

The Strategic Placement of Mobile Agents on a Hexagonal Graph using Game Theory

Taisiia Plekhanova and Ekaterina Gromova
Faculty of Applied Mathematics and Control Processes
Saint-Petersburg State University
Saint Petersburg, Russia

Email: taisiia.plekhanova@gmail.com, ekaterina.shevkoplyas@gmail.com

Stewart Blakeway and Anna Kirpichnikova
Department of Mathematics and Computer Science
Liverpool Hope University
Liverpool, United Kingdom

Email: blakews@hope.ac.uk, kirpica@hope.ac.uk

Dmitry Gromov

Institute of Mathematics and Mechanics

Ural Branch of the Russian Academy of Sciences, Yekaterinburg, Russia
and

Faculty of Applied Mathematics and Control Processes
St. Petersburg State University, St. Petersburg, Russia
Email: dv.gromov@gmail.com

Abstract—This paper considers an application of Dynamic Game Theory with the goal of increasing the performance of a Mobile Ad Hoc Network in relation to increasing packet delivery ratio and reducing end-to-end delay by the strategic placement of drones. A multi-stage sequential game of two players, each with one drone is played out in a scenario depicted by a hexagonal graph, obstacles such as forests are also included. The algorithm for the game has been implemented in JavaScript and the results show that the best strategic locations of the placement of the drones can be established during the sequential moves of the players. Results also show that the Nash Equilibrium is achieved during play.

Index Terms—MANET; Game Theory; Nash Equilibria; Drone

I. INTRODUCTION

A Mobile Ad-Hoc Network (MANET) consists of a collection of nodes (also called agents) with the capability of communicating using wireless technology over a network established between the agents. The collection of agents could be devices such as mobile phones and tablets [1] or any other portable device that is able to transmit and receive radio waves. This forms a communications network without the need for any pre-existing infrastructure, because of this MANETs are quick to deploy [2].

There are many applications for a MANET ranging from sharing files in a personal area network setting to streaming video services between vehicles [3]. In this research we consider a MANET facilitating communication in a disaster recovery scenario after a natural catastrophe (i.e. after an earthquake) which may have rendered any existing communications infrastructure unusable. Restoring communications in this situation could potentially save lives [4], thus, the quick deployment that a MANET offers would be invaluable in this scenario.

This research builds upon research published in [5] by introducing a more realistic environment by using hexagonal

grid formations, see Figure 1, which is more realistic in terms of wireless transmission characteristics.

The nodes are split into two groups of agents. A graph is used to represent the agents positions and the communication links between the agents. The vertices of the graph identify the positions of the agents and the edges have a unit length of 1. It is assumed that when two agents have a unit length of one that they are in direct communications range. Any agents on the graph that have a unit length greater than one can communicate using intermediate agents that form part of the path [6]. If an agent becomes disconnected, i.e. is not connected to another node, the agent is said to be isolated and can not communicate with any other agent. In this paper we consider a connected graph, therefore, there will not be any isolated agents.

Each group of agents is associated with a player, i.e., a decision unit. The goal of the player is to minimize the diameter of the graph formed by the respective agents. We define the graph diameter as the maximum distance between any two agents in the graph when the shortest path is followed.

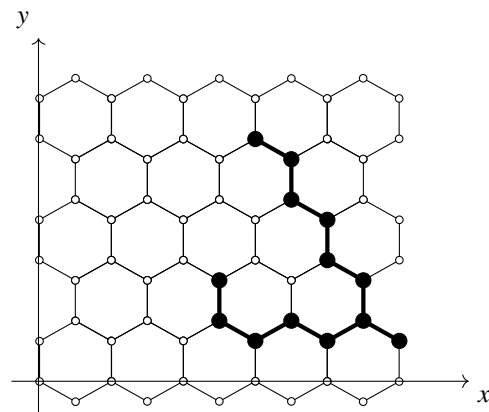


Fig. 1. A possible location of agents of a player on a hexagonal grid.

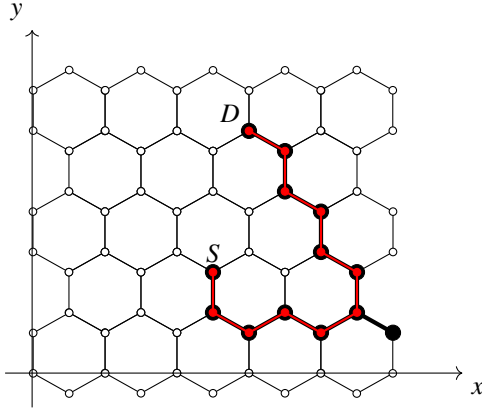


Fig. 2. Sub-graph diameter is 11 in this case.

Each player is able to utilise a moving agent (drone) which can be placed in different locations during the game, the placement of the drone creates an extended graph. We define a pay-off function as the difference in length between the original diameter (before drone placement) and the diameter of the extended graph (after node placement). The goal is to achieve a reduction in the length of the diameter for the extended graph in comparison to the initial graph - thus creating a more optimal path.

The problem is formulated as a multi-stage sequential game for two players which has been implemented using JavaScript. The game results in a Nash equilibrium.

This paper is structured as follows: Section II Game Formulation defines the notation used, and the game settings, the section also explains alternatives and the pay-off function. Section III Nash Equilibria Algorithm defines the Nash Equilibrium solution and presents the algorithm of obtaining such a solution. Section IV discusses a particular example and a numerical implementation is given. In Section V we conclude and detail our future plans.

II. GAME FORMULATION

A. General network topology

Consider the set of players $N = \{1, \dots, n\}$. Each player $i \in N$ has a non-empty set $M_i \neq \emptyset$ of agents. Every agent is placed in the vertices of a hexagonal grid $W = \{1, \dots, W_x\} \times \{1, \dots, W_y\} \subset \mathbb{R}^2$, where $W_x, W_y \in \mathbb{R}$. The position of each agent $a_i \in M_i$ is characterised by a pair of positive coordinates $v_j^i = (x_i^j, y_i^j)$ on the hexagonal grid (see Figure 1). Note that the coordinates on a hexagonal grid are defined in a specific coordinate system, typically the axial coordinate system.

It is possible to position agents of different players in the same vertex on the hexagonal grid. Stable communication (v_p^i, v_s^i) between agents v_p^i and v_s^i , $s \neq p$, of the i^{th} player exists if they are in the adjacent vertices on hexagonal grid. The two agents considered for communication are those that are the furthest apart (labelled as S and D on Figure 2).

Set of agents $V_i = \{v_p^i \in M_i, i \in N\}$ and the set of established communication links between them $E_i = \{(v_p^i, v_s^i), s \neq p\}$

define graph $R_i = (V_i, E_i)$. The union of the graphs of all the players defines the game graph R , with the non-empty finite set of vertices V and finite set of non-ordered pairs of elements from V , called edges of the graph E [7].

$$V = \bigcup_{i=1}^n V_i, E = \bigcup_{i=1}^n E_i.$$

Graph R defines the general network structure of the game, which consists of $|W_x \times W_y|$ possible positions of the agents. The number of unoccupied positions $|O|$ can be roughly estimated as [5]:

$$|W_x \times W_y| - (n_1 + n_2 + \dots + n_N) \leq |O| \leq |W_x \times W_y| - \max_{i \in N} \{n_i\}$$

If sub-graph R_i is connected, this guarantees the existence of the path between any two vertices in R_i . Note that the two graphs/sub-graphs are not connected with each other i.e. the agents of different players do not connect nor do they interact with each other.

A sequence $e = (e_1, \dots, e_k, \dots)$ of edges in the sub-graph is called a path in the sub-graph R_i , if the vertex adjacent to the edge coincides with the vertex adjacent to the next edge. Path length of the path $e = (e_1, \dots, e_k)$ is denoted by $d(e) = k$ [7].

The length of each edge on the hexagonal grid is equal to 1, therefore, the length of the edge e_i is also 1.

The sub-graph diameter $D(R_i)$ is defined to be the maximum among all the shortest paths between any two agents in the sub-graph, i.e. the shortest path for the two agents furthest apart (see Figure 2).

$$D(R_i) = \max_{(v_k, v_l) \in V_i \times V_i} d(v_k, v_l),$$

where $d(v_k, v_l)$ is the path between vertices v_k and v_l .

Since each graph R_i is connected, the diameter can be roughly estimated as [5] $D(R_i) \leq |M_i| - 1$.

B. Mobile agents (drones)

Each player controls one mobile agent (drone) denoted by q^i for the i^{th} player, $i \in N$ (we associate drones with their positions/coordinates on the grid), depicted in Figure 3.

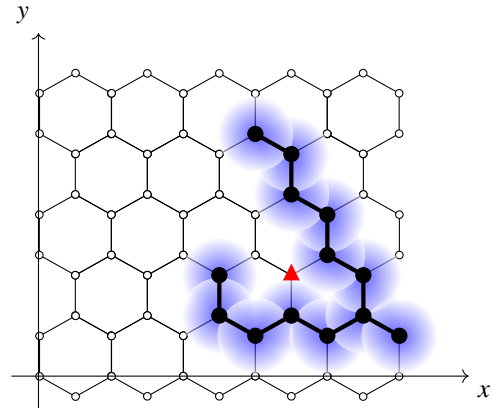


Fig. 3. Drone (denoted by a triangle) can be positioned on the grid.

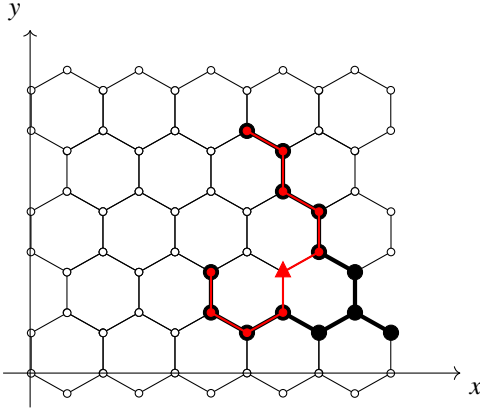


Fig. 4. Extended graph R^* after the drone has joined the game.

A mobile agent can change its position on the hexagonal grid (see Figure 3, where the mobile agent is represented by a triangle).

Communication between a mobile agent and an agent is exactly the same as an agent with another agent, they both have identical transmission characteristics.

The additional feature drones offer is the ability to communicate with the agents of other players. Therefore, the strategic placement of a mobile agent may be beneficial to both players in terms of shortening their communication paths.

C. Initial deployment pattern

At the start of the game all drones are located in such positions q_0^i , that the path length from a mobile agent to another agent is greater than 1:

$$d(q_0^i, v_p^j) > 1, \forall p \in M_i \forall i, j \in N.$$

It should be noted that a player is not initially concerned with shortening paths of another player and places its mobile agent for its own advantage only.

Subgraph R_i^* is an extended graph of the i^{th} player, if the set of vertices is defined in the following way:

$$V(R_i^*) = M_i \cup \bigcup_{i=1}^n q^i, i \in N,$$

and the set of edges is defined as the ability to communicate on the set $V(R_i^*)$. See Figure 4 which depicts the strategic placement of a mobile agent and the new path shown using red lines, resulting in a shorter diameter of 9, as opposed to the original path as shown in Figure 2 which has a diameter of 11.

D. Alternatives and the pay-off function

Each player aims to position its mobile agent such that the diameter of its extended sub-graph is smaller than the diameter of its original sub-graph.

The game is the one with complete information, i.e., each player knows the location of all their own agents and of the ones of another players; each player remembers all their own

moves and the moves of other players. The players take their turns in a priori agreed order.

Denote by \mathcal{W}_i the set of all available positions for the drone of the i^{th} player. We can limit this set by looking at a finite and countable number of locations such that they do not increase their own diameter. The way we limit the number of available positions, is that we allow the drone to take the non-taken grid vertices and the points in the centers of the cells. Hence, \mathcal{W}_i is the set of alternatives for player i .

Each turn, the i^{th} player chooses an element from a set of alternatives \mathcal{W}_i built from all the available positions that decrease the diameter of the extended graph, together with the locations that do not change the diameter, but might minimize the diameter of extended sub graph in the future. Thus, the set $\mathcal{W}_i = \mathcal{W}_i^{\text{now}} \cup \mathcal{W}_i^{\text{hope}}$ consists of:

- the alternatives, that decrease the diameter of the extended sub graph of the i^{th} player, denoted by $\mathcal{W}_i^{\text{now}}$
- the alternatives that do not change the diameter on the current turn, however, we hope, will bring benefits later (in the future) denoted by $\mathcal{W}_i^{\text{hope}}$

In regards to the second subset, initially all players know the number of available drones and all the locations of the agents of other players, thus, at the initial state of the game the players can reason towards the alternatives a few steps ahead.

The search for *hope alternatives* can be formulated as follows:

- Define the set $\mathcal{W}_i^{\text{now}}$ as the set of all drone positions that lead to decreasing the diameter.
- Set $\mathcal{W}_i^{\text{hope}} = \emptyset$. The i^{th} player solve the the following optimization problems:

$$\bar{q}_h^i = \underset{q^i \in W \setminus V(R_i^{\text{ext}})}{\operatorname{argmin}} D(R_i^*),$$

where R_i^{ext} is the graph R_i extended with $N - 1$ agents and R_i^* is R_i^{ext} extended with q_h^i .

- The search is to be performed over all possible R_i^{ext} . If $q_h^i \notin \mathcal{W}_i^{\text{now}}$, add it to $\mathcal{W}_i^{\text{hope}}$, i.e., we set $\mathcal{W}_i^{\text{hope}} = \mathcal{W}_i^{\text{hope}} \cup q_h^i$.

The number of steps is limited since the number of mobile agents is limited (each player has one mobile agent). For the second player the number of alternatives to consider is even smaller. The future benefit can reduce the game length/duration, and also can result in better solutions. The choice at the first step made by the i^{th} player of such an alternative (for the future benefits) leads to a zero pay-off at the first step, $H_i^1 = 0$.

Definition 1. An alternative w_i is a *hope alternative* w^{hope} in the two-players multi-stage game, if

- $w^{\text{hope}} \in \mathcal{W}_1 \cap \mathcal{W}_2$, where $\mathcal{W}_i, i = 1, 2$ is the alternative of the i^{th} player
- $H_i^1 = 0, i = 1, 2$
- $x_1^s = x_2^p$ and $y_1^s = y_2^p$, where (x_i^s, y_i^s) are the coordinates of the s^{th} agent of the i^{th} player, $s \in M_i$.

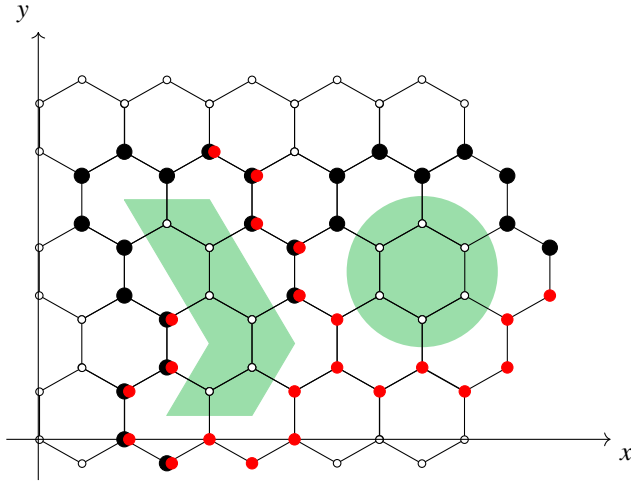


Fig. 5. The initial topology of the game before it begins.

Note that for each set of alternatives at each step of the i^{th} player depends on the previous steps and previous steps of the other player. Therefore, after the first step of each i^{th} player the following optimisation problem is solved [5]:

$$\bar{q}^i = \underset{q^i \in \mathcal{W}_i}{\operatorname{argmin}} D(R_i)$$

The game is optimising the mobile agent placement for a search and rescue scenario, therefore, the player's strategies will be "benevolent", i.e. when there are two equivalent alternatives, the i^{th} player will choose the option which is more advantages for other player [9]. In our case each player moves their own mobile agents so that it does not increase the sub graph diameter of another player. This process is formulated in terms of a multi-stage game with complete information:

$$\Gamma = \{N, \mathcal{W}, H\}$$

where N is the set of players; $\mathcal{W} = \prod_{i \in N} \mathcal{W}_i$ is a Cartesian product of strategy sets \mathcal{W}_i and \mathcal{W}_i is the set of available strategies for player i ; H is a vector-valued function such that for any given communication infrastructure G , $H_i : \mathcal{W} \rightarrow R$ is the utility (pay-off) function for player i .

The game is over when none of the players cannot reduce the diameter of extended sub graph any further. The pay-off function for the i^{th} player is:

$$H_i = d(R_i) - d(R_i^*) \geq 0 \quad (1)$$

Therefore, this is formulated as a finite game with complete information.

III. NASH EQUILIBRIA ALGORITHM

This section presents a numerical example that searches for the Nash equilibrium.

Definition 2. 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:

$$H_i(G, Q^{NE}) \geq H_i(G, Q_{-i}^{NE})$$

where $Q_{-i}^{NE} = \{q_j^{NE}\} \cup \{q_i\}$, $j \neq i$, $q_i \in \mathcal{W}_i$.

A. The Algorithm for Searching for the Nash Equilibrium

In section II-D we have considered the following alternatives:

- 1) the first player searches for the hope alternatives (Definition 1) and for alternatives which reduces the diameter of sub graph R_1 ;
- 2) the first player computes a pay-off function for the alternatives found at step 1. By definition 1 for the hope alternative pay-off function will be zero for the first player, therefore, we do not have to compute the pay-off function for the hope alternatives;
- 3) the second player searches only for alternatives which reduces the diameter of sub graph R_2 ;
- 4) the second player computes a pay-off function for the alternatives found. This might be final the step.
- 5) if the previous step was not final, the first player searches for alternatives which reduce the diameter of the sub graph R_1 ;
- 6) then repeat the algorithm from step 3 until condition II-D is satisfied.

IV. NUMERICAL EXAMPLE

A. Simulation software

The game has been implemented using JavaScript. During the initialisation stage each player are given the coordinates of the agents, four arrays are used to store these coordinates, X for player 1, Y for player 1, X for player 2 and Y for player 2. Then several draw functions are called which creates a visual representation of the graph, the agents and the obstacles.

A Dijkstra algorithm [10] is used to create a distance matrix for the agents, one for each player. The longest path is then calculated for each player using their distance matrix. All possible coordinates for the drone are obtained and stored in a vector, each player maintains its own drone coordinate vector.

For each possible location of each player's drone, the algorithm places the drone at that location which results in an extended graph. The diameter of the extended graph is calculated.

The difference between the diameter of the original graph and the diameter of the extended graph is used to give the value of the payoff function. The code follows the algorithm presented in Section III-A.

B. Example

In the following example, the initial state of the game for the two players is discussed and depicted in Figure 5. The drones are not yet the part of the network and thus are not included in Figure 5 because they are not connected to any agents initially. The agents of the first player are depicted by red circles. The agents of the second player are depicted by black circles. The agents of both players are present at the same location in places. The green areas represent obstacles such as lakes or forests.

The initial diameters in 5 are $d(R_1) = 22$, $d(R_2) = 15$.

TABLE I
SIGNIFICANT STRATEGIES FOR BOTH STAGES

No. alternative	First Player Drone	first Pay-off	Second Player Drone	pay-off function
1	A($3\sqrt{3}$; 1.5)	(1; 1)	N($4.5\sqrt{3}$, 3)	(1; 3)
2	B($2.5\sqrt{3}$; 3)	(1; 0)	A($3\sqrt{3}$, 1.5)	(2; 1)
3	C($2.5\sqrt{3}$; 6)	(2; 0)	A($3\sqrt{3}$, 1.5)	(3; 1)
4	D($3\sqrt{3}$; 5.5)	(2; 0)	A($3\sqrt{3}$, 1.5)	(3; 1)
5	E($3.5\sqrt{3}$; 6)	(1; 0)	A($3\sqrt{3}$, 1.5)	(2; 1)
6	F($4.5\sqrt{3}$; 6)	(2; 0)	A($3\sqrt{3}$, 1.5)	(3; 1)
7	G($4.5\sqrt{3}$; 7)	(2; 0)	A($3\sqrt{3}$, 1.5)	(3; 1)
8	H($4\sqrt{3}$; 4.5)	(0; 0)	K($3.5\sqrt{3}$, 4)	(7; 3)
9	I($5.5\sqrt{3}$; 6)	(1; 0)	A($3\sqrt{3}$, 1.5)	(2; 1)
10	J($6.5\sqrt{3}$; 6)	(1; 0)	A($3\sqrt{3}$, 1.5)	(2; 1)
11	K($3.5\sqrt{3}$; 4)	(0; 0)	H($4\sqrt{3}$, 4.5)	(7; 3)
	K($3.5\sqrt{3}$; 4)	(0; 0)	M($3.5\sqrt{3}$, 5)	(6; 2)
12	L($4\sqrt{3}$; 3.5)	(0; 0)	K($3.5\sqrt{3}$, 4)	(6; 3)
13	M($3.5\sqrt{3}$; 5)	(0; 0)	K($3.5\sqrt{3}$, 4)	(6; 2)

C. Using algorithm for searching Nash equilibrium

The algorithm presented in section III-A was implemented in JavaScript, and the above game was simulated. The algorithm shows that optimal paths were created by the strategic placement of drones using a game theory approach.

Table I presents the results and shows alternatives found for the first player by implementing steps 1 and 2 of the algorithm discussed in section III-A, the alternatives are in the second column of the Table. The set of alternatives of the first player is \mathcal{W}_1 and contains 13 potential drone positions A–M with the corresponding changes in the diameters listed in the third column. The set of hope alternatives $\mathcal{W}_1^{\text{hope}} \subset \mathcal{W}_1$ are highlighted in bold in table I, and they are

$$\mathcal{W}_1^{\text{hope}} = \{H, K, L, M\}$$

None of the alternatives bring a decrease in the diameter at this stage. Figure 6 presents the alternatives for the first player, the hope alternatives are depicted by blue stars (see also bold rows in Table I).

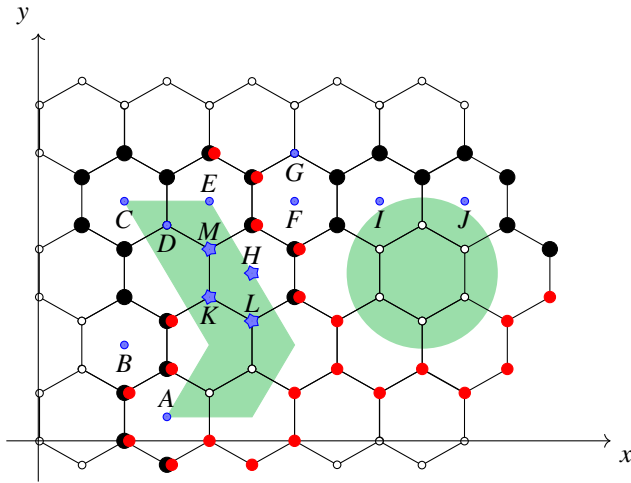


Fig. 6. Alternatives of the first player .

The remaining alternatives $\mathcal{W}_1 \setminus \mathcal{W}_1^{\text{hope}}$ are currently optimising the diameter and are represented by blue circles.

Table I also presents the alternatives for the second player by implementing step 3 and 4 of algorithm presented in section III-A.

Hope alternatives are highlighted in bold, see also Figure 7 which corresponds to the alternatives. The mobile agents appear on Figure 7 which are represented by blue stars and show the possible beneficial locations of the first player. The mobile agent for the second player is denoted by yellow diamonds.

The situation is quite different now, comparing the final pay-offs (listed in the fifth column of the Table), one can see that the best diameter decrease was due to the hope alternatives $\mathcal{W}_1^{\text{hope}} = \{H, K, L, M\}$, with the following results: drones placed at H, K in whichever order, give a payoff value of 7 and 3 correspondingly to the first and second player; drones placed at K, M in whichever order, give payoff of 6 and 2 correspondingly to the first and second player; drones

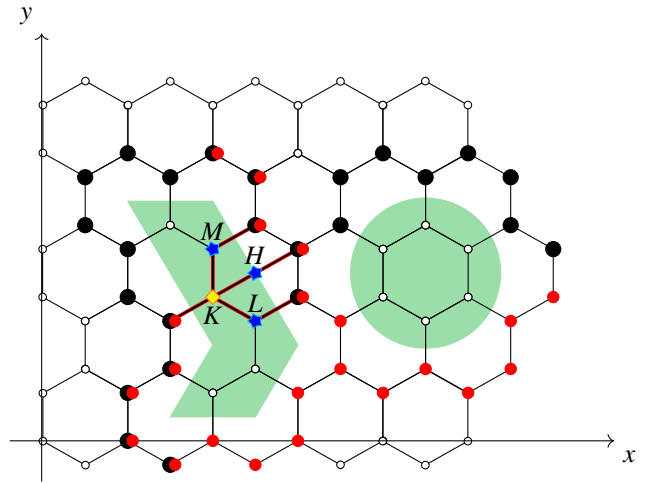


Fig. 7. Hope alternatives resulted in optimisation

placed at L, K , give payoff of 6 and 3 correspondingly to the first and second player.

After this step the game is over because the players alternatives have been exhausted, which means that the diameters of their sub graphs R_1, R_2 can not be further reduced. The found alternatives K, H give us $d(R_1^*) = 15$ and $d(R_2^*) = 12$ which are the Nash equilibria, because they satisfy definition 2. The discussion of the uniqueness of Nash equilibria is outside the scope of this work.

V. CONCLUSIONS

This paper extends the findings in [5] by implementing a hexagonal topology rather than a topology made up of squares. This research also found new classes of solutions. This novel approach and introduction of new types of alternatives allowed us to define a finite multi-stage game with complete information, which led to the Nash Equilibrium. Our future research will consider more players with the optionality of allowing variable numbers of mobile agents between the players.

Other future work in this area may consist of creating a Cooperative framework of the game, for example Cooperative solutions for the network structure such as Shapley value, Core, Mayerson, Owen, and Aumann-Dreze vectors.

ACKNOWLEDGEMENT

The work of D. Gromov on the construction of optimal Nash strategies for the framework of MANET has been supported by the grant 17-11-01093 from Russian Scientific Foundation. E. Gromova has been supported by the grant 17-51-53030 from Russian Foundation for Basic Research.

The authors thank the Faculty of Applied Mathematics and Control Processes, Saint-Petersburg State University, Russia and the Department of Mathematics and Computer Science, Liverpool Hope University, Liverpool, UK for the continued support with this international cross-discipline research.

REFERENCES

- [1] J. Walton, S. Blakeway and A. Kirpichnikova, "An Analysis of MANET Routing Protocol Performance for an Interactive User Engaging Quiz," 2016 International Conference on Systems Informatics, Modelling and Simulation (SIMS), Riga, 2016, pp. 147-152.
- [2] M. Gerla, C. F. Chiasserini, K. Mase, E. Modiano and J. Redi, "Mobile ad hoc wireless networks," in *Journal of Communications and Networks*, vol. 6, no. 4, pp. 291-294, Dec. 2004. doi: 10.1109/JCN.2004.6596848
- [3] 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, P. 715-718.
- [4] P. Begerow, S. Krug, S. Schellenberg and J. Seitz, "Robust reliability-aware buffer management for DTN multicast in disaster scenarios," 2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM), Munich, 2015, pp. 274-280. doi: 10.1109/RNDM.2015.7325240
- [5] E. Gromova, D. Gromov, N. Timonin, A. Kirpichnikova and S. Blakeway, "A Dynamic Game of Mobile Agent Placement in a MANET," 2016 International Conference on Systems Informatics, Modelling and Simulation (SIMS), Riga, Latvia, 2016, pp. 153-158.
- [6] A. P. Mandhare and S. V. Kadam, "E-TWRP: Establishing trust worthy reliable path in Mobile Adhoc Network," 2016 International Conference on Emerging Technological Trends (ICETT), Kollam, 2016, pp. 1-5. doi: 10.1109/ICETT.2016.7873711
- [7] Robin J. Introduction to Graph Theory // Edinburgh: Oliver and Boyd, 1972. 209 p.
- [8] Novikov D. . Games and networks // *Game theory and applications*, 2010. . 107-124 (in Russian)
- [9] L. Petrosyan, N. Zenkevich and E. Shevkoplyas (2012), *Game Theory*, BHV-St. Petersburg, 2012 (in Russian).
- [10] Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 269271.