Adigar: a drone simulator for agriculture

Akarshani Amarasinghe[1,]*, Lakshman Jayaratne[1] and Viraj B. Wijesuriya[2]

[1]University of Colombo School of Computing, UCSC Building Complex, 35, Reid Avenue, Colombo 7, 00700, Sri Lanka
[2]Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford, OX1 3QD, United Kingdom

Adigar is a drone simulator developed to reduce the adverse effects of pesticides during the spraying process. Here, we propose a path planning algorithm to cover all arable areas of a farmland, while avoiding unsafe areas. The proposed solution outputs the optimal path for the farmland and the drone can fly over along this path to spray pesticides without human intervention. This approach highlights the concept of using drones for agricultural purposes with minimum human intervention.

Keywords: Agriculture, autonomous drones, human intervention, reinforcement learning, pesticides.

DUE to the higher demand for food in both developing and developed countries, farmers tend to use pesticides. These are chemical mixtures which consist of heavy metals such as arsenic, lead, cadmium, etc.[1,2]. According to the World Health Organization (WHO), pesticides are the fundamental cause of chronic kidney disease (CKD)[1,2]. Unfortunately, the farmers and their families living near commercial farmlands are victims of the adverse effects of pesticides like abdominal pain, drowsiness, poor memory, vomiting, etc. due to overexposure to pesticides[3].

Farmers in the developing countries use more than the approved amount of pesticides to increase the quality and quantity of their harvest, ignoring their personal safety[4]. As a result, remnants of pesticides during the spraying process mix with the air, soil and groundwater, and harm non-targeted vegetation and organisms.

It is essential to develop a long-term solution for regulating pesticides usage, minimizing human intervention during pesticides spraying and reducing environmental contamination. Drones are popular due to their manoeuvrability, micro-scaled size and lightweight. Using a drone in the agricultural sector is not a novel approach, since it has already been successfully applied in soil and field analysis, planting, crop spraying, crop monitoring, irrigation and health assessment in farmlands[5]. Utilizing a fully autonomous drone to spray pesticides can accelerate the spraying process and minimize the extravagant cost of expensive labour. Also, it can diminish human intervention during spraying and ensure safe pesticide usage.

*For correspondence. (e-mail: m.c.a.amarasinghe@gmail.com)

In this study we address the question: 'How do we develop a drone solution for spraying pesticides in arable lands with optimal safe pesticide usage and minimum human intervention?'. As a solution, we introduce a simulator called Adigar[6–8] for an autonomous drone to spray pesticides in arable lands. Adigar has two modules: the 2D grid maker module and the path planning algorithm module. Here, when the user (the farmer) inputs the relevant details of the farmland to Adigar, it outputs the 2D grid of the farmland using the 2D grid maker module. Then, the path planning algorithm module outputs the optimal path for the generated 2D grid. Thereafter, the drone can fly along the proposed optimal path to spray pesticides/fertilizers.

Scope and assumptions

(1) Adigar is a simulator for a single autonomous drone.
(2) It proposes the optimal path suitable for a single drone and not a swarm of drones.
(3) The user of Adigar is a farmer and we assume that the user has sufficient technical literacy to work with the simulator.
(4) We assume the flying environment of the drone has no wind.
(5) At present, we do not optimize the drone parameters like energy consumption, type and the amount of pesticides/fertilizers sprayed, battery draining, altitude, speed and acceleration.
(6) Also, we assume the environment has good internet connectivity to enable communication link among the farmer, the drone and the central system.
(7) The user knows all the safe and unsafe areas of the farmland and the farmland is a fully observable environment.
(8) Adigar is applicable only for paddy fields.
(9) Adigar does not indicate the landing of the drone after finishing the spraying process or at the death of the drone battery.
(10) The starting location (initial distribution) of the flight of the drone is always located at the top left corner of the farmland and it is an obstacle-free area.
(11) Adigar considers only the four main directional movements of a drone; left, right, up and down, and does not include the diagonal directional movements.

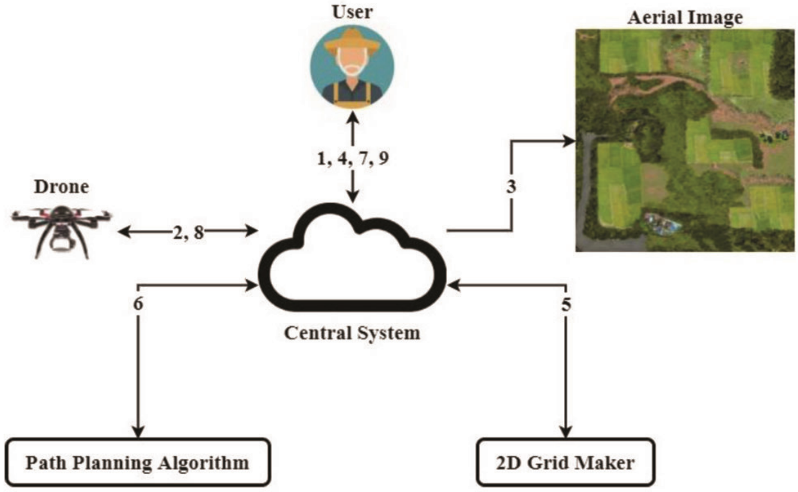

Figure 1. The workflow of Adigar.

Significance

Adigar has been developed with the purpose of minimizing human intervention in the pesticides spraying process. Since Adigar is a drone simulator, it can help regulate the over-usage of pesticides while decreasing the overexposure to them and their adverse effects. It ensures the safety of not only humans, but also the other organisms in the environment. The regulated pesticides usage will help secure the quality and quantity of crop production while saving the farmer from excessive pesticides cost.

Moreover, a single autonomous drone can cover a much larger area more quickly, compared to the manual spraying process. This simulator is a good and safe platform for both pesticide and fertilizer industries, since it can be utilized to spray not only pesticides, but also fertilizers.

Related work

Using drones for mission planning is popular. Mission planning is used to perform a set of tasks autonomously with the help of one or more robots[9]. It can execute a set of dangerous tasks without human intervention and at low cost[10]. Logically constrained neural fitted Q-iteration (LCNFQ) is one such mission-planning approach that has been developed for a continuous state space Markov decision process (MDP). Here, the authors have utilized linear temporal logic (LTL) since the MDP traces must satisfy the given LTL formula[11]. Then they convert the LTL formula to a deterministic Büchi automata (DBA), leading to the product Markov decision process (PMDP). However, the traditional Q-learning algorithm is not capable of covering a continuous state space of the MDP, and the environment is unknown[12]. Thus, researchers have introduced LCNFQ[11]. Even though this algorithm has a 98% and 99% success rate[11], it cannot be executed for a fully observable environment like our research problem.

Drone simulator is a software program which stimulates a real drone flight in a simulated environment[9]. DJI simulator is one of the popular drone simulators[13]. It has different functions, including powerline inspection, and search and rescue[13]. However, it cannot fly autonomously and needs a drone pilot for navigation. Moreover, it cannot avoid unsafe areas automatically and is incapable of providing an optimal path for the drone flight. Its cost is also relatively high[14].

FPV Freerider[15] is another drone simulator of lower cost[14]. However, it also needs a drone pilot for navigation. It cannot avoid unsafe areas and cannot propose an optimal path for navigation.

Workflow of Adigar

Thus it is necessary to come up with a cost-effective, autonomous drone simulator for pesticide/fertilizer spraying in a fully observable environment with optimal path planning. We have implemented Adigar focusing and including all these features. As represented in Figure 1, Adigar involves nine steps as follows:

Step 1: Initially, the user carries a global positioning system (GPS) device around the boundary of the farmland to obtain its geo-locations. Then, he uploads the acquired geo-locations to the central system of Adigar via a mobile application. The central system manages the operations of the drone. The mobile application is the user interface (UI), and it helps the farmer communicate with the central system.

Step 2: The central system commands the drone to fly over the area based on the geo-locations and capture images of the farmland. After the drone captures these images, it uploads them to the central system. Here, the drone has a GPS sensor for geo-location identification, a general packet radio service (GPRS) sensor to communicate via internet with the central system, a camera to capture images of the farmland, a spray tank to spray pesticides and a charging port to recharge the drone battery.

Step 3: The central system generates the relevant aerial image of the farmland using the uploaded images by the drone.

Step 4: The user can view the aerial image through the mobile application, and identify the safe and unsafe areas. Then, he sends these details to the central system. Here, the safe areas are those with crops, while the unsafe areas are traps, dead zones and forbidden areas (dams, ponds, wastelands, streams and private properties). The reason for avoiding dams, ponds and streams is because these areas consist of huge water bodies, and they can restrict the drone signals. There is a higher probability of obstacles such as big trees which can interfere with the flying path of the drone in traps, wastelands, dead zones and private properties. Moreover, private properties may have interference areas like telecommunication towers and electricity supply lines, and these can drop the drone signals. The other reason for considering a private property as an unsafe area is because it may harm the property owner's privacy, and it is against Government rules and regulations.

Step 5: The central system orders the 2D grid maker module to generate the 2D grid image of the farmland based on the inputs in step 4.

Step 6: The central system instructs the path planning algorithm module to generate the optimal path for the farmland based on its 2D grid image.

Step 7: The central system informs the user that the drone is ready to fly. The user can then command the central system to fly the drone to spray pesticides/fertilizers.

Step 8: Subsequently, the central system commands the drone to spray pesticides/fertilizers according to the proposed path.

Step 9: The user can view the spraying process in real-time using the mobile application.

Implementation

As mentioned earlier, Adigar has two modules.

2D grid maker module

This module is used to generate the relevant 2D grid image for the obtained aerial image. It consists of five steps.

Step 1: After the user observes the aerial image, he divides the grid into identical squares. As illustrated in

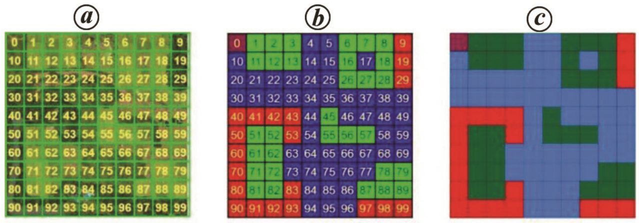

Figure 2. Outputs of the 2D grid maker module. ***a***, Aerial image with cell numbers; ***b***, Cell categorization of the aerial image; ***c***, 2D grid.

Figure 2 *a*, the user, the numbers the squares and the square numbering process starts from 0. The user can divide the aerial image into any number of squares as required. Figure 2 *a* is a 10×10 grid and the ending cell number is 99.

Step 2: As stated earlier, the initial distribution is located at the top left corner of the 2D grid. The user needs to identify the safe and unsafe areas of the farmland. The rest of the cells are considered as obstacle-free areas.

Step 3: Adigar represents the initial distribution in purple, the safe areas in green, the unsafe areas in red and the obstacle-free areas in blue.

Step 4: According to Figure 2 *b*, the user inputs the following data to the 2D grid maker module:

(a) The number of horizontal cells (10).
(b) The number of vertical cells (10).
(c) The cell number of the initial distribution (0).
(d) The cell numbers of the safe areas (1, 2, 3, 6, 7, 8, 11, 12, 13, 16, 18, 26, 27, 28, 45, 51, 52, 55, 56, 57, 61, 62, 71, 72, 78, 79, 81, 82, 87, 88 and 89).
(e) The cell numbers of the unsafe areas (9, 19, 29, 40, 41, 42, 43, 50, 53, 60, 70, 80, 83, 90, 91, 92, 93, 97, 98 and 99).

The user does not need to input the cell numbers of the obstacle-free areas.

Step 5: Figure 2 *c* shows the output of this module.

Path planning algorithm module

After obtaining the 2D grid image from the 2D grid maker module, Adigar uses the path planning algorithm module to generate the optimal path for the drone to fly, while spraying. The generated path covers all the safe areas while keeping away from the unsafe areas in a single drone flight. The algorithm of this module is shown in Figure 3 and the implementation details are explained below.

Step 1: This module converts the 2D grid to the relevant state-action graph (SAG). SAG represents state-action transitions of the grid, avoiding transitions to and from unsafe areas. The reason for avoiding these transitions is to reduce the complexity of the state space.

Figure 4 *a* represents a sample SAG. S_0, S_1, S_2, S_3 and S_4 can be safe or obstacle-free areas, or both, in the 2D grid. If the drone is in S_0, it can move to S_1, S_2, S_3 and S_4 with action values 0 (left), 1 (right), 2 (up) and 3 (down) respectively. As mentioned earlier, we do not consider diagonal directional movements. The relevant SAG transitions from S_0 are $\{S_0, 0, S_1\}$, $\{S_0, 1, S_2\}$, $\{S_0, 2, S_3\}$ and $\{S_0, 3, S_4\}$. Moreover, we can consider movements of S_1, S_2, S_3 and S_4 to S_0 as $\{S_1, 1, S_0\}$, $\{S_2, 0, S_0\}$, $\{S_3, 3, S_0\}$ and $\{S_4, 2, S_0\}$ respectively. SAG contains all the transitions, except those to and from unsafe areas. If we consider these transitions as well, then the state space of SAG may be much larger and more difficult for the training process.

Step 2: The obtained SAG is used for clustering. According to Figure 2 *c*, there are 31 safe areas. It is hard to find a path from the initial distribution to cover all the safe areas within a single visit. Traditional path-finding algorithms cannot cover multiple goals in a single visit. Reducing the number of safe areas (goals count) helps accelerate the training process. As a solution, we use the clustering step.

Here, we merge successive safe areas into a single goal cluster (Figure 5). There are five goal clusters: G_0, G_1, G_2, G_3 and G_4.

Step 3: After merging the successive safe areas into a single goal cluster, we must acquire the relevant LTL formula to cover all goal clusters. Otherwise, SAG is not sufficient to check whether we have covered all goal clusters. The relevant LTL formula for the grid is as follows:

$$\Diamond G_0 \wedge \Diamond G_1 \wedge \Diamond G_2 \wedge \Diamond G_3 \wedge \Diamond G_4.$$

According to the formula, the drone should eventually cover all goal clusters.

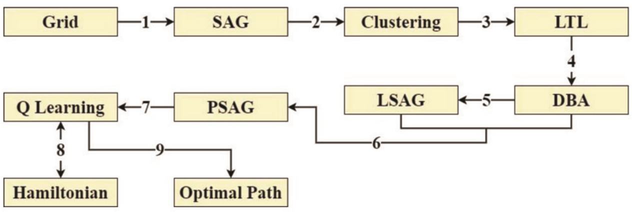

Figure 3. The proposed path planning algorithm.

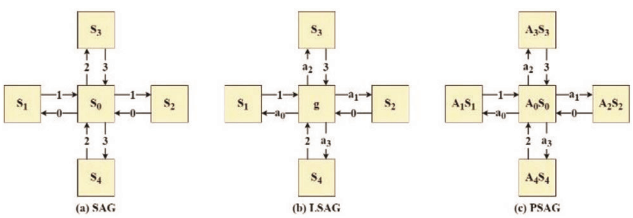

Figure 4. Graphs used in the path planning algorithm.

Step 4: To satisfy the LTL formula, we apply DBA[16].

Step 5: Since we must update transitions in the SAG for the new goal cluster transitions, we apply lumped state-action graph (LSAG). Here, we start the numbering process of the goal clusters from 100 as the last cell number of the 2D grid is 99, and increase the number until we finish the process for each goal cluster (Figure 6). Next, we change the action values of the goal cluster (Figure 4 *b*).

In Figure 4 *b*, *g* is the goal cluster and its action values (a_0, a_1, a_2 and a_3) are different from the previous action

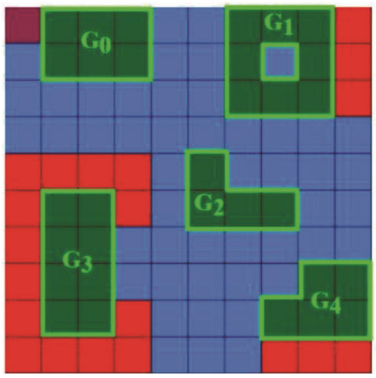

Figure 5. Goal clusters.

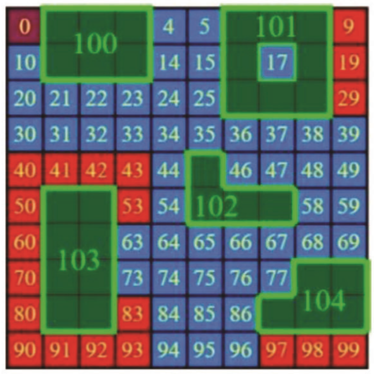

Figure 6. Updated 2D grid representation.

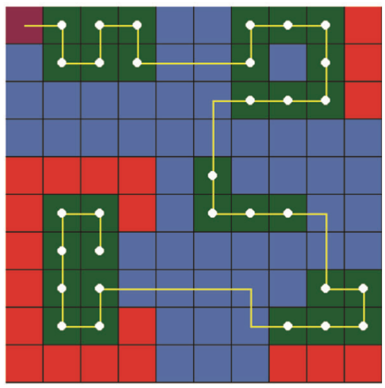

Figure 7. 2D grid with optimal path.

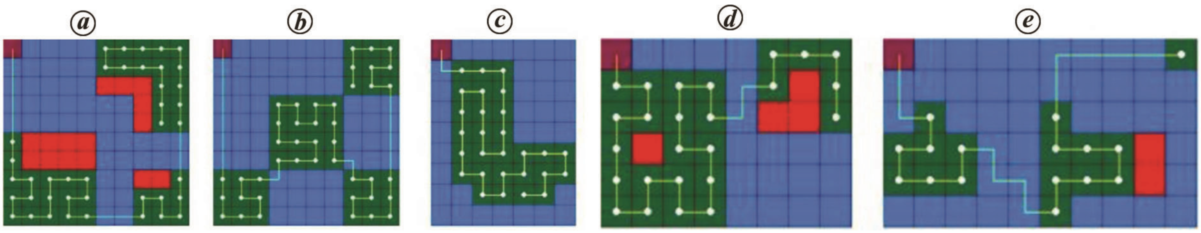

Figure 8. Test results for different 2D grids.

Table 1. Path accuracy for different 2D grids

Grid	Goals count	Deterministic Büchi automata states count	Accuracy (%)
a	3	8	70
b	4	16	70
c	1	2	100
d	2	4	100
e	3	8	90
Overall accuracy (%)			86

values (0, 1, 2 and 3). Here, S_1, S_2, S_3 and S_4 are obstacle-free areas. These a_0, a_1, a_2 and a_3 values start from 10, to distinguish them from the normal action values (0, 1, 2 and 3). However, the relevant LSAG transitions from g have the same format of the SAG transitions.

Step 6: By combining DBA and LSAG, we obtain the product state-action graph (PSAG). According to the representation in Figure 4 *c*, PSAG has hybrid states (A_0S_0, A_1S_1, A_2S_2, A_3S_3 and A_4S_4) of the DBA states (A_0, A_1, A_2, A_3 and A_4) and the LSAG states (S_0, S_1, S_2, S_3 and S_4). The representation of the PSAG transitions is similar to that of the SAG and LSAG transitions.

Step 7: Since the PSAG has several acceptable paths and a large address space, a typical path planning algorithm is not feasible enough. To solve this, we utilize the Q-learning algorithm which is classified under reinforcement learning[12].

Step 8: Q-learning can approach only the goal clusters, but cannot visit inside a goal cluster to cover all the safe areas inside it. As a solution, we check whether there is a Hamiltonian cycle inside a goal cluster while Q-learning is in the training process[17]. If there is no Hamiltonian path inside the goal cluster, the training process tries another path to approach the same goal cluster until it finds a Hamiltonian path inside it.

Step 9: At the end of the training process of Q-learning, this module outputs the grid with the optimal path (Figure 7). In Figure 7, the yellow line shows the proposed optimal path for the drone flight, which starts from the initial distribution and covers all the safe areas in a single visit. A white dot in the middle of each safe area represents the pesticides/fertilizers spraying locations. The drone needs to hover in-place at the location, which is represented by the white cell, sprays the approved amount of pesticides/fertilizers, and moves forward along the proposed path. Finally, the obtained path covers all the safe areas while avoiding all the unsafe areas in a single visit.

Here, we do not consider the landing of the drone after completion of the spraying process.

Evaluation and results

We next evaluated the accuracy of the proposed path planning algorithm. We have tested whether the proposed algorithm in Adigar outputs the optimal path for different 2D grids (Figure 8). We executed the proposed algorithm ten times per grid, and Table 1 shows the results obtained. We also tested the percentage of the obtained shortest path count using the proposed path planning algorithm.

According to Table 1, grids (a) and (b) have the lowest accuracy. This is because their goals count and DBA states count are relatively higher than the other three grids, and there are comparatively more obstacle-free areas between the initial distribution and the goal clusters. Since grid (e) has many obstacle-free areas between the initial distribution and the goal clusters, it also has a lower accuracy. However, the proposed path planning algorithm outputs the shortest path for all grids more than six times out of ten executions. As an average we get an 86% accuracy for all the 2D grids illustrated in Figure 8. According to these results, the proposed path planning algorithm outputs the optimal path for a given 2D grid.

Limitations

The proposed path planning algorithm in Adigar has a few limitations:

(1) It cannot access a goal cluster covered with unsafe areas unless it has more than two entrance areas (safe or obstacle-free areas).
(2) It can only visit goals which are either square or rectangular in shape.

Conclusion and future work

In this study, we propose an optimal path planning algorithm in a simulated environment, viz. Adigar for an autonomous drone to spray pesticides/fertilizers in a farmland to minimize human intervention during the spraying process and reduce overexposure to pesticides. Adigar consists of two modules: the 2D grid maker module and the path planning algorithm module. Their outputs are the 2D grid environment of the farmland and the optimal path for the generated 2D grid to spray pesticides/fertilizers. Both are offline modules due to the farmland being a fully observable environment. According to the obtained results, the proposed path planning algorithm gives the optimal path.

Most of the drone simulators available are not capable of achieving multiple safe areas with optimal path planning. However, we have implemented Adigar to overcome all the problems in the available drone simulators and mission planning algorithms.

In the future, we are planning to employ Adigar in a real-world environment with a fully customized autonomous drone to reduce the overuse of pesticides. Moreover, we hope to test Adigar for its optimal pesticides/fertilizers usage while reducing human intervention during the spraying process as a pilot study before releasing it to the commercial market. Furthermore, we anticipate to conduct an evaluation based on the positive impact of Adigar against the adverse effects of pesticides while saving costs. Moreover, we hope to optimize the drone parameters. In the future we expect to improve Adigar for a swarm of drones, and sell or rent it to the farmers in a cost-effective manner[7]. Furthermore, this approach ensures the safety of not only humans, but also of other organisms.

1. Jayasekara, K. B., Dissanayake, D. M., Sivakanesan, R., Ranasinghe, A., Karunarathna, R. H. and Priyantha Kumara, G. W. G., Epidemiology of chronic kidney disease, with special emphasis on chronic kidney disease of uncertain etiology, in the north central region of Sri Lanka. *J. Epidemiol*., 2015, **25**, 275–280.
2. Hill, N. R. *et al*., Global prevalence of chronic kidney disease – a systematic review and meta-analysis. *PLoS ONE*, 2016, **11**, e0158765.
3. Forget, G., Goodman, T. and De Villiers, A. J., Impact of pesticide use on health in developing countries, International Development Research Centre, 1993.
4. Pesticide exposure of users and agricultural workers | Anses – Agence nationale de sécurité sanitaire de l'alimentation, de l'environnement et du travail; https://www.anses.fr/en/content/pesticide-exposure-users-and-agricultural-workers (accessed on 25 April 2021).
5. Six ways drones are revolutionizing agriculture. MIT Technology Review; https://www.technologyreview.com/s/601935/six-ways-drones-are-revolutionizing-agriculture/ (accessed on 1 April 2018).
6. Amarasinghe, A., Wijesuriya, V. B., Ganepola, D. and Jayaratne, L., A swarm of crop spraying drones solution for optimising safe pesticide usage in arable lands: poster abstract. In Proceedings of the 17th Conference on Embedded Networked Sensor Systems Association for Computing Machinery, New York, USA, 2019, pp. 410–411.
7. Amarasinghe, A., Wijesuriya, V. B., Ganepola, D. and Jayaratne, L., Adigar, 2019.
8. Amarasinghe, A., Wijesuriya, V. B. and Jayaratne, L., A path planning algorithm for an autonomous drone against the overuse of pesticides. In IEEE Tenth International Conference on Information and Automation for Sustainability, Institute of Electrical and Electronics Engineers Inc, Negambo, Sri Lanka, 2021, pp. 446–451.
9. Drone Flight Simulators: Your Guide to the Top 8 Drone Simulators of 2020; https://uavoach.com/drone-flight-simulator/ (accessed on 11 January 2022).
10. González, V., Monje, C. A., Moreno, L. and Balaguer, C., UAVs mission planning with flight level constraint using fast marching square method. *Robot Autonom. Syst.*, 2017, **94**, 162–171.
11. Hasanbeig, M., Abate, A. and Kroening, D., Logically-constrained neural fitted Q-iteration. In Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS International Foundation for Autonomous Agents and Multiagent Systems, Montreal QC, Canada, 2019, pp. 2012–2014.
12. Sutton, R. S. and Barto, A. G., Reinforcement learning: an introduction, by Sutton. *Trends Cogn. Sci.*, 1999, **3**, 360.
13. DJI flight simulator – DJI; https://www.dji.com/simulator (accessed on 11 January 2022).
14. 9 Best drone flight simulators for 2022 (FPV and commercial) – Droneblog.com; https://www.droneblog.com/drone-flight-simulator/ (accessed on 11 January 2022).
15. FPV Freerider by FPV Freerider.
16. Baier, C. and Katoen, J., *Principles of Model Checking (Representation and Mind Series)*, The MIT Press, USA, 2008.
17. Gould, R. J., *Advances on the Hamiltonian Problem – A Survey*, 2002.

Received 16 October 2021; revised accepted 19 February 2022

doi: 10.18520/cs/v122/i8/945-950