Journal of Advanced Research (2011) 2, 253–264

Cairo University

Journal of Advanced Research

Self-organization of nodes in mobile ad hoc networks using evolutionary games and genetic algorithms

Janusz Kusyk [a], Cem S. Sahin [a,*], M. Umit Uyar [a,b], Elkin Urrea [a], Stephen Gundry [b]

[a] *The Graduate Center of the City University of New York, New York, NY 10016, USA*
[b] *The City College of the City University of New York, New York, NY 10031, USA*

Received 23 September 2010; revised 4 March 2011; accepted 10 April 2011
Available online 14 May 2011

KEYWORDS

Evolutionary game;
Genetic algorithms;
Mobile ad hoc network;
Self-organization

Abstract In this paper, we present a distributed and scalable evolutionary game played by autonomous mobile ad hoc network (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 makes movement decisions based on localized data while the movement probabilities of possible next locations are assigned by a forced-based genetic algorithm (FGA). Because FGA takes only into account the current position of the neighboring nodes, our NSEG, combining FGA with game theory, can find better locations. In NSEG, autonomous node movement decisions are based on the outcome of the locally run FGA and the spatial game set up among it and the nodes in its neighborhood. NSEG is a good candidate for the node spreading class of applications used in both military tasks and commercial applications. We present a formal analysis of our NSEG to prove that an evolutionary stable state is its convergence point. Simulation experiments demonstrate that NSEG performs well with respect to network area coverage, uniform distribution of mobile nodes, and convergence speed.

* Corresponding author. Tel.: +1 603 318 5087.
E-mail address: csafaksahin@gmail.com (C.S. Sahin).

Peer review under responsibility of Cairo University.
doi:10.1016/j.jare.2011.04.006

Production and hosting by Elsevier

ELSEVIER

Introduction

The main performance concerns of mobile ad hoc networks (MANETs) are topology control, spectrum sharing and power consumption, all of which are intensified by lack of a centralized authority and a dynamic topology. In addition, in MANETs where devices are moving autonomously, selfish decisions by the nodes may result in network topology changes contradicting overall network goals. However, we can benefit from autonomous node mobility in unsynchronized networks by incentivizing an individual agent behavior in order to attain an optimal node distribution, which in turn can alleviate many problems MANETs are facing. Achieving better spatial placement may lead to an area coverage improvement with reduced sensing overshadows, limited blind spots, and a better utilization of the network resources by creating an uniform node distribution. Consequently, the reduction in power consumption, better spectrum utilization, and the simplification of routing procedures can be accomplished.

The network topology is the basic infrastructure on top of which various applications, such as routing protocols, data collection methods, and information exchange approaches are performed. Therefore, the topology (or physical distribution) of MANET nodes profoundly affects the entire system performance for such applications. Achieving a better spatial placement of nodes may provide a convenient platform for efficient utilization of the network resources and lead to a reduction in sensing overshadows, limiting blind spots, and increasing network reliability. Consequently, the reduction in power consumption, the simplification of routing procedures, and better spectrum utilization with stable network throughput can be easily accomplished.

Among the main objectives for achieving the optimum distribution of mobile agents over a specific region of interest, the first is to ensure connectivity among the mobile agents by preventing the isolated node(s) in the network. Another objective is to maximize the total area covered by all nodes while providing each mode with an optimum number of neighbors. These objectives can be accomplished by providing a uniform distribution of nodes over a two-dimensional area.

As it is impractical to sustain complete and accurate information at each node about the locations and states of all the agents, individual node's decisions should be based on local information and require minimal coordination among agents. On the other hand, autonomous decision making process promotes uncooperative and selfish behavior of individual agents. These characteristics, however, make game theory (GT) a promising tool to model, analyze, and design many MANET aspects.

GT is a framework for analyzing behavior of a rational player in strategic situations where the outcome depends not only on her but also on other players' actions. 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 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 without cooperation enforcement mechanisms.

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, such as economy, sociology, anthropology, and philosophy. Some of the EGT contributions to the traditional theory of game are: (i) alleviation of the rationality assumption, (ii) refinement of traditional GT solution concepts, (iii) and 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 the traditional GT approaches, it may also become an important tool for designing and evaluating MANETs.

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) or heuristically (e.g., hill climbing, gradient decent, and others). However, they may arrive at a local maximum point or miss the group of optimal 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 approach, a forced-based genetic algorithm (FGA) is used by the nodes to select the best location among exponentially large number of choices.

In this paper, we introduce a new approach to topology control where FGA, GT, and EGT are combined. Our NSEG is a distributed game with each node independently computing its next preferable location without requiring global network information. In NSEG, a movement decision for node i is based on the outcome of the locally run FGA and the spatial game set up among i and the nodes in its neighborhood. Each node pursues its own goal of reducing the total virtual force inflicted on it by effectively positioning itself in one of the neighboring cells. In our approach, each node runs FGA to find the set of the best next locations. Our FGA takes into account only the neighboring nodes' positions to find the next locations to move. However, NSEG, combining FGA with GT, can find even better locations since it uses additional information about the neighbors' payoffs. We prove that the optimal network topology is evolutionary stable and once reached, guarantees network stability. Simulation experiments show that NSEG provides an adequate network area coverage and convergence rate.

One can envision many military and commercial applications for our NSEG topology control approach, such as search and rescue missions after an earthquake to locate humans trapped in rubble, controlling unmanned vehicles and transportation systems, clearing mine-fields, and spreading military assets (e.g., robots, mini-submarines, etc.) under harsh and bandwidth limited conditions. In these types of applications, a large number of autonomous mobile nodes can gather information from multiple viewpoints simultaneously, allowing them to share information and adapt to the environment quickly and comprehensively. A common objective among these applications is the uniform distribution of mobile nodes operating on geographical areas without a priori knowledge of the geographical terrain and resources location.

The rest of this paper is organized as follows. Section 'Related work' provides an overview of the existing research. Basics in GT, EGT, and GA are outlined in Section 'Background to GT, EGT, and GA'. Our distributed node spreading evolutionary game NSEG and its properties are presented in Section 'Our node spreading evolutionary game: NSEG'. Section 'Analysis of NSEG convergence' analyzes the convergence of NSEG. The simulation results are evaluated in Section 'Experimental results'.

Related work

The traditional GT applications in wireless networks focus on problems of dynamic spectrum sharing (DSS), routing, and topology control. The topology control in MANETs can be analyzed from two different perspectives. In one approach, the goal is to manage the configuration of a communication network by establishing links among nodes already positioned in a terrain. In this method, connections between nodes are selected either arbitrarily or by adjusting the node propagation power to the level which satisfies the minimal network requirements. In the second approach, the relative and absolute locations of the mobile nodes define the network topology. Topological goals in this scheme are achieved by the movement of the nodes. Our approach falls into the second category where the network desired topology is achieved by the mobile nodes autonomously determining their own locations.

Managing the movement of nodes in network models where each node is capable of changing its own spatial location could be achieved by employing various methods including potential field [1–4], the Lloyd algorithm [5], or nearest neighbor rules [6]. In our previous publications [7–10], we introduced a node spreading potential game for MANET nodes to position themselves in an unknown geographical terrain. In this model, decisions about node movements were based on localized data while the best next location to move was selected by a GA. This GA-based approach in our node spreading potential game used game's payoff function to evaluate the goodness of possible next locations. This step significantly reduced the computational cost for applications using self-spreading nodes. Furthermore, inherent properties of the class of potential games allowed us to prove network convergence. In this paper, we introduce a new approach such that the spatial game played between a node and its neighbors evaluates the goodness of the GA decision (as opposed to our older approach which uses a game to evaluate network convergence).

Some of EGT applications to wireless networks address issues of efficient routing and spectrum sharing. Seredynski and Bouvry [11] 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 Vocking [12], where the traditional GT assumptions are replaced with a lightweight learning process based on players' previous experiences. Wang et al. [13] 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 have been popular in diverse distributed robotic applications and successfully applied to solve many network routing problems [14,15]. The FGA used in this paper was introduced by Sahin et al. [16–18] and Urrea et al. [19], where each mobile node finds the *fittest* next location such that the artificial forces applied by its neighbors are minimized. It has been shown by Sahin et al. [16] 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).

Background to GT, EGT, and GA

In this section, we present fundamental GT, EGT, and GA concepts and introduce the notation used in our publication. An interested reader can find extensive and rigorous analysis of GT in the book by Fudenberg and Tirole [20] and several GT applications to wireless networks in the work of Mackenzie and DeSilva [21], the fundamentals of EGT can be found in the books by Smith [22] and Weibull [23], while Holland [24] and Mitchell [25] present in their works essentials of GA.

Game theory

A game in a normal form is defined by a nonempty and finite set I of n players, a strategy profile space S, and a set U of payoff (utility) functions. We indicate an individual player as $i \in I$ and each player 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 $\boldsymbol{s} = (s_1, s_2, \ldots, s_n)$ and a strategy profile space S is a set $S = S_1 \times S_2 \times \cdots \times S_n$, hence $\boldsymbol{s} \in S$. If $\boldsymbol{s}$ is a strategy profile played in a game, then $u_i(\boldsymbol{s})$ denotes a payoff function defining i's payoff as an outcome of $\boldsymbol{s}$. It is convenient to single out i's strategy by referring to all other players' strategies as $\boldsymbol{s}_{-i}$.

If a player is randomizing 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, i's mixed strategy σ_i is a probability distribution over S_i and $\sigma_i(s_i)$ represents a probability of s_i being played. The support of mixed strategy profile σ_i is a set of pure strategies for which player i assigns probability greater than 0. Similar to a pure strategy game, we denote a mixed strategy profile as a vector $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_n) = (\sigma_i, \boldsymbol{\sigma}_{-i})$, where in the last case we singled out i's mixed strategy. However, contrary to i's deterministic payoff function $u_i(\boldsymbol{s})$ defined for pure strategy games, the payoff function in mixed strategy game $u_i(\boldsymbol{\sigma})$ expresses an expected payoff for player i.

A Nash equilibrium (NE) is a set of all players' strategies in which no individual player has an incentive to unilaterally change her own strategy, assuming that all other players' strategies stay the same. More precisely, a strategy profile $(\sigma_i^*, \boldsymbol{\sigma}_{-i}^*)$ is a NE if

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

A NE is an important condition for any self-enforcing protocol which lets us predict outcomes in a game played by rational players. Any game where mixed strategies are allowed has at least one NE. However, some pure strategy normal form games may not have a NE solution at all.

Evolutionary game theory

The first formalization of EGT could be traced back to Lewontin, who, in 1961, suggested that the fitness of a population member is measured by its probability of survival [26]. Subsequent introduction of an evolutionary stable strategy (ESS) by Smith and Price [27] and a formalization by Taylor and Jonker [28] 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) lead to the 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 be. However, in contrast to the traditional GT models, each organism's strategy is not selected through its reasoning process but determined by its genes and, as such, individual's strategy is hard-wired. EGT focuses on a distribution of strategies in the population rather than on actions of an individual rational player. In EGT, changes in a population are understood as an evolution through time process resulting from natural selection, crossover, mutation, or other genetic mechanisms favoring one phenotype (strategy) over the other(s). Individuals in EGT are not explicitly modeled 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 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 in the game history (i.e., each individual match can be analyzed as an independent game). Because a population size is assumed to be large and the agents are matched randomly, we concentrate on an average payoff for each player, which is an expected outcome for her when matched against a randomly selected opponent. Also, each repeated interaction between players results in their advancing from one generation to the next, at which point their strategy can change. This mechanism may represent organism's evolution from generation to generation by adopting an evermore suitable strategy at the next stage.

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 holds:

$$u(s^*, s^*) > u(s', s^*) \tag{2}$$

$$(u(s^*, s^*) = u(s', s^*)) \wedge (u(s^*, s') > u(s', s')) \tag{3}$$

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

The game model where each player has an equal probability of being matched against any of the remaining population members maybe inappropriate to analyze many realistic applications. Nowak and May [29] recognized that organisms often interact only with the population members in their proximity and proposed a group of spatial games where members of the population are arranged on a two dimensional lattice with one player occupying each cell. In their model, at every stage of the game, each individual plays a simple 2-player base game with its closely located neighbors and sums her payoffs from all these matches. If her result is better than any of her opponents result, she retains her strategy for the next round. However, if there is a neighbor whose fitness is higher than hers, she adopts this neighbor's strategy for the future. Proposed by Nowak and May games [29] offer an appealing learning process for inheritance mechanism which is based on the imitation of the best strategies in the given environment. Spatial games are extensions of deterministic cellular automata where the new cell state is determined by the outcomes of a pure strategy game played between neighbors. They can also be extended to model a node movement in MANETs where the agents' decisions are based only on the local information and where the goal is to model the population evolution rather than an individual agent's reasoning process.

Genetic algorithms

Genetic algorithms represent a class of adaptive search techniques which have been intensively studied in recent years. In the 1970s, GAs were proposed by Holland as a heuristic tool to search large poorly-known problem spaces [30]. His idea was inspired by biological evolution theory, where only the individuals who are better fitted to their environment are likely to survive and generate *offspring*; thus, they transmit their genetic information to new generations. A GA is an iterative optimization method. It works with a number of candidate solutions (i.e., a *population*), instead of working with a single candidate solution in each iteration. A typical GA works on a population of binary strings – each called a *chromosome* and represents a candidate solution. The desired individuals are selected by the evolution of a specified fitness function (i.e., objective function) among all candidate solutions. Candidate solutions with better fitness values have higher probability to be selected for the breeding process. To create a new, and eventually better, population from an old one, GAs use biologically inspired operators, such as tournaments (fitter individuals are selected to survive), crossovers (a new generation of individuals are selected from tournament winners), and mutations (random changes to children to provide diversity in a population) [25,30].

GAs have been used to solve a broad variety of problems in a diverse array of fields including automotive and aircraft design, engineering, price prediction in financial markets, robotics, protein sequence prediction, computer games, evolvable hardware, optimized telecommunication network routing and others. GAs are chosen to solve complex and NP-hard problems since: (i) GAs are intrinsically parallel and, hence, can easily scan large problem spaces, (ii) GAs do not get trapped at local optimum points, and (iii) GAs can easily handle multi-optimization problems with proper fitness functions. However, the success of a GA application lies in defining its fitness function and its parameters (i.e., the chromosome structure).

In most general form of GA, a population is randomly created with a group of individuals (possible solutions) created randomly (Fig. 1). Commonly, the individuals are encoded into a binary string. The individuals in the population are then evaluated. The evaluation function is given by the user which assigns the individuals a score based on how well they perform at the given task. Individuals are then selected based on their fitness scores, the higher the fitness then the higher the probability of being selected. These individuals then reproduce to create one or more offspring, after which the offspring are mutated randomly. A new population is generated by replacing some of the individuals of the old population by the new ones. With this process, the population evolves toward better regions of the search space. This continues until a suitable solution has been found or a certain number of generations have passed.

The terminology used in GA is analogous to the one used by biologists. The connections are somewhat strained, but are still useful. The individuals can be considered to be a chro-

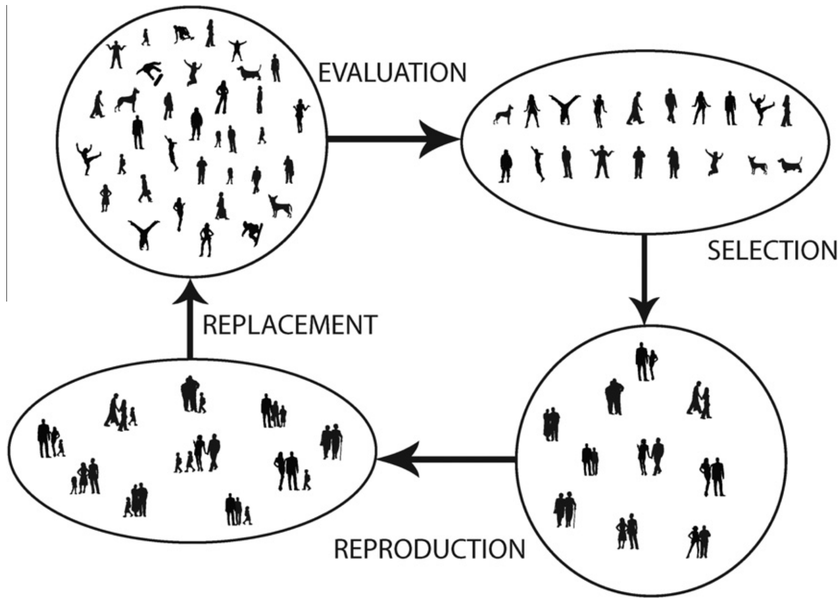

Fig. 1 Basic form of genetic algorithm (GA).

mosome, and since only individuals with a single string are considered, this chromosome is also the genotype. The organism, or phenotype, is the result produced by the expression of the genotype within the environment. In GAs this will be a particular set of unidentified parameters, or an individual candidate solution.

In our NSEG, each mobile node runs FGA introduced by Sahin et al. [16–18] and Urrea et al. [19]. Our FGA is inspired by the force-based distribution in physics where each molecule attempts to remain in a balanced position and to spend minimum energy to protect its own position [31,32]. A virtual force is assumed to be applied to a node by all nodes located within its communication range. At the equilibrium, the aggregate virtual force applied to a node by its neighbors should sum to zero. If the virtual force is not zero, our agent uses this non-zero virtual force value in its fitness calculation to find its next location such that the total virtual force on the mobile node is minimized. The value of this virtual force depends on the number of neighboring nodes within its communication range and the distance among them. In FGA, a smaller fitness value indicates a better position for the corresponding node.

Our node spreading evolutionary game: NSEG

In our NSEG, the goal for each node is to distribute itself over an unknown geographical terrain in order to obtain a high coverage of the area by the nodes and to achieve a uniform node distribution while keeping the network connected. Initially, the 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.

Because our model is partially based on a game theory, we will refer to a node as a player or an agent, interchangeably. Player's strategies will refer logical cells into which she can move, and the payoff will reflect the goodness of a location.

For each node, the set of neighboring cells is defined with respect to its location and its communication radius (R_C) indicating the maximum possible distance to another node to establish a communication channel. In our model, R_C also determines the terrain covered by a node for various different purposes such as monitoring, data collection, sensing, and others. For simplicity, but without loss of generality, we consider a monomorphic population where all the nodes are equipotent and able to perform versatile tasks related to network maintenance and data processing. 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 the cells within a Chebyshev distance smaller or equal to 1) resulting in the set of 9 neighboring cells. In our NSEG, 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.

Fig. 2 shows an area divided into 5×5 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. 2, 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. 2, 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.

In NSEG, a set I of n players represents all active nodes in the network. For all $i \in I$, a set of strategies $S_i = \{NW, N, NE, W, U, E, SW, S, SE\}$ stand for all possible next cells that

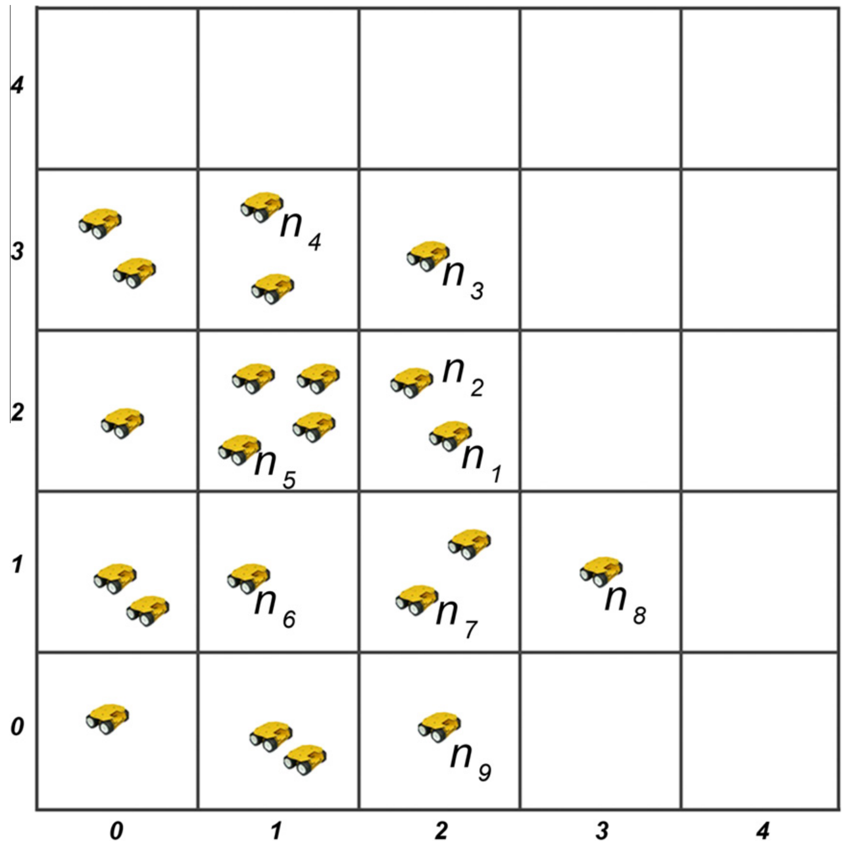

Fig. 2 An example of 5×5 logical lattice populated with 22 nodes (n_1 and n_7 can communicate, but n_1 cannot communicate with n_9).

i can move into. The definitions of NSEG strategies are shown in Table 1.

For example, *NW* is a new location in the adjacent cell *North-West* of i's current location and *U* is the same unchanged location that i inhabits now. In Fig. 2, 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 i by node j located within the same cell (e.g., in Fig. 2, a force on node n_1 caused by node n_2). Similarly, f^1_{ik} is defined as the virtual force inflicted on i by node k located in a cell one Chebyshev distance away from it (e.g., in Fig. 2, a force inflicted on node n_1 by node n_3). A node i is not aware of any other agents more than R_C away from it and, hence, their presence has no effect on node i's actions. Let us define $f^0_{i,j}$ as follows:

Table 1 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

$$F^0_{i,j} = F_0 \quad \text{for } 0 < d_{i,j} \leqslant d_{th} \tag{4}$$

where d_{ij} is the Euclidean distance between n_i and n_j which are in the same logical cell, d_{th} is the dimension of the logical cell, and 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:

$$\sum_{j \in D^0_i} f^0_{i,j} = \sum_{j \in D^0_i} F_0 \tag{5}$$

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

Similarly, f^1_{ik} can be defined as:

$$F^1_{i,k} = \gamma(d_{th} - d_{ik}) \quad \text{for } d_{th} < d_{ik} < R_c \tag{6}$$

where d_{ik} is the Euclidean distance between n_i and its neighbor n_k (one Chebyshev distance away), γ_i is the expected node degree which is a function of mean node degree, as presented in Urrea et al. [19], 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:

$$\sum_{k \in D^1_i} f^1_{i,k} = \sum_{k \in D^1_i} \gamma_i(d_{th} - d_{ik}) \tag{7}$$

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

To encourage the dispersion of nodes, we assign a large value to the force from the neighbors located in D^0_i (i.e., F_0 in Eq. (5)) than the total force exerted by the neighbors in D^1_i (i.e., f^1_{ik} from Eq. (6)):

$$F_0 > \sum_{k \in D_i^1} f_{i,k}^1 \quad (8)$$

In NSEG, player 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:

$$U_i(S) = \begin{cases} \sum_{j \in D_i^0} F_o + \sum_{k \in D_i^1} f_{i,k}^1 & \text{if } D_i^0 \cup D_i^1 \neq \varnothing \\ \mathscr{F}_{\max} & \text{otherwise} \end{cases} \quad (9)$$

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

$$\mathscr{F}_{\max} = n \times F_0 \quad (10)$$

where n is the total number of nodes in the systems.

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. (9).

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.

Evaluation of player's current

After moving to a new location, n_i computes $u_i(\boldsymbol{s})$ defined in Eq. (9) 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 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, i's mixed strategy profile is defined as:

$$\boldsymbol{\sigma}_i = (\sigma_i(S_0), \sigma_i(S_1), \ldots, \sigma_i(S_8)) \quad (11)$$

where $\sigma_i(s_k)$ represents a probability of strategy k being played. The mixed strategy profile σ_i reflects i's preferences over its next possible locations by assigning positive probability only to these locations that may improve its payoff. Fig. 3 shows the probability state transition diagram for a node in state s_4. In Fig. 3, the probability of each transition is assigned by the FGA locally run by this node.

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

$$\sigma_i(S_4) > (1 - \epsilon) \quad (12)$$

where ε is a small positive number.

If Eq. (12) 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 the 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 j, we will refer to $\bar{u}$ and $\bar{\sigma}$ for each player located in the logical cell in which 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 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.

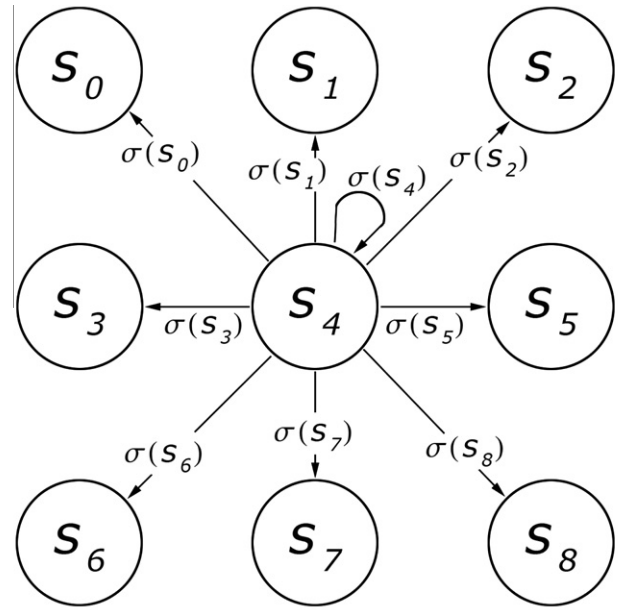

Fig. 3 The probability state transition derived from a stochastic vector σ_i.

Spatial game setup

If player i decides to move to a new location using Eq. (12), she gathers $\bar{u}_j$ and $\bar{\sigma}_j$ for all $j \in \bar{I}$. Node i constructs its payoff matrix M_i with an entry for each possible strategy profile $\boldsymbol{s}$ that can arise among members $\bar{I}$. Each element of M_i reflects the goodness of i's next location over possible combinations of all other players' strategies. After that, i computes its expected payoff for this game as:

$$U_i(\bar{\boldsymbol{\sigma}}) = \sum_{s \in S} (\Pi_{j \in I} \sigma_j(s_j)) u_i(s) \quad (13)$$

Expected payoff $u_i(\bar{\boldsymbol{\sigma}})$ is an estimation of what the total forces inflicted on player i will be if she plays her mixed strategy profile $\bar{\sigma}_j$ against her opponents' strategy profiles $\bar{\boldsymbol{\sigma}}_{i-1}$. As such, $u_i(\bar{\boldsymbol{\sigma}})$ is an indication of 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. (9). We formalize this notion in the lemma below.

Lemma 1. *Player i's mixed strategy profile σ_i obtained from FGA may not reflect the best new location(s) for player i.*

Proof. Let us consider a case where set D_i^1 (Eq. (7)) consists of equally distanced neighbors from *i*. Suppose also that there is a node m in the same cell as *i*. Consequently, our FGA will decide that *i* should move into one of its neighboring cells because of m. In this setting, 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 the forces inflicted on a player by its neighbors (Eqs. (7) and (5)).

It is clear that FGA cannot distinguish the optimal choice among the 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 *k* (i.e., $\sigma_i(s_k) = 1$) while a better strategy *j* exists (requiring to move to cell *j*) with $u_j(s) < u_k(s)$ (Eq. (9)). □

Lemma 1 shows that player *i*'s mixed strategy profile may not be the most profitable strategy in her proximity. Therefore, player i should utilize additional information about its neighbors' payoffs and mixed strategy profiles (Eqs. (9) and (11)) to determine if locations obtained from FGA are indeed the best and what her next location should be. Hence, player *i* sets up a spatial game among her and all other members of $\bar{I}$ to compute her expected payoff from this interaction (Eq. (13)).

Let us consider the neighboring cells for player *i*. Recall that each neighboring cell $j \in \bar{I}$ 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 *i*'s neighboring cell such that the forces inflicted on it is the minimum.

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

$$\mathscr{C}_{\min} + \alpha < u_i(\bar{\sigma}) \tag{14}$$

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

Therefore, as a direct result of Lemma 1 and Eq. (14), we can state the following corollaries which govern decisions of our NSEG.

Corollary 1. *If the expected improvement for player i resulting from moving into a location obtained by FGA is worse than moving into* $\mathscr{C}_{\min}$ *(Eq. (14)), player i's next position should be* $\mathscr{C}_{\min}$.

Corollary 2. *If the expected improvement for player i obtained from FGA is better than (or the same as) moving into* $\mathscr{C}_{\min}$ *(Eq. (14)), player i selects her next location according to her mixed strategy profile* σ_i.

Analysis of NSEG convergence

In NSEG, a movement decision for node *i* is based on the outcome of the locally run FGA and the spatial game set up among *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 toward a uniform node distribution as a 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 *i* is to become isolated from the other nodes, in which case $u_i = \mathscr{F}_{\max}$ and player *i* cannot interact with any other nodes to improve its payoff. From the entire network perspective, the disconnected node adds little to the network performance and can be considered a lost resource. Eq. (9) guarantees that no individual node chooses a new location which will result in becoming disconnected.

Since an additional node located in the same cell as player *i* (i.e., $D_i^1 = 1$) affects *i*'s payoff adversely to the greater degree than the distant located neighbors (i.e., members of D_i^1), player i prefers to be the only occupant of its current logical cell. Multiple nodes in a single cell are also undesirable from the network perspective, as the 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^* be a strategy for a non-isolated player *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 $S^* = (S_1^*, S_2^*, \ldots, s_n^*)$ 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. (9) and (13). Let an invading sub-optimal strategy $S_j' \neq s^*_{\text{opt}}$ be played by player *j*. Then s^*_{opt} is ESS if the following condition holds:

$$U(s^*_{\text{opt}}, s^*_{\text{opt}}) < u(s_j', s^*_{\text{opt}}) \tag{15}$$

where an optimal strategy s^*_{opt} can be played by any $i \in I \setminus j$. The following

lemma shows that a strategy s^*_{opt} is evolutionary stable and, hence, no strategy can invade a population playing s^*.

Lemma 2. *A strategy* s^*_{opt} *is evolutionary stable.*

Proof. There are two cases in which player *j*'s strategy S_j' may differ from s^*_{opt}. In one of them, strategy S_j' represents a case where player *j* is disconnected and, as stated in Eq. (9), receives payoff $\mathscr{F}_{\max}$, which is strictly greater than any possible $u(s^*_{\text{opt}}, s^*_{\text{opt}})$. 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. (8), $u(s^*_{\text{opt}}, s^*_{\text{opt}}) < u(s_j', s^*_{\text{opt}})$. Consequently, in both cases in which $s_j' \neq s^*_{\text{opt}}$ invades a population playing strategy s^*_{opt} (i.e., a population playing a strategy profile s^*), first condition of ESS (Eq. (15)) holds, establishing that s^*_{opt} is an ESS. □

Lemma 2 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 2 in the following corollary.

Corollary 3. *A strategy s^* represents a stable network topology that will maintain its stability since no node has any incentive to change its current position.*

Experimental results

We implemented NSEG using Java programming language. Our software implementation consists of more than 3,000 lines of algorithmic Java code. 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. This initial distribution represents realistic situations where nodes enter the terrain from a common entry point (e.g., starting node deployment into an earthquake area from a single location) compared to random or any other types of initial distributions we see in the literature. Each simulation experiment was repeated 10–15 times and the results were averaged to reduce the noise in the observations.

The snapshot in Fig. 4 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 are occupied by a population of 80 nodes (i.e., $n = 80$). 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. 5. 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 a notable node concentration in the area of initial deployment.

A stable node distribution after running NSEG for 60 time units is shown in Fig. 6. 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. 6 represents the stable state for this population. As presented in Lemma 2 and Corollary 3, 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 2 and Corollary 3.

Network area coverage (NAC) is an important metric of our NSEG effectiveness. NAC is defined as the ratio of the area covered by the communication ranges of all nodes and the total geographical area. NAC value of 1 implies that the entire area is covered. Fig. 7 shows the improvement of NAC and the 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 (i.e., NAC = 1) after approximately 40 iterations of NSEG. However, the number of occupied cells keeps increasing for another 20 steps up to a point where each cell becomes occupied by at most one node. We can derive two conclusions from this observation: (i) for the deployment of 100×100 unit square area divided into 10×10 logical cells,

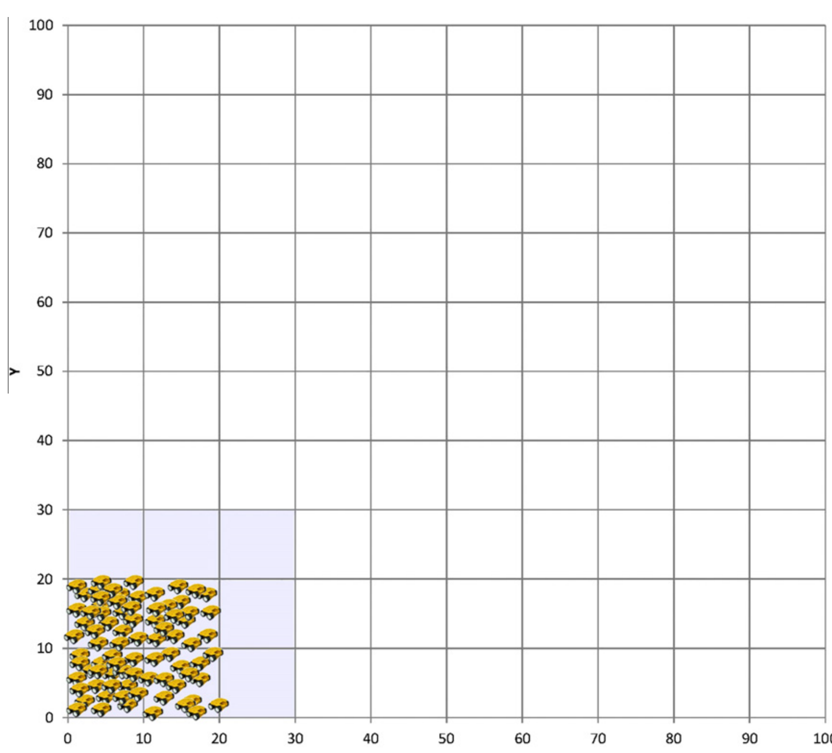

Fig. 4 The probability state transition derived from a stochastic vector σ_i.

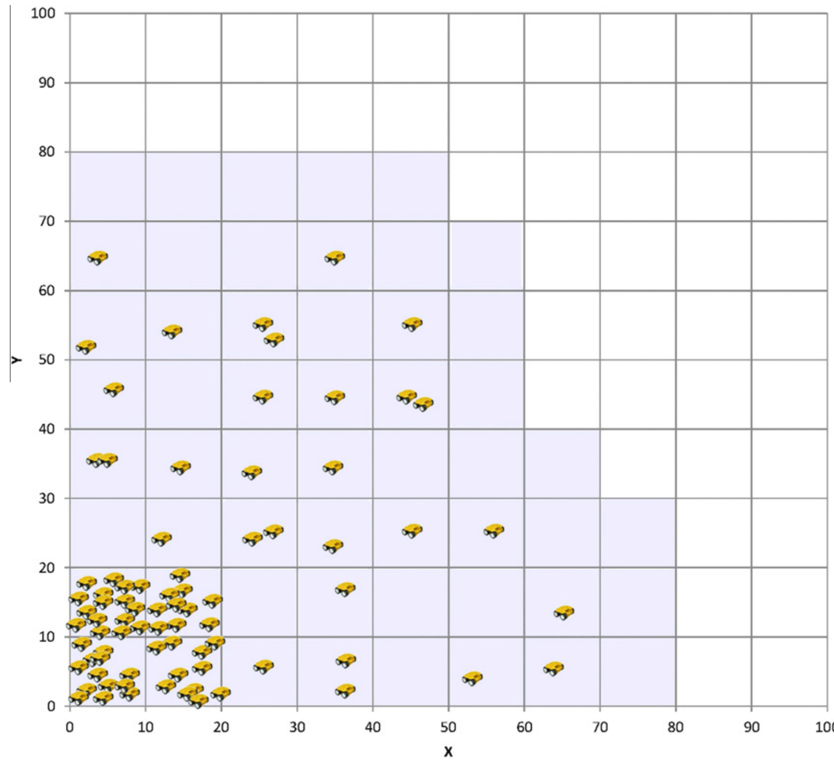

Fig. 5 Node distribution obtained by 80 autonomous nodes running NSEG for 10 steps.

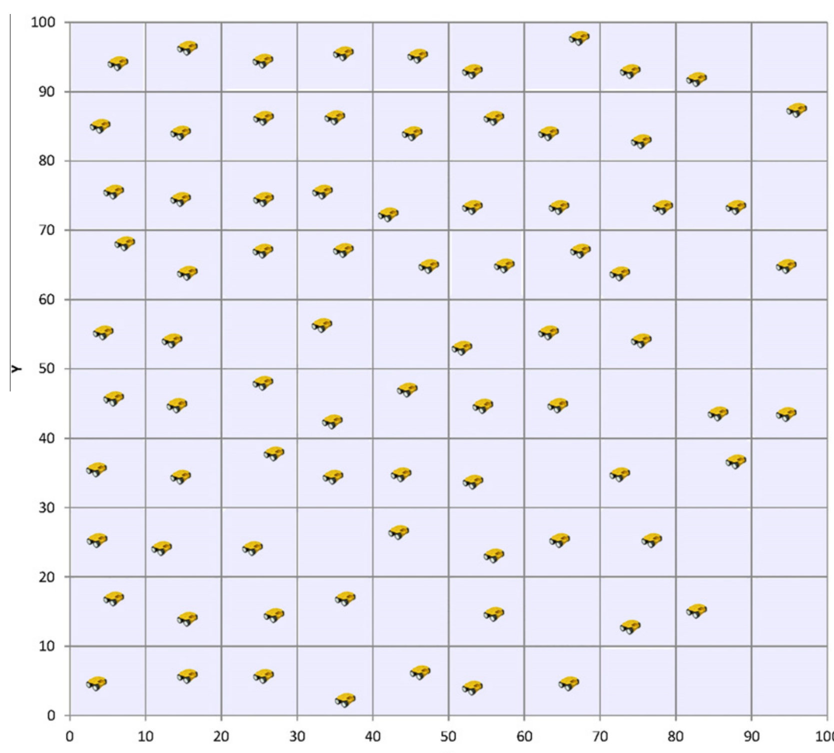

Fig. 6 Stable node distribution obtained by 80 autonomous nodes after running NSEG for 60 steps.

80 nodes are sufficient to achieve NAC = 1, and (ii) even when the goal of the total area coverage is achieved, the network topology do not stabilize until the optimal strategy profile s^* is realized by the entire network.

Fig. 8 shows the improvement in NAC for networks with different number of mobile nodes. We can see in this figure that for larger values of n, the network requires more time to achieve its maximal terrain coverage since there are more

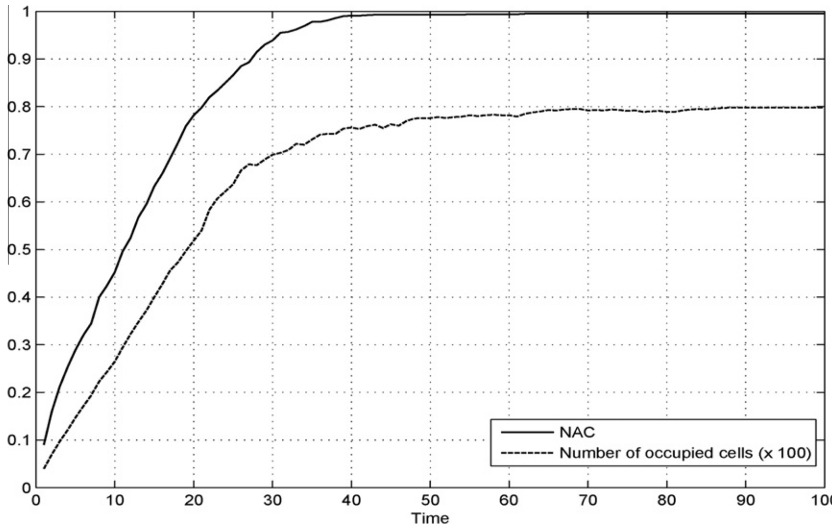

Fig. 7 NAC and the number of occupied logical cells obtained by 80 autonomous nodes running NSEG.

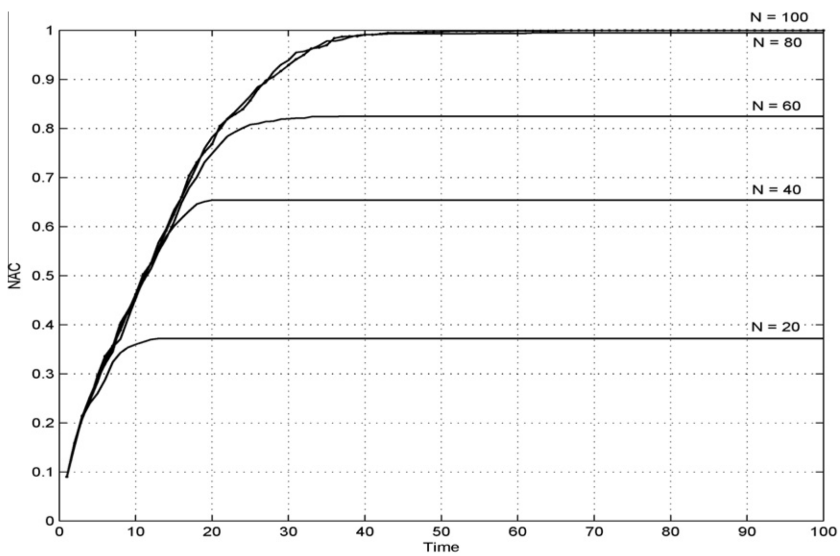

Fig. 8 Improvement of NAC by NSEG in different network sizes (n = 20 to 100).

nodes to disperse from the same small initial deployment area. However, maximal NAC achieved by NSEG increases notably as the number of nodes deployed in the same geographical area increases. It can also be seen in Fig. 8 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). This observation allows us to project the performance of NSEG in a larger area than 100×100 unit squares or in the situations where the logical cells are smaller than selected for our experiments. In Fig. 8, 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. 4–7.

Our simulation results show that NSEG can be effective in providing a satisfactory level of area coverage with near uniform node distribution while utilizing only the local information by each autonomous agent. Since our model does not require a global coordination, a priori knowledge of a deployment environment, or a strict synchronization among the nodes, it presents an easily scalable solution for networks composed of self-positioning autonomous nodes.

Concluding remarks

We introduce a new approach for self-spreading autonomous nodes over an unknown geographical territory by combining a force-based genetic algorithm (FGA), traditional game theory and evolutionary game theory. Our node spreading evolutionary game (NSEG) runs at each mobile node making independent movement decisions based on the outcome of a locally run FGA and the spatial game set up among itself and its neighbors. 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 toward a uniform and stable node distribution.

Our FGA only takes into account the current positions 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. We present a formal analysis of our NSEG and prove that the evolutionary stable state 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.

Since NSEG does not require global network information nor strict synchronization among the nodes, future extension of this research will focus on real-life applications of NSEG to the node spreading class of problems in both military and commercial tasks.

References

[1] Howard A, Mataric MJ, Sukhatme GS. Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem. Distrib Auto Robot Syst 2002;5:299–308.

[2] Leonard NE, Fiorelli E. Virtual leaders, artificial potential and coordinated control of groups. Proceedings of the 40th IEEE Conference on Decision and Control 2001. p. 2968–73.

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

[4] Xi W, Tan X, Baras JS. Gibbs sampler based control of autonomous vehicle swarms in the presence of sensor errors. Conference on Decision and Control, 2006. p. 5084–90.

[5] Cortes J, Martinez S, Karatas T, Bullo F. Coverage control for mobile sensing networks. IEEE Trans Robot Autom 2004;20(2):243–55.

[6] Jadbabaie A, Lin J, Morse AS. Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans Automat Contr 2003;48(6):988–1001.

[7] Kusyk J, Urrea E, Sahin CS, Uyar MU. Self Spreading Nodes Using Potential Games and Genetic Algorithms. IEEE Sarnoff Symp, 2010. p. 1–5.

[8] Kusyk J, Urrea E, Sahin CS, Uyar MU. Resilient node self-positioning methods for MANETs based on game theory and genetic algorithms. In: IEEE Military Communications Conference (MILCOM); 2010.

[9] Kusyk J, Uyar MU, Urrea E, Fecko M, Samtani S. Applications of game theory to mobile ad hoc networks: node spreading potential game. IEEE Sarnoff Symposium, 2009. p. 1–5.

[10] Kusyk J, Uyar MU, Urrea E, Sahin CS, Fecko M, Samtani S. Efficient node distribution techniques in mobile ad hoc networks using game theory. IEEE Military Communications Conference (MILCOM), 2009.

[11] Seredynski M, Bouvry P. Evolutionary game theoretical analysis of reputation-based Packet forwarding in Mivilian mobile ad hoc networks. IEEE International Symposium on Parallel and Distributed Processing, 2009.

[12] Fischer S, Vocking B. Evolutionary game theory with applications to adaptive routing. European Conference on Complex Systems (ECCS), 2005. p. 104.

[13] Wang B, Liu K, Clancy TC. Evolutionary game framework for behavior dynamics in cooperative spectrum sensing. IEEE Global Telecommun Conf (GLOBECOM), 2008.

[14] Ahn C, Ramakrishna RS. A genetic algorithm for shortest path routing problem and the sizing of populations. IEEE Trans Evol Comput 2002;6(6):566–79.

[15] Barolli L, Koyama A, Shiratori N. 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), 2003. p. 175.

[16] Sahin CS, Urrea E, Uyar MU, Conner M, Hokelek I, Bertoli G, et al. Genetic algorithms for self-spreading nodes in MANETs. Proceedings of the 10th annual conference on Genetic and evolutionary computation (GECCO), 2008. p. 1141–42.

[17] Sahin CS, Urrea E, Uyar MU, Conner M, Hokelek I, Bertoli G, et al. Uniform distribution of mobile agents using genetic algorithms for military applications in MANETs. IEEE Military Communications Conference (MILCOM). 2008 November; p. 1–7.

[18] Sahin CS, Urrea E, Uyar MU, Conner M, Bertoli G, Pizzo C. 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 2009.

[19] Urrea E, Sahin CS, Hokelek I, Uyar MU, Conner M, Bertoli G, et al. Bio-inspired topology control for knowledge sharing mobile agents. Ad Hoc Netw 2009;7(4):677–89.

[20] Fudenberg D, Tirole J. Game theory. The MIT Press; 1991.

[21] MacKenzie AB, DeSilva LA. Game theory for wireless engineers. 1st ed. Morgan and Claypool Publishers; 2006.

[22] Smith JM. Evolution and the theory of games. Cambridge University Press; 1982.

[23] Weibull JW. Evolutionary game theory. The MIT Press; 1997.

[24] Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence. Cambridge, MA, USA: MIT Press; 1992.

[25] Mitchell M. An introduction to genetic algorithms. Cambridge, MA, USA: MIT Press; 1998.

[26] Lewontin RC. Evolution and the theory of games. J Theoret Biol 1961;1:382–403.

[27] Smith JM, Price GR. The logic of animal conflict. Nature 1973.

[28] Taylor PD, Jonker LB. Evolutionary stable strategies and game dynamics. Math Biosci 1978;16:76–83.

[29] Nowak MA, May RM. The spatial dilemmas of evolution. Int J Bifurcat Chaos 1993;3(1):35–78.

[30] Holland JH. Adaptation in natural and artificial systems. University of Michigan Press; 1975.

[31] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. Int J Robot Res 1986;5(1):90–8.

[32] Heo N, Varshney PK. A distributed self spreading algorithm for mobile wireless sensor networks. IEEE Wireless Commun Network (WCNC) 2003;3(1):1597–602.