Increasing the performance of a Mobile Ad-hoc Network using a game-theoretic approach to drone positioning*

S. Blakeway[1], *D. V. Gromov*[2], *E. V. Gromova*[2], *A. S. Kirpichnikova*[3], *T. M. Plekhanova*[2]

[1] Wrexham Glyndŵr University, Mold Road, Wrexham, LL11 2AW, Great Britain
[2] St. Petersburg State University, 7–9, Universitetskaya nab., St. Petersburg,
   199034, Russian Federation
[3] University of Stirling, Stirling, FK9 4LA, Scotland, Great Britain

We describe a novel game-theoretic formulation of the optimal mobile agents' placement problem which arises in the context of Mobile Ad-hoc Networks (MANETs). This problem is modelled as a sequential multistage game. The definitions of both the Nash equilibrium and cooperative solution are given. A modification was proposed to ensure the existence of a Nash equilibrium. A modelling environment for the analysis of different strategies of the players was developed in MATLAB. The programme generates various game situations and determines each player move by solving respective optimisation problems. Using the developed environment, two specific game scenarios were considered in detail. The proposed novel algorithm was implemented and tested using Network Simulator 3 (NS-3). The results show that the proposed novel algorithm increases network performance by using game theory principles and techniques.

Keywords: MANET, dynamic games, multistage games, drone placement, graphs, Nash equilibria, NS-3.

1. Introduction. Recently, there has been a growing interest in the qualitative analysis and performance optimisation of Mobile Ad-hoc Networks (MANETs). A MANETs is formed by a collection of wireless nodes communicating directly with other wireless nodes within their transmission range. To facilitate long distance communication, i. e. to nodes outside of their transmission range, other nodes will forward the packet towards the intended destination, thus these intermediate nodes along the path take on the role of routers. A MANETs can operate independently of other networks and does not require a predefined infrastructure [1]. MANETs play an increasingly important role in data communication networks and can be used for many applications, such as disaster recovery after a natural catastrophe, for instance after an earth quake. The quick deployment time of a MANETs makes it an ideal solution for search-and-rescue operations [2, 3] where the existing communications infrastructure has been compromised or damaged. One of the biggest problems associated with the operation of a MANETs (and other wireless broadcast systems) is the need to share/use limited resources for the transmission of radio waves. Inappropriate use of the resources can cause a severe degradation in network performance

https://doi.org/10.21638/11702/spbu10.2019.102

[4], another issue is the formation of communication paths because of the mobility of each of the nodes within the network.

When the number of nodes wanting to transmit data increases, some of the routing paths become congested and performance drops [5, 6]. The amount of data could also large, for example, in the case of video streaming [7]. In some cases obstacles on the ground (lakes, buildings, etc.) can cause link breakages. To mitigate the congestion of a given link, or to fix a link breakage, drones can be used as an intermediate node to facilitate the forwarding of the traffic.

Recently, utilization of teams of Unmanned Aerial Vehicles (UAVs) became extremely po-pular because they allow to extend the operational scope and significantly reduce response time [8]. Introducing mobile agents (drones) and finding the best possible locations for their placement is the focus of this research. Recent research suggests forming networks of UAVs using star topology, or uniform coverage [8, 9]. In most cases, the number of drones is rather limited so they should be placed at strategic locations to maximally increase the performance of the network [10]. Typically, research that discusses MANETs uses all available nodes to form one big communications network; however, there are situations when there may be several groups of nodes which are deployed to solve their specific tasks. These grouping of nodes may or may not be able to communicate with each other; the latter case corresponds to the situation when the nodes from different groups use different frequency ranges [11].

We address the described problem in a decentralized manner. That is, we assume that each group of nodes has a single control centre which is in possession of a single drone. The goal of the control centre is to strategically place the drone to maximise the performance of the subnetwork formed by the nodes. We assume that the number of available drones doesn't allow full coverage of the restricted zone.

It turns out that game theory lends itself perfectly well to addressing the described problem. Game theory is a powerful tool to study situations of sharing limited resources, and it deals with finding the best actions for individual decision makers (players) and for finding the best available outcome [12, 13]. Using the game-theoretic methods one can explicitly design and analyse strategic choices and model the decision-making process for the player per their own interests. Much of the research conducted in the application of the game theory in the area of MANETs is related to malicious or selfish node detection [14–18]. Existing applications also includes looking for large communities in the networks [19–21], and investigation of cooperative games for various network games [22–24]. In this paper, we take a different approach and apply game theory to a special class of network optimisation problem as described below. Note that there are several interesting results on games theory applied to networks (e. g., [25–27]); however, this particular problem statement appears to be novel thus opening wide opportunities for further research.

MATLAB was used to code the game theory algorithm to determine the most strategic locations of the players drone. Results show that our algorithm can determine several possible strategic locations for the placement of the drones. In addition, we were interested how these placements would increase the overall performance of the MANETs, thus, Network Simulator 3 (NS-3) was used to simulate a realistic network environment. This research extends the published work in [28, 29].

The paper is structured as follows: in section 2, we give a formal description of the considered objects and introduce the mathematical notation that will be used thereafter. In section 3 we present the game-theoretic formulation of the drone placement problem. Section 4 contains numerical examples which were coded in MATLAB. Section 5 discusses

the implementation of the simulation and an analysis of the results, while in section 4 we draw conclusions and outline the directions of future research.

2. Problem statement

2.1. Two types of communication networks. We consider the set $\mathcal{N} = \{1, ..., N\}$ of N players, each player $i \in \mathcal{N}$ has a non-empty set $\mathcal{M}_i = \{1, ..., M_i\}$ of agents. Each agent $m_i^j \in \mathcal{M}_i$, $i \in \mathcal{N}$, is characterized by a pair of coordinates (x_i^j, y_i^j), $\mathcal{M} = \bigcup_{i \in N} \mathcal{M}_i$ is the set of all agents.

We assume that the agents can be located at the vertices of a uniform tiling of a connected subset of the Euclidean plane $C \subset \mathbb{R}^2$, for example see Fig. 1 (graph on the left) which depicts the location of the agents at a moment in time for a given player. In practice, one considers three kinds of uniform tiling: the triangular, the Cartesian (integer), and the hexagonal tiling, which are formed by unit equilateral triangles, squares, and hexagons. Henceforth, we will denote the set of all admissible coordinates by $W \subset C$. In the case of the Cartesian grid, the coordinates of admissible points are the pairs of integer, i. e. $W \subset \mathbb{N}^2 \cap C$.
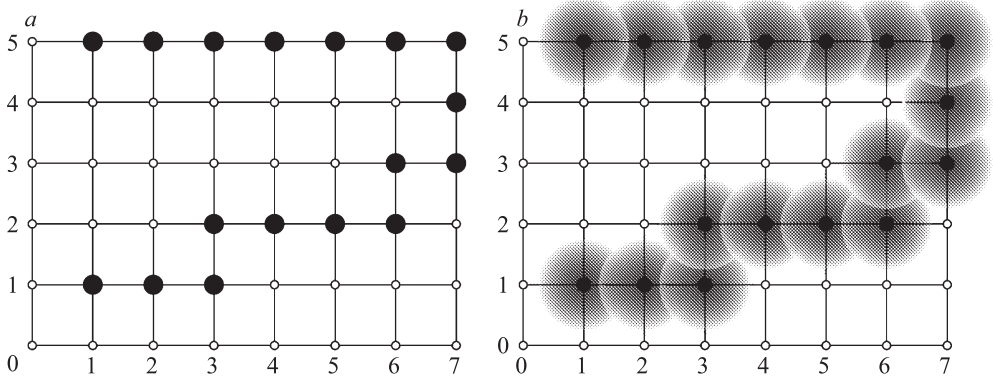

Figure 1. Example of node placement (a) and transmission radius (b)

The agents form a communication network whose structure is determined by the spatial alignment of agents and their transmission capabilities, Fig. 1 depicts the agents' transmission radius as a gradient outer circle, the use of the gradient illustrates that the transmitted signal attenuates with distance. The agent at $(1, 1)$ is in range with agent at $(2, 1)$ and no other agent, however, the agent at position $(3, 2)$ is in range of agents at positions $(3, 1)$ and $(4, 2)$. While the very nature of a MANETs suggests a dynamic topology, we look at a snapshot of the network at a given point in time. Thus, we assume that the communication network formed by the respective agents does not change with time. In the following, we will consider two types of communication structures: separate and joint use of communication infrastructure. We denote these two types of communication networks by **S** (separate) and **J** (joint). Below we consider these two cases in more detail.

S-*network.* The communication network consists of N disjoint graphs $\mathcal{G}_i^{\mathbf{S}} = \left(\mathcal{V}_i^{\mathbf{S}}, \mathcal{E}_i^{\mathbf{S}}\right)$, $i \in \mathcal{N}$, where the sets of vertices $\mathcal{V}_i^{\mathbf{S}} = \{v_i^k = (x_i^k, y_i^k)\}_{k=1}^{M_i}$ are the collections of the Cartesian coordinates of agents of player i; the sets of edges $\mathcal{E}_i^{\mathbf{S}} = \{e_i^{k,s} = \left(v_i^k, v_i^s\right) \in \mathcal{V}_i^{\mathbf{S}} \times \mathcal{V}_i^{\mathbf{S}} : dist\left(v_i^k, v_i^s\right) = 1\}$ are the sets of all pairs of agents for player i, that are at a unit Euclidean distance from each other. Note that the coordinate grid is such that the distance between two neighbouring points is always equal to 1. We assume that each graph

$\mathcal{G}_i^{\mathbf{S}}$ is connected, i. e. there is a connected path between any two agents of a player. Note that there are no connections (links) between the elements of different subgraphs, i. e. the agents of one player do not participate in the transmission between the agents of another player, i. e. the agents of the first player are transmitting at a range of frequencies different to that of the second player.

J-*network*. This case corresponds to a single communication network whose communication structure is modelled by the graph $\mathcal{G}^{\mathbf{J}} = \left(\mathcal{V}^J, \mathcal{E}^{\mathbf{J}} \right)$, where the set of vertices $\mathcal{V}^{\mathbf{J}} = \left\{ v_i^k = \left(x_i^k, y_i^k \right) \right\}$, $i \in \mathcal{N}$, $v_i^k \in \mathcal{M}_i$, is the collection of the Cartesian coordinates of all agents; the set of edges $\mathcal{E}^{\mathbf{J}} = \{ e^{k,s} = \left(v_i^k, v_j^s \right) \in \mathcal{V}^{\mathbf{J}} \times \mathcal{V}^{\mathbf{J}} : dist \left(v_i^k, v_j^s \right) = 1 \}$ is the set of all pairs of agents that are at a unit Euclidean distance from each other. As in the previous case, we assume that the graph $\mathcal{G}^{\mathbf{J}}$ is connected, i. e. there is a connected path between any two agents.

Note that in both cases the coordinates of agents of two different players can coincide. While in the case of the **S**-network this would mean that the graphs may overlap (they can be considered to lie in different layers), for the **J**-network this would imply that there is no direct connection between two agents located in the same point. This can be relaxed by defining the set of edges to be the set of all pairs of agents at the distance less or equal to 1, i. e. $\widetilde{\mathcal{E}}^{\mathbf{J}} = \{ e^{k,s} = \left(v_i^k, v_j^s \right) \in \mathcal{V}^{\mathbf{J}} \times \mathcal{V}^{\mathbf{J}} : dist \left(v_i^k, v_j^s \right) \leqslant 1 \}$. Note, however, that this is not important in the context of our research as we are interested in determining shortest paths between different nodes, which obviously must contain only non-overlapping nodes.

Note also that the choice of the grid can influence the connectivity of the respective graphs and the related characteristics. While any agent located on a hexagonal grid can have at most adjoin 3 edges, for the triangular grid this number can be up to 6.

2.2. Graph theoretic ingredients. Before proceeding to the game-theoretic formulation of the problem we present a couple of facts from graph theory which will be used hereafter.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph and m be an agent. We define the union $\widetilde{\mathcal{G}} = \mathcal{G} \cup \{m\}$ to be a new graph with an extended set of vertices $\widetilde{\mathcal{V}} = \mathcal{V} \cup \{m\}$ and the accordingly adjusted set of edges $\widetilde{\mathcal{E}}$. The union operation can be obviously extended to the case $\mathcal{G} \cup M$, where M is a set of agents $M = \{m_i\}_{i=1}^k$. Note that $\widetilde{\mathcal{G}}$ can be disconnected.

The location of the agents is determined by the application or the scenario, for example, consider a search and rescue operation, where a person was reported missing in an area with lakes. The search operation might involve firstly coordinating the searching of the circumference of the lake as depicted in Fig. 2.

The **diameter of the graph** [30] $\mathcal{G}$, denoted by $D(\mathcal{G})$, is the maximum among all shortest paths between the agents in the graph, $D(\mathcal{G}) = \max\limits_{(v_i, v_j) \in \mathcal{V} \times \mathcal{V}} d(v_i, v_j)$, where $d(v_i, v_j)$ is the graph distance between two vertices v_i and v_j which is defined as the minimum length of the paths connecting them. If no such path exists, $d(v_i, v_j) = \infty$, in our scenario (see Fig. 2) the maximum path among all shortest paths between the agents is between the agents located at $(1, 1)$ and $(1, 5)$, for convenience we have assigned S (source) and D (destination) to illustrate a communications path.

Since each graph $\mathcal{G}_i^{\mathbf{S}}$ is connected, for the integer grid the diameter can be roughly estimated as

$$2 \left(\lfloor \sqrt{M_i} \rfloor - 1 \right) \leqslant D \left(\mathcal{G}_i^{\mathbf{S}} \right) \leqslant M_i - 1.$$

Obviously, while the upper bound does not depend on the type of the grid, the lower bound does. Namely, it is minimal for the triangular grid and maximal for the hexagonal one.
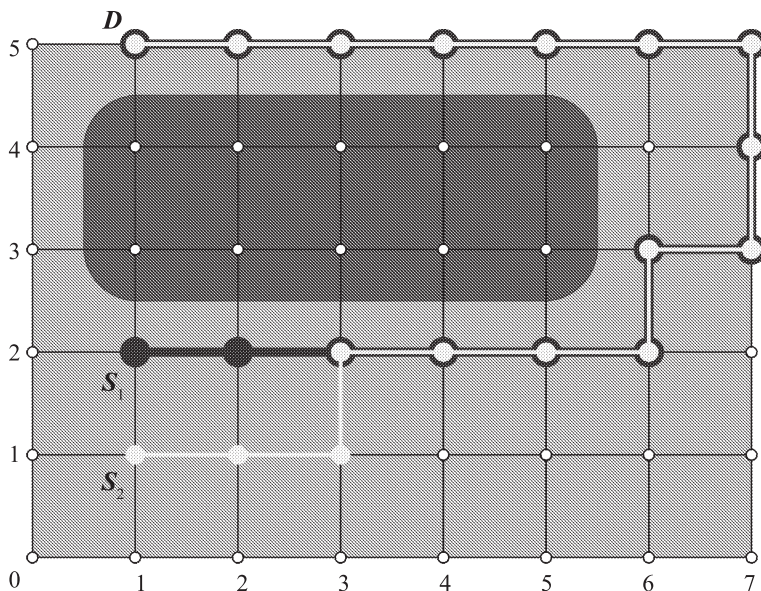
Figure 2. Example 1 with one lake and two players

Furthermore, let $\mathbf{V} = \{\mathcal{V}_i\}$ be a disjoint partition of the set of vertices $\mathcal{V}$ of the graph $\mathcal{G}$, i. e. $\mathcal{V} = \bigcup_i \mathcal{V}_i$ and $\mathcal{V}_i \bigcap \mathcal{V}_j = \varnothing$ for all $i, j \in \mathcal{N}, \ i \neq j$. We define the diameter of the graph $\mathcal{G}$ with respect to the element $\mathcal{V}_i$ of the partition $\mathbf{V}$ as

$$D(\mathcal{G}, \mathcal{V}_i) = \max_{(v_i, v_j) \in \mathcal{V}_i \times \mathcal{V}_i} d(v_i, v_j).$$

That is to say, when computing $D(\mathcal{G}, \mathcal{V}_i)$ we consider only the paths between the pairs of vertices from $\mathcal{V}_i$. However, the intermediate vertices in the respective paths are not required to be in $\mathcal{V}_i$.

2.3. Mobile agent: drone. Each player has a mobile agent (drone) q_i at their disposal. The drone is placed at any admissible point in W. While all the agents have fixed coordinates, the position of the moving agent $q_i, i \in \mathcal{N}$, can be changed during the game. The link between the moving agent and another agent is defined similarly to the links between the stationary agents. The moving agent q_i can establish a link with any other moving agent q_j and with any stationary agent $v_i^p, \ v_i^p \in \mathcal{M}_i, i \in \mathcal{N}$.

At the initial time, all drones are assumed to be located at some initial position q^0. We also assume that the Euclidean distance between q^0 and the agents is greater than 1, i. e. $dist(q^0, v_i^k) > 1 \ \forall i \in \mathcal{N}, v_i^k \in \mathcal{M}_i$. That is to say, the initial position of the drones is such that they cannot establish communication with the agents of any other player.

Finally, we denote the set of all drones by $Q = \{q_i\}_{i=1}^N$ and the set of all drones except the i-th one by $Q_{-i} = Q \backslash \{q_i\}$.

3. Game formulation

3.1. Optimisation problem. At each step the i-th player aims at placing its moving agent q_i such that it minimises the diameter of the respective graph computed for all player's agents. Note that the drones themselves are not considered when computing the diameter of the graph as they are not assumed to be the sources of any useful information, but merely transmit the information packages between the agents. In this statement, we

assume that each drone can be used by the agents of any player. The set of admissible locations of the ith drone, $W_i^* = W \backslash Q$, is introduced to avoid the collision of two drones.

For any type of communication structure the i-th player solves the following optimisation problem:

$$\bar{q}_i = \arg \min_{q_i \in W_i^*} D(\mathcal{G}_i^*, \mathcal{V}_i), \tag{1}$$

where $\mathcal{G}^*$ is the extended graph: for the case of the **S**-network, $\mathcal{G}_i^* = \mathcal{G}_i^{\mathbf{S}} \cup Q_{-i} \cup \{q_i\}$, for the **J**-network, $\mathcal{G}_i^* = \mathcal{G}^{\mathbf{J}} \cup Q_{-i} \cup \{q_i\}$, respectively.

Accordingly, the payoff (utility) function of the i-th player is defined as the difference between the diameter of the respective graph before placing the drone and afterwards:

$$H_i\left(\mathcal{G}, Q\right) = D\left(\mathcal{G}, \mathcal{V}_i\right) - D\left(\mathcal{G}_i^*, \mathcal{V}_i\right) \geqslant 0, \tag{2}$$

note that the payoff function is always nonnegative. We will modify this definition for the multi-staged game in the subsequent section.

3.2. Game aims and strategic (normal) form. The idea of the game is that each of the players aims to place its moving agent such that it minimises the maximal distance between the player's agents while taking into account the existing communications infrastructure. The degree by which the player minimises the mentioned distance is captured by the payoff function (2). Thus the goal of the player is to maximise its payoff function. The game finishes if none of the players can maximise further the respective payoff function.

The game is now formulated in normal form (see [31, 32]), i. e. the game is a triple $\Gamma = \{\mathcal{N}, \mathbf{S}, H\}$, where $\mathcal{N}$ is the set of players; $\mathbf{S} = \prod_{i \in \mathcal{N}} S_i$ is a Cartesian product of strategy sets S_i. The set of available strategies for player i is S_i; $H = \{H_i\}_{i=1,N}$. Here H_i is the utility (payoff) function for player i (see (2)).

The size of the action set of each player is limited from above: $|S_i| \leqslant |W| - N + 1$. This estimate shows that our game is discrete and finite. However, the above estimate can be greatly improved. For instance, for the rectangular grid and **S**-network the number of all meaningful actions satisfies

$$4\lfloor \sqrt{M_i} - N + 1 \rfloor \leqslant |S_i| \leqslant 2\left(M_i + 1\right),$$

when computing the lower bound we considered the requirement that two drones cannot occupy the same location.

3.3. Multistage game. The formulated game is a game with perfect information. By the latter, we mean that all the players have all the required information about the current and the previous network states, and all the elements of the game are common knowledge. Furthermore, due to the obvious restrictions we confine ourselves to the class of pure strategies.

Below we concentrate on the sequential version of the game, rather than simultaneous. In a *sequential game*, the players make their decisions in a certain, a priori fixed order [31]. Thus, the order in which players choose their actions is a crucial parameter of the game. Intuitively, the player that has the final decision has an advantage over the other players.

Let $\sigma : \mathcal{N} \to \mathcal{N}$, be a bijective map (permutation of $\mathcal{N}$, we assume it is random) defining the sequence of moves. We will refer to σ as the move sequence. At each stage, the players make their choices per this sequence, i. e. first moves the player $\sigma(1)$, then $\sigma(2)$ and so forth. Each player places its drone in order to minimise the diameter of the respective graph, i. e. solves problem (1).

Initially, all drones are in the starting location $Q^0 = \{q^0, ..., q^0\}$. When the ith player moves the set Q updates: $Q = Q_{-i} \cup \{q_i\}$, where q_i is the solution of the respective optimisation problem (1). Note that the decisions taken by subsequent players may change the payoff function of the player that made its move before. Therefore, the final value of the payoff function is computed at the end of the stage, when all players have made their moves. Each stage consists of all players making their moves according to sequence σ.

3.4. Solutions and equilibria. We are interested in two particular types of solutions to the considered game: a cooperative solution and a Nash equilibrium solution. Below, we give a formal definition of these concepts for the case of the **S**-network. All formulated results will hold, *mutatis mutandis*, for the case of a **J**-network.

Definition 3.1. The solution $Q^{NE} = \{q_i^{NE}\}_{i \in N}$ is said to be the *Nash equilibrium* solution if for any $i \in N$ the following holds:

$$H_i\left(\mathcal{G}, Q^{NE}\right) \geqslant H_i\left(\mathcal{G}, Q_{-i}^{NE}\right),$$

here $Q_{-i}^{NE} = \{q_j^{NE}\}_{j \neq i} \cup \{q_i\}$, $q_i \in W_i^*$. This condition can be reformulated in terms of graph diameters:

$$D\left(\mathcal{G}_i^{\mathbf{S}} \cup Q^{NE}, \mathcal{V}_i\right) \leqslant D\left(\mathcal{G}_i^{\mathbf{S}} \cup Q_{-i}^{NE}, \mathcal{V}_i\right).$$

In plain words, this means that any player cannot improve its payoff function (i. e. decrease the diameter of its graph) by unilaterally changing the position of its drone.

Definition 3.2. The solution Q^C is said to be the cooperative solution if it minimises the sum of all individual payoff functions: $Q^C = \underset{Q \in W^*}{\arg\min} \sum_{i \in \mathcal{N}} D\left(\mathcal{G}_i^{\mathbf{S}} \cup Q, \mathcal{V}_i\right)$, where $W^* = W_1^* \times W_2^* \times ... \times W_N^*$ is the set of all admissible control actions.

Note that the cooperative solution always exists, as follows from the finiteness of the set of actions. Obviously, it can be non-unique. The case of Nash equilibrium is, however, subtler.

While our dynamic game is represented in a strategic (normal) form, because of the presence of sequential decision making, an alternative representation highlighting the sequence of the moves can be used. The sequence of moves is thus represented in a game-tree form, which is called an extensive form. It is known that every finite extensive-form game with perfect information has a pure-strategy Nash equilibrium [33]. However, the existence of the Nash equilibrium does not imply that the game has a finite extensive-form representation.

That is to say, even the Nash equilibrium solution exists, it may happen that it cannot be reached by any sequence of players' moves. Such situation occurs, when the Nash equilibrium solution contains certain configuration of drones, which we will call coherent structures. One typical example of a coherent structure is shown in Fig. 3, which depicts the case of a 3-bridge connecting the agents of three different players (shown as a square, a triangle, and a cross). The respective mobile agents are encircled. The figure presents a fragment of the total communication network, in this figure, Fig. 3, three drones form a bridge, the structure which cannot be reached without some extra coordination between players. When the Nash equilibrium cannot be reached, the game will have a periodic solution. This is similar to the situation observed in the theory of dynamic systems, when a stable limit cycle surrounds an equilibrium point.

One approach to overcome this problem is to restrict the set of admissible players' moves in order to ensure that the resulting game-tree is finite. This achieved by adding an additional restriction to the optimisation problem (1). Namely, the players are to be "benevolent", i. e. when there are two equivalent alternatives, the i-th player chooses the
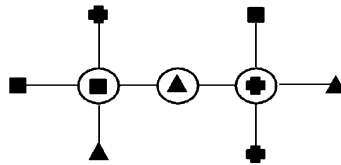
Figure 3. The case of a 3-bridge

option which does not decrease the payoff functions of other players [33]. In our case this amounts to each player moving their own mobile agents so that it does not increase the sub graph diameter of another player. We have the following theorem.

Theorem. *Let all the players be benevolent, i. e. the set of admissible moves be restricted to the* $W_i^* = W \setminus \{Q_{i-1} \cup W_{-i}\}$*, where* W_{-i} *is the set of location that lead to the decrease of the payoffs of the other players. Then the game-tree is finite and a Nash equilibrium exists.*

Note that the Nash equilibrium may not be unique. However, the question of the Nash equilibrium uniqueness is beyond the scope of the paper.

The requirement that the players are to be benevolent is pretty natural as we consider a search and rescue scenario which implies cooperation between different players. However, there can be different scenarios in which the players compete with each other, for example, the authors in [34] considers the use of drones as delivery systems of online goods and explain that these are rapidly becoming a global norm, as corroborated by Amazon's "Prime Air" and Google's "Project Wing" projects, in this work they explain that in this scenario there would not likely be collaboration between the drones and that they would face cyber and physical security challenges.

4. Numerical examples. The simulation of the multi-staged game for the **S**-network has been implemented in MATLAB for two players. For testing purposes the world was restricted by 5×7 and 8×8 rectangular grids. A number of restricted areas were introduced, those which can be thought of as obstacles which prevent player placing agents, i. e. where there might be a lake or building (see Figs 2 and 4). Only mobile agents are allowed to be placed there. For each player, the placement algorithm places a predefined number of nodes randomly on the grid whilst ensuring a connected graph. Then the adjacency matrix $A = [\alpha_{ij}]$, $i, j \in \{1, .., M_i\}$, was calculated such that

$$\alpha_{ij} = \begin{cases} 1, d(\mathcal{V}_i, \mathcal{V}_j) = 1, \\ 0, \text{otherwise}. \end{cases}$$

A modified Dijkstra algorithm was used to calculate the lengths of the paths between the agents for each player [35]. Note that the lengths were computed only for the nodes from a given subset as described in section 2. The length of the longest path was stored as the player's initial diameter. The code then evaluated if there was any competition for the drones position by playing the first stage of the game. After a suitable number of game situations were generated and stored, the game itself was played. The game consisted of a compulsory first stage and then later stages run continuously until none of the players could decrease their diameter.

We present two examples here, one is a game situation with one restricted area (lake), the second illustrates the situation when one of the players finds a new position for their drone, which increases the diameter of another player. Players consider their own diameter optimisation, however, sometimes the use of the second drone can be seen by us as a bridge building process. We should emphasize that the game is non-cooperative, and the players have no intention to build a bridge at any stage.
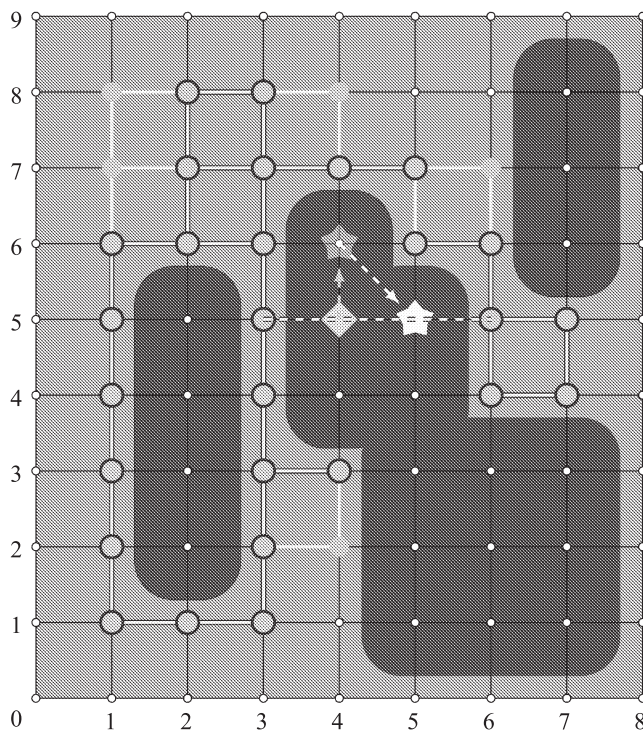
Figure 4. Example of the destruction of an optimal path the 9th strategy from table 3
One of the possible implementations of stage 2: initially drone 1 is at (4,6), drone 2 is at (4,5); at the beginning of the stage 2 drone 1 goes to (5,5) to build a bridge, however, at this time, drone 2 can leave (4,5) and take (4,6) instead (see example 2).

4.1. Example 1. Example 1, which is depicted on Fig. 2, presents a situation, when a bridge is not available at the first stage. The first player moves the drone to $(6, 4)$ (table 1, iteration 1), this reduces the diameter for player 1, to 14, whilst also reducing the diameter for player 2, to 13, which means the payoff after stage 1 is $[2, 2]$. It also leaves the second player with 11 options. However, at stage 2 opportunities for the better use of the second player's drone for the first player are formed (table 1), i. e. two 2-bridges appear $((3, 3) - (3, 4)$ and $(4, 3) - (4, 4))$ (highlighted in bold in table 1). Each of the strategies result in payoff $[16 - 10, 15 - 9] = [6, 6]$. The third 2-bridge is also available $((5, 3) - (5, 4))$, but it only brings the payoff of $[16 - 12, 15 - 11] = [4, 4]$. For the rest of the strategies, both players cannot improve achievements of the first stage.

4.2. Example 2. Example 2 has been generated randomly on the 8×8 field with three lakes (Fig. 4) and it illustrates the non-cooperative nature of this game. The initial diameters for the players are 15 and 12, correspondingly (table 2). The first player has the only minimising position $(4, 6)$, which brings 11 options for the second player. The pay-off function after stage 1 is $[15 - 13, 12 - 11] = [2, 1]$ for all 11 strategies. The next stage begins with the first player's attempt to minimise its diameter by using the second drone. Thus, some of the strategies give multiple options and the total number of strategies becomes 13 (see table 3).

The first 5 strategies do not allow any improvements (as well as strategies 7, 10, and 11) for both players, thus make no changes to the drones' positions. However,

Table 1. **Example 1 strategies and payoffs**

Strategy	Position		Diameters	Payoff	
	drone 1	drone 2	updated	stage 1	stage 2
1	$(6,4)$	$(7,2)$	$[14,13]$	$[2,2]$	$[2,2]$
2	$(6,4) \to (\mathbf{3,4})$	$(\mathbf{3,3})$	$[14 \to 10, 13 \to 9]$	$[2,2]$	$[6,6]$
3	$(6,4) \to (\mathbf{4,4})$	$(\mathbf{4,3})$	$[14 \to 10, 13 \to 9]$	$[2,2]$	$[6,6]$
4	$(6,4) \to (\mathbf{3,3})$	$(\mathbf{3,4})$	$[14 \to 10, 13 \to 9]$	$[2,2]$	$[6,6]$
5	$(6,4) \to (\mathbf{4,3})$	$(\mathbf{4,4})$	$[14 \to 10, 13 \to 9]$	$[2,2]$	$[6,6]$
6	$(6,4)$	$(1,3)$	$[14,13]$	$[2,2]$	$[2,2]$
7	$(6,4)$	$(2,3)$	$[14,13]$	$[2,2]$	$[2,2]$
8	$(6,4) \to (\mathbf{5,4})$	$(5,3)$	$[14 \to 12, 13 \to 11]$	$[2,2]$	$[4,4]$
9	$(6,4)$	$(1,4)$	$[14,13]$	$[2,2]$	$[2,2]$
10	$(6,4)$	$(2,4)$	$[14,13]$	$[2,2]$	$[2,2]$
11	$(6,4) \to (\mathbf{5,3})$	$(5,4)$	$[14 \to 10, 13 \to 9]$	$[2,2]$	$[4,4]$

Table 2. **Example 2 strategies and payoffs**

Strategy	Position		Diameters	Payoff	
	drone 1	drone 2	updated	stage 1	stage 2
1	$(6,4)$	$(4,1)$	$[13,11]$	$[2,1]$	$[2,1]$
2	$(4,6)$	$(2,2)$	$[13,11]$	$[2,1]$	$[2,1]$
3	$(4,6)$	$(5,2)$	$[13,11]$	$[2,1]$	$[2,1]$
4	$(4,6)$	$(2,3)$	$[13,11]$	$[2,1]$	$[2,1]$
5	$(4,6)$	$(2,4)$	$[13,11]$	$[2,1]$	$[2,1]$
6	$(4,6) \to (5,4)$	$(4,4) \to (4,6)$	$[13 \to 12 \to 15, 11]$	$[2,1]$	$[0,1]$
7	$(4,6)$	$(2,5)$	$[13,11]$	$[2,1]$	$[2,1]$
8	$(4,6) \to (\mathbf{5,5})$	$(\mathbf{4,5})$	$[13 \to 12, 11]$	$[2,1]$	$[3,1]$
9	$(4,6) \to (5,5)$	$(4,5) \to (4,6)$	$[13 \to 12 \to 15, 11]$	$[2,1]$	$[0,1]$
10	$(4,6) \to (\mathbf{4,5})$	$(\mathbf{5,5})$	$[13,11]$	$[2,1]$	$[3,1]$
11	$(4,6) \to (4,5)$	$(5,5) \to (4,6)$	$[13 \to 12 \to 15, 11]$	$[2,1]$	$[0,1]$
12	$(4,6)$	$(5,8)$	$[13,11]$	?	$[2,1]$
13	$(4,6)$	$(6,8)$	$[13,11]$	?	$[2,1]$

the 6th strategy (drone 1 at (4, 6), drone 2 at (4, 5)) gives an opportunity for the first player to use the second drone as a 2-bridge ((4, 5), (5, 5)), which brings the diameter to 12 for both of the players. Now the second player finds a better position for the second drone, namely (4, 6), which bring the diameter down to 11, however, that action increases the diameter of the first player to 15. A very similar situation happens with strategies 6 and 10 (see table 3). Strategy 8, where the first drone was at (4, 6) and the second one at (4, 4), after the first stage, allows the 2-bridge ((5, 4) – (4, 4)) to appear after second stage minimisation (see Fig. 4). Strategy number 9 is special, it allows two options for the position of the second drone ((4, 5) and (4, 6)) after the first drone was moved to (5, 5). Both makes the diameter of the second player 11. However, only the first option leaves the diameter of 12 for the first player, the second option makes the diameter 15. Thus, two strategies lead to the bridge formation and bring the maximal total network improvement, but at the same time, the two strategies degrade the total payoff compared with the outcomes of the first stage.

4.3. Example 3. Allowing some foreseeing for the first player would be a natural extension of the game under consideration, since we look at the complete information games. Consider the network built on a hexagonal grid (Fig. 5): the agents of the first player are depicted by black circles, the agents of the second player are red circles. The green objects are obstacles, say forests. The game formulation is very similar, apart from

this time we will allow the first player to consider the strategies, that do not decrease the diameter, but might allow to build "a bridge" on the next stage of the game.

Table 3. **Network Configuration**

| Parameter	Value
Transmission radius	105 m
Geographical area	700 m×500 m
Number of nodes	16 for player 1
	17 for player 2
Simulation time	1200 seconds
Source node (player 2)	Node 0
Destination node (player 2)	Node 15
Physical mode	DSSS at 11 Mbps
Transfer rate	CBR at 11 Mbps
Packet size	64 bits
Propagation loss model	Friis propagation loss model
Player 1 channel number	1
Player 2 channel number	6
Routing protocol	AODV
Transport protocol	UDP
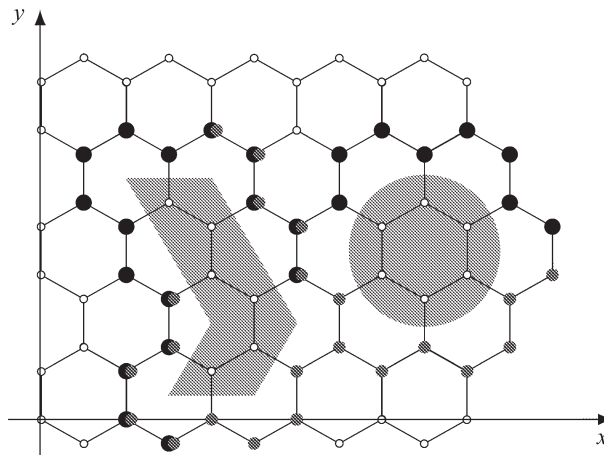
Figure 5. Example 3 topology

The set of strategies for the first player is $\{A, B, C, D, E, F, G, H, I, J, K, L, M\}$ contains four potential drone positions (Fig. 6), that are not changing the diameters, namely $\{H, K, L, M\}$. Table 4 presents the resulting payoff after the first stage of the game in the fourth column. The latter strategies are highlighted in this table. However, if we allow the first drone to be in either of H, K, L, M, the second drone might be able to install a connection over the polygonal forest, building one of the "bridges" $K - H, K - M, K - L$. Table 4 presents the resulting strategies for example 3, the allowed positions H, K, L, M, have brought significant improvement of connection (see the last column of table 4).

5. Network Simulator 3. We have clearly discussed the benefits of applying game theory for the placement of mobile agents and we have used MATLAB to implement the solution, however, we are also interested in the benefits that the strategic placements of the mobile agents would give in terms of network performance. Thus, scenarios and
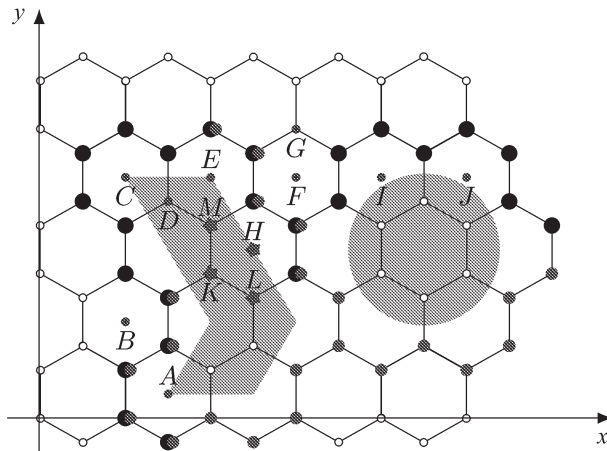
Figure 6. Possible strategies for the player 1, example 3

Table 4. **Example 3 strategies and payoffs**

Strategy	Position		Payoff	
	drone 1	drone 2	stage 1	stage 2
1	A	N	$[1,1]$	$[1,3]$
2	B	A	$[1,0]$	$[2,1]$
3	C	A	$[2,0]$	$[3,1]$
4	D	A	$[2,0]$	$[3,1]$
5	E	A	$[1,0]$	$[2,1]$
6	F	A	$[2,0]$	$[3,1]$
7	G	A	$[2,0]$	$[3,1]$
8	H	K	$[0,0]$	$\mathbf{[7,3]}$
9	I	A	$[1,0]$	$[2,1]$
10	J	A	$[1,0]$	$[2,1]$
11	K	H	$[0,0]$	$\mathbf{[7,3]}$
12	L	K	$[0,0]$	$[6,3]$
13	M	K	$[0,0]$	$[6,2]$

strategies from example 1 (see Fig. 2, table 1) were coded in C++ and simulated. Network Simulator 3 (NS-3), which is a discrete network simulator used by many researchers [36], was used.

The aim was to compare the network performance for the strategies played in example 1 above. The resulting strategies presented in table 2 consist of six strategies with a bridge being formed (strategies 2, 3, 4, 5, 8 and 11), with the clear game theory winners (strategies 2–5).

The following experiment was designed: run each strategy for player 2 for 100 seconds, then the same scenario was run with no drones at all, this enabled us to compare the results. Thus, we have a simulation running for 1100 seconds working through the 11 strategies that considered the placement of the mobile agents, and for a further 100 seconds with all drones removed (the drones were placed to positions where they could not communicate with the other nodes, or with each other at simulation time, 1100 seconds).

Several callbacks (traces) were configured, two of which to captured packets transmitted and received at the application layer, these were written to a vector and then written to disk as a comma separated value file for analysis. The mobility was traced

by capturing the time, x and y positions of each node, when the position of a node changed, this was written to disk as a ASCII mobility file. Other data captured consisted of routing tables and a file for use in Network Animator.

We measure network performance by the averaging the amount of successfully delivered packets per unit of time. We envisaged that scenarios with bridges would perform better than the rest of the scenarios.

5.1. Network configuration. Table 3 gives the other configuration parameters used for the scenario and are discussed subsequently. The simulation ran for 20 minutes to allow time for each scenario (each placement of the drone), this also allowed for stabilisation time for the route discovery process.

The routing protocol chosen for this experiment was Ad-hoc On-Demand Distance Vector (AODV) because this protocol establishes routes to destinations on demand, this kept the routes fresh. In line with the MATLAB experiment we placed our nodes in locations akin to the placements used in MATLAB, the transmission radius for the nodes was configured so that it was limited to 105 m, we placed our nodes 100 m apart to correctly map with MATLAB, which gave a total geographical area of 700×500 m, a visual representation can be seen in Fig. 2, Network Animator (NetAnim) was used to validate the placement and transmission characteristics.

Traffic flows between node 0 and node 15 were configured, with node 0 being the source node and node 15 being the destination (sink), represented by S_2 (source) and D (destination) on Fig. 2. The User Datagram Protocol (UDP) transport protocol was used to transmit a Constant Bit Rate (CBR) from the source at a rate of 11 Mbit/s.

The data used for the analysis consisted of the average received packet count for each of the situations of drone placement so that we could measure the performance of the network for each newly formed path based on the placement of the drone. In addition to this, the analysis of hop counts at 90 second intervals, silence interval (times at which no data were being received) and changes in topology were conducted.

In this research, we report on the network performance for player 2 only, however, results for player 1 correlate with those of player 2. During the running of the simulation we wanted to ensure that the paths between the source and the destination were as expected, and that the data flowed through the correct intermediate nodes and when applicable through the mobile agent, therefore, before and after the placement of a mobile agent routing tables were written to disk so this could be validated (Fig. 7).

```
Node: 0, Time: +1190.0s, Local time: +1190.0s, Ipv4ListRouting table
  Priority: 100 Protocol: ns3::aodv::RoutingProtocol
Node: 0; Time: +1190.0s, Local time: +1190.0s, AODV Routing table

AODV Routing table
Destination Gateway      Interface    Flag    Expire      Hops
10.1.1.2    10.1.1.2     10.1.1.1     UP  1.99            1
10.1.1.16   10.1.1.2     10.1.1.1     UP  5.89            15
10.1.1.255  10.1.1.255   10.1.1.1     UP  9223370846.85   1
127.0.0.1   127.0.0.1    127.0.0.1    UP  9223370846.85   1
```

Figure 7. AODV Routing Table for node 0 at 1190 seconds

5.2. Network performance analysis. The simulation allowed for the evaluation of twelve strategies, eleven of which map the solutions in example 1 (see table 1). The twelfth scenario was added so that the network performance could be evaluated without the use of drones (mobile agents). This was expected to be the worst performing topology.

The summary of results can be found in table 5 (as mentioned above, we are presenting player 2 results due to the limited space and the outcome qualitatively being very similar).

Table 5. **Example 1 simulations outcomes**

Position		Resulting diameter	Number of hops	Average rate	Total silence interval
drone 1	drone 2				
$(6,4)$	$(7,2)$	13	13	21.396	24
$(3,4)$	$(3,3)$	9	7	28.447	15
$(4,4)$	$(4,3)$	9	9	32.389	11
$(3,3)$	$(3,4)$	9	7	32.568	6
$(4,3)$	$(4,4)$	9	9	30.802	10
$(6,4)$	$(1,3)$	13	13	19.968	24
$(6,4)$	$(2,3)$	13	13	20.874	23
$(5,4)$	$(5,3)$	11	11	26.609	13
$(6,4)$	$(1,4)$	13	13	22.723	18
$(6,4)$	$(2,4)$	13	13	25.907	11
$(5,3)$	$(5,4)$	9	11	27.899	8
$(10,10)$	$(8,8)$	15	15	16.768	15

Table 5 presents the summary of the measurements for all twelve strategies (scenario) with the visualisation shown in Fig. 8.
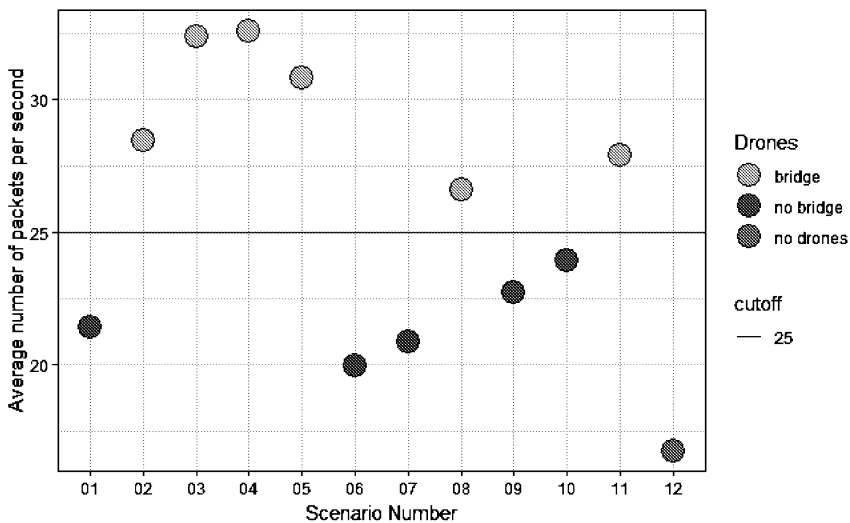

Figure 8. Mean number of packets per second

The diameter was calculated as the longest of all the shortest paths between any two nodes on the graph, while the number of hops was obtained from the AODV routing tables (see Fig. 7) recorded every 100 seconds throughout the simulation. There was a slight discrepancy due to the different algorithms calculating the most optimal route, however, the number of hops is always taken between node 0 and node 15 (nodes S_2 and D on Fig. 2), whereas the diameter could be longer for the other pair of nodes in the graph.

The total silence interval has been calculated as the number of seconds with no successful communication between the source and the destination. The bridges of drones were built and broken throughout the simulation, so we expected some delays in established communication. For simplicity of analysis, we plotted the mean amount of successfully

delivered packets per second for each scenario on Fig. 8. The latter numbers are treated as a measure of the network performance. It is clear from Fig. 8, that six scenarios, where the drones were placed such that the "bridges" were formed, are the best in terms of performance depicted by green circles. One can even imagine a clear threshold line (say, 25 packets per second) that separates them from the rest of the scenarios shown as a red horizontal line. The last scenario with no drones included, performed the worst as expected (coloured red in Fig. 8).

6. Conclusion. A novel game-theoretic model of mobile agents' placement on a Mobile Ad-hoc network was considered and an example of a game played has been implemented in MATLAB. To test if the scheme increased the performance of the network NS-3. The analysis performed on the data derived from the simulation shows that the results are positive, and that network performance clearly increased in the cases of mobile agents' positions proposed by the application of game theory.

The future research will include a detailed analysis of the observed phenomena and will concentrate on the design of new classes of strategies, including cooperative ones, to allow all players achieving the best possible outcomes. We will also consider more players and the use of more mobile agents. We are also interested in adding some randomness, say, the agents randomly are taken off the game with some apriori given probability.

Introducing the cost of drone usage, would allow to include energy-efficiency. Drone's active time is currently quite limited, thus it is vital to limit its utilization time. Hence we plan to consider the game which might finish at any random point in time [37].

References

1. Chaubey N., Aggarwal A., Gandhi S., Jani K. A. Performance analysis of TSDRP and AODV routing protocol under black hole attacks in MANETs by varying network size. *Advanced Computing & Communication Technologies (ACCT)*, IEEE, 2015, pp. 320–324.

2. Anjum S. S., Noor R. M., Anisi M. H. Survey on MANET based communication scenarios for search and rescue operations. *IT Convergence and Security (ICITCS)*, IEEE, 2015, pp. 1–5.

3. Erim O., Wright C. Optimized mobility models for disaster recovery using UAVs. *2017 IEEE 28thAnnual Intern. Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. Montreal, QC, 2017, pp. 1–5. doi: 10.1109/PIMRC.2017.8292716

4. Benkaouha H., Abdelli A., Badache N., Ben-Othman J., Mokdad L. Towards improving failure detection in mobile ad hoc networks. *IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2015, pp. 1–6.

5. Son T. T., Le Minh H., Sexton G., Aslam N., Boubezari R. A new mobility, energy and congestion aware routing scheme for MANETs. *Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, IEEE, 2014, pp. 771–775.

6. Blakeway S., Pullin A. The effect of node density on routing protocol performance in mobile Ad-hoc Networks. *Proc. of the Convergence of Telecommunications, Networking and Broadcasting*, 2007, pp. 237–240.

7. Resende C., Almulla M., Boukerche A. The use of Erasure Coding for video streaming unicast over Vehicular Ad-hoc Networks. *Local Computer Networks*, 2013, no. 38, pp. 715–718.

8. Sahingoz O. K. Mobile networking with UAVs: Opportunities and challenges, *Intern. Conference on Unmanned Aircraft Systems (ICUAS)*, 2013, pp. 933–941. doi: 10.1109/ICUAS.2013.6564779

9. Gupta L., Jain R., Vaszkun G. Survey of important issues in UAV communication networks. *IEEE Communications Surveys & Tutorials*, 2016, vol. 18, no. 2, pp. 1123–1152. doi: 10.1109/COMST.2015.2495297

10. Zhu M., Cai Z., Zhao D., Wang J., Xu M. Using multiple unmanned aerial vehicles to maintain connectivity of MANETs. *23rd Intern. Conference on Computer Communication and Networks (ICCCN)*. Shanghai, 2014, pp. 1–7. doi: 10.1109/ICCCN.2014.6911826

11. Xuelin C., Zuxun S. An overview of slot assignment (SA) for TDMA. *Signal Processing, Communications and Computing (ICSPCC)*, IEEE, 2015, pp. 1–5.

12. McKinsey J. C. C. *Introduction to the Theory of Games*. Dover, Dover Publ., 2003, 384 p.

13. Fudenberg D., Tirole J. *Game Theory*. Cambridge, The MIT Press, 1991, 604 p.

14. Poongothai T., Jayarajan K. A noncooperative game approach for intrusion detection in Mobile Ad-hoc Networks. *Computing, Communication and Networking (ICCCn)*, IEEE, 2008, pp. 1–4.

15. Wang F., Mo Y., Huang B. Defending reputation system against false recommendation in Mobile Ad-hoc Network. *Networking, Sensing and Control*, IEEE, 2008, pp. 488–493.

16. Li F., Yang Y., Wu J. Attack and flee: game-theory-based analysis on interactions among nodes in MANETs. *Systems, Man, and Cybernetics. Pt B. Cybernetics, IEEE Transactions on*, 2010, vol. 40, no. 3, pp. 612–622.

17. Wang X., Feng R., Wu Y., Che S., Ren Y. A game theoretic malicious nodes detection model in MANETs. *Mobile Ad-hoc and Sensor Systems*, IEEE, 2012, pp. 1–6.

18. Wang K., Wu M., Ding C., Lu W. Game-based modelling of node cooperation in Ad-hoc Networks. *Wireless and Optical Communications Conference (WOCC)*, 2010, pp. 1–5.

19. Ermolin N. A., Mazalov V. V., Pechnikov A. A. Teoretiko-igrovye metody nakhozhdeniia soobshchestv v akademicheskom Vebe [Game-theoretic methods for finding communities in the academic Web]. *Trudy SPIIRAN*, 2017, no. 55, pp. 237–254. (In Russian)

20. Gubanov D. A., Novikov D. A., Chkharteshvili A. G. *Sotsialnye seti: modeli informatsionnogo vliianiia, upravleniia i protivoborstva* [*Social networks: models of information influence, control and confrontation*]. Moscow, Phizmatlit Publ., 2010, 228 p. (In Russian)

21. Jackson M. O. *Social and Economic Networks*. Princeton, Princeton Univ. Press, 2008, 520 p.

22. Petrosyan L. A., Sedakov A. A. Multistage network games with complete information. *Autom. Remote Control*, 2014, vol. 75, no. 8, pp. 1532–1540.

23. Bulgakova M. A., Petrosyan L. A. Kooperativnye setevye igry s poparnymi vzaimodeistviiami [Cooperative network games with pairwise interactions]. *Mathematical game theory and its applications*, 2015, vol. 7, no. 38, pp. 7–18. (In Russian)

24. Parilina E. M. Cooperative game on sending data in the wireless network. *UBS*, 2010, vol. 31, no. 1, pp. 191–209.

25. Novikov D. A. Games and networks. *Automation and Remote Control*, 2014, vol. 75, no. 6, pp. 1145-1154.

26. Han Z., Niyato D., Saad W., Basar T., Hjorungnes A. *Game Theory in Wireless and Communication Networks. Theory, Models, and Applications*. New York, Cambridge University Press, 2012, 530 p.

27. Bazenkov N. I. Double best response dynamics in topology formation game for Ad-hoc Networks. *Autom. Remote Control*, 2015, vol. 76, pp. 323–335. doi: 10.1134/S0005117915020125

28. Gromova E., Gromov D., Timonin N., Kirpichnikova A., Blakeway S. A dynamic game of Mobile Agent placement in a MANET. *Systems Informatics, Modelling and Simulation (SIMS)*, 2016, pp. 153–158. doi: 10.1109/SIMS.2016.25

29. Plekhanova T., Gromova E., Gromov D., Kirpichnikova A., Blakeway S. The strategic placement of Mobile Agents on a hexagonal graph using game theory. *Proc. of the IEEE conference ICAT*, 2017. doi: 10.1109/ICAT.2017.8171635

30. Albert R., Jeong H., Barabasi A. L. Diameter of the World-Wide Web. *Nature*, 1999, vol. 401, pp. 130–131.

31. Petrosyan L., Zenkevich N., Shevkoplyas E. *Teoriia igr* [*Game Theory*]. Saint Petersburg, BHV-Petersburg Publ., 2012, 480 p. (in Russian)

32. Petrosyan L., Kuzyutin D. *Ustoichivye resheniia pozitsionnykh igr* [*Consistent solutions of positional games*]. Saint Petersburg, Izd. Dom Saint Petersburg University Press, 2008, 326 p. (in Russian)

33. Kuhn H. W. Extensive games and the problem of information, *Contributions to the Theory of Games*, 1953, vol. 2, no. 28, pp. 193–216.

34. Sanjab A., Saad W., Basar T. Prospect theory for enhanced cyber-physical security of drone delivery systems: A network interdiction game. *IEEE Intern. conference on Communications (ICC)*. Paris, 2017, pp. 1–6.

35. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. *Introduction to algorithms*. 2nd ed. Cambridge, Cambridge MIT Press, 2001, 1314 p.

36. Wehrle K., Guneş M., Gross J. (editors). *Modeling and Tools for Network Simulation*. Berlin, Springer Verlag Publ., 2010, 547 p.

37. Gromova E. V., Plekhanova T. M. On the regularization of a cooperative solution in a multi-stage game with random time horizon. *Discrete Applied Mathematics*, 2018, vol. 255, pp. 40–55. https://doi.org/10.1016/j.dam. 2018.08.008

Author's information:

Blakeway Stewart — PhD, Lecturer; s.blakeway@glyndwr.ac.uk

Dmitry V. Gromov — Dr.-Ing., Associate Professor; d.gromov@spbu.ru

Ekaterina V. Gromova — Dr. Sci. in Physics and Mathematics, Professor; e.v.gromova@spbu.ru

Anna S. Kirpichnikova — PhD, Lecturer; anya@cs.stir.ac.uk

Taissia M. Plekhanova — postgraduate student; taisiiaplekhanova@gmail.ru

Повышение качества работы беспроводной децентрализованной сети с использованием теоретико-игрового подхода к размещению дронов[*]

С. Блэйквэй[1], *Д. В. Громов*[2], *Е. В. Громова*[2], *А. С. Кирпичникова*[3], *Т. М. Плеханова*[2]

[1] Рексемский университет, Великобритания, LL11 2AW, Рексем, Молд Роад

[2] Санкт-Петербургский государственный университет, Российская Федерация, 199034, Санкт-Петербург, Университетская наб., 7–9

[3] Стирлингский университет, Великобритания, FK9 4LA, Стирлинг, Шотландия

В статье описывается новая теоретико-игровая постановка задачи размещения мобильных агентов при наличии беспроводных децентрализованных сетей (MANET). Задача сформулирована как многошаговая игра с полной информацией, даны определения как равновесия по Нэшу, так и кооперативного решения. Предложена модификация игры для обеспечения существования равновесия по Нэшу. В MATLAB разработана среда моделирования, позволяющая анализировать различные стратегии игроков. Программа генерирует различные игровые ситуации и определяет местоположение мобильного агента для каждого игрока, решая соответствующие задачи оптимизации. Применяя разработанную среду, подробно были рассмотрены два конкретных игровых сценария. Предложенный алгоритм был реализован и протестирован с использованием Network Simulator 3 (NS-3). Результаты показывают, что данный алгоритм повышает производительность сети.

Ключевые слова: MANET, динамические игры, многошаговые игры, местоположение дронов, графы, равновесие по Нэшу, беспроводные сети.

Контактная информация:

Стюарт Блэйквэй – PhD, лектор; s.blakeway@glyndwr.ac.uk

Громов Дмитрий Валерьевич – Dr.-Ing., доц.; d.gromov@spbu.ru

Громова Екатерина Викторовна – д-р физ.-мат. наук, проф.; e.v.gromova@spbu.ru

Кирпичникова Анна Сергеевна – PhD, лектор; anya@cs.stir.ac.uk

Плеханова Таисия Михайловна – аспирант; taisiiaplekhanova@gmail.ru