, March 2006.

[46] J. Mitola. Software radios: Survey, critical evaluation and future directions. *IEEE Aerospace and Electronic Systems Magazine*, 8(4):25–36, April 1993.

[47] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(44):124–143, 1996.

[48] E. F. Moore. Machine models of self-reproduction. *Proceedings of Symposia in Applied Mathematics*, 14:17–33, 1962.

[49] H. Nguyen, J. Burkardt andM. Gunzburger, L. Ju, and Y. Saka. Constrained CVT meshes and a comparison of triangular mesh generators. *Computetional Geometry*, 42(1):1–19, January 2009.

[50] M. A. Nowak and R. M. May. The spatial dilemmas of evolution. *Intenational Journal of Bifurcation and Chaos*, 3(1):35–78, 1993.

[51] R. Olfati-Saber and R. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. *IFAC World Congress*, 2002.

[52] M. Pan, S. Liang, H. Xiong, J. Chen, and G. Li. A novel bargaining based dynamic spectrum management scheme in reconfigurable systems. *Inter-*

national Conference on Systems and Networks Communications, pages 54–54, October 2006.

[53] P. Rong and M. L. Sichitiu. Angle of arrival localization for wireless sensor networks. *Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd*, 1:374–382, September 2006.

[54] C. S. Sahin. *Genetic algorithms for topology control problems*. Lap Lambert Academic Publishing, 2011.

[55] 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. *Special Issue of the International Journal of Applied Decision Sciences (IJADS) on Decision Support Systems for Unmanned Vehicles*, 3(3):221–238, 2010.

[56] 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. *Proceedings of the 10th annual conference on Genetic and evolutionary computation (GECCO)*, pages 1141–1142, 2008.

[57] 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 Military Communications Conference (MILCOM)*, pages 1–7, November 2008.

[58] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys*, 37(2):164–194, June 2005.

[59] M. Seredynski and P. Bouvry. Evolutionary game theoretical analysis of reputation-based packet forwarding in civilian mobile ad hoc networks. *IEEE International Symphosium on Parallel and Distributed Processing*, pages 1–8, May 2009.

[60] T. Shinchi, M. Tabuse, T. Kitazoe, and A. Todaka. Khepera robots applied to highway autonomous mobiles. *Artificial Life and Robotics*, 7:118–123, Sep. 2000.

[61] M. Simaan and J. B. Cruz. On the Stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications*, 11(5), May 1973.

[62] R. S. Sisodia, B. S. Manoj, and C. S. R. Murthy. A preferred link based routing protocol for ad hoc wireless networks. *Journal of Communications and Networks*, 4(1):14–21, March 2002.

[63] J. M. Smith. *Evolution and the theory of games*. Cambridge University Press, 1982.

[64] J. M. Smith and G. R. Price. The logic of animal conflict. *Nature*, 246:15–18, November 1973.

[65] I. Stojmenovic and S. Datta. Power and cost aware localized routing with guaranteed delivery in unit graph based ad hoc networks: research articles. *Wireless Communication and Mobile Computing*, 4(2):175–188, 2004.

[66] P. D. Taylor and L. B. Jonker. Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 16:76–83, 1978.

[67] E. Urrea. *Knowledge Sharing Agents Using Genetic Algorithms in Mobile Ad Hoc Networks*. PhD thesis, The Graduate Center of the City University of New York, October 2010.

[68] 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. *Ad Hoc Networks*, 7(4):677–689, 2009.

[69] S. van Hoesel. An overwiew of Stackelberg pricing in networks. *METEOR Research Memoranda 042*, 2006.

[70] B. Wang, Z. Ji, and K. J. R. Liu. Self-learning repeated game framework for distributed primary-prioritized dynamic spectrum access. *Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 631–638, June 2007.

[71] B. Wang, K. Liu, and T.C. Clancy. Evolutionary game framework for behavior dynamics in cooperative spectrum sensing. *IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–5, November 2008.

[72] J. W. Weibull. *Evolutionary game theory*. The MIT Press, 1997.

[73] S. Winfree. *Angle of arrival estimation using received signal strength with directional antennas*. PhD thesis, Ohio State University, 2007.

[74] W. Xi, X. Tan, and J. S. Baras. Gibbs sampler based control of autonomous vehicle swarms in the presence of sensor errors. *Conference on Decision and Control*, pages 5084–5090, December 2006.

[75] L. Yanjun, S.H. Chung, S. Ye-Qiong, W. Zhi, and S. Youxian. A two-hop based real-time routing protocol for wireless sensor networks. *IEEE International Workshop on Factory Communication Systems*, pages 65–74, May 2008.

[76] J. Zou, S. Gundry, J. Kusyk, C. S. Sahin, and M. U. Umit. Particle swarm optimization for autonomous vehicles in a three-dimensional space. *IEEE Sarnoff Symposium*, 2012 (in review).

[77] J. Zou, S. Gundry, J. Kusyk, M. U. Uyar, and C. S. Sahin. 3D genetic algorithms for underwater sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 2012 (in review).