European Journal of Technique

journal homepage: https://dergipark.org.tr/en/pub/ejt

Vol.13, No.1, 2023

Q-Learning Based Obstacle Avoidance Data Harvesting Model Using UAV and UGV

Erdal AKIN[1,3*], **Yakup SAHIN**[2]

[1*]Bitlis Eren University, Computer Engineering Department, 13100, Merkez, Bitlis, Türkiye. (e-mail: e.akin@beu.edu.tr).
[2] Bitlis Eren University, Electrical and Electronics Engineering Department, 13100, Merkez, Bitlis, Türkiye. (e-mail:ysahin@beu.edu.tr).
[3] Malmö University, Department of Computer Science and Media Technology, Malmö, Sweden. (e-mail:erdal.akin@mau.se).

ARTICLE INFO

Received: Jan., 17. 2023
Revised: May., 22. 2023
Accepted: June., 26. 2023

Keywords:
Q-Learning, UAV, UGV, Data Harvesting, IoT

Corresponding author: *Erdal AKIN*

ISSN: *2536-5010* | e-ISSN: *2536-5134*

DOI: https://doi.org/10.36222/ejt.1237590

ABSTRACT

The Internet of Things (IoT) has revolutionized our lives by providing convenience in various aspects. However, for the IoT environment to function optimally, it is crucial to collect data from IoT devices regularly. This is because timely data collection enables more accurate evaluations and insights. Additionally, energy conservation is another crucial aspect to consider when collecting data, as it can significantly impact the sustainability of the IoT ecosystem. To achieve this, Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) are increasingly being used to collect data. In this study, we delve into the problem of how UAVs and UGVs can effectively and efficiently collect data from IoT devices in an environment with obstacles. To address this challenge, we propose a Q-learning-based Obstacle Avoidance Data Harvesting (QOA-DH) method, which utilizes reinforcement learning principles to make data collection decisions. Additionally, we conduct a comparison of the performance of UAVs and UGVs, considering the different restrictions and assumptions that are unique to each type of vehicle. This research aims to improve the overall efficiency and effectiveness of data collection in IoT environments and pave the way for sustainable IoT solutions.

1. INTRODUCTION

Internet of Things (IoT) devices benefit us from many applications in diverse areas, such as health management, military monitoring, and environmental measurements. The IoT devices collect or generate some data based on the usage purpose. In order to evaluate the data, it should be harvested periodically. For this purpose, an autonomous vehicle can visit all IoT devices and collect the data. Unmanned aerial vehicles (UAV) and unmanned ground vehicles (UGV) are well-known autonomous vehicles for data harvesting. These vehicles generally use electric and similar components. However, they also differ in many aspects, such as maneuverability, speed, battery capacity, and cost. The energy in the vehicle's battery should be sufficient to perform the abovementioned operation. Accordingly, the vehicle should be chosen according to the environment, time, and cost of data harvesting. However, energy efficiency is a significant issue for data harvester autonomous vehicles. Therefore, the best trajectory planning is needed to ensure optimum energy efficiency. Accordingly, we propose a novel framework where an autonomous vehicle must cover a maximum number of IoT devices to maximize the harvested data. Then, we propose a Q-Learning Based Obstacle Avoidance Data Harvesting method, called QOA-DH, which provides trajectories for the autonomous device to maximize the number of visited IoT devices during their first attempt (no battery replenishment).

This paper compares two unmanned vehicles, UAV and UGV, with various features detailed in Section 3. Finally, we perform a comprehensive experiment to assess the performance of the trajectories provided by the QOA-DH concerning the different features of UAV and UGV.

The remainder of this paper is structured as follows. In Section 2, we provide the literature review for Q-learning-based trajectory designs. Section 3 describes Q-learning, while the proposed QOA-DH model and assumptions are presented in Section 4. In Section 5, simulation details and performance analyses are provided. Lastly, we conclude the paper in Section 6.

2. RELATED WORKS

Data harvesting from IoT devices by an autonomous vehicle is a fast, reliable, and economical approach. The UAV is a commonly used data harvester vehicle, and it is independent of the roads and can visit IoT devices more quickly from UGV. But, the battery capacity of the UAV is limited

compared to the UGV. To evaluate an effective data collection process, optimum trajectory planning is desired. Therefore, Artificial Intelligence (AI) based trajectory planning studies have been presented in the literature [1]. Swarm UAVs can harvest data simultaneously, but trajectory planning for swarms needs more effort, time, and processing power for trajectory planning [2, 3, 4].

On the other hand, 2D path planning needs less effort with the probabilistic roadmaps (PRM) algorithms [5, 6] from 3D path planning [7]. In some research, after the swarm UAVs collect the data, the data is immediately loaded into the cloud [8, 9, 10]. These swarm data harvesters use the global system for mobile communications (GSM) to load the data to the cloud. This process leads to a time difference between sensor data.

A UGV can visit more IoT devices than a UAV because of its large battery capacity. Although the UGV has a larger battery capacity, time and efficiency should be considered during data harvesting. Consequently, trajectory planning is quite essential for the UGV. The meta-heuristic search algorithms can help to determine the optimum trajectory. Particle swarm optimizer (PSO) is another approach to planning the right trajectory [11]. Ant colony optimization method presents remarkable results in optimal path planning [12]. Near-optimal algorithm [13] gives better results than A* in a static environment. Finally, local-search-based and edge-learning systems are used to determine the efficient trajectory [14, 15].

3. REINFORCEMENT LEARNING

Machine Learning (ML) can be categorized into three techniques; supervised learning, unsupervised learning, and reinforcement learning. Reinforcement Learning (RL) differs from other techniques in attempting a specific task in a previously unknown environment [16]. In RL, an agent takes actions based on a policy to achieve a goal in an unknown environment. To do that, the agent tries to learn a policy to maximize the value and reach the specific task. As seenin Figure 1, the agent moves from the state $s_t \in S$ by taking an action $a_t \in A$ concerning the reward $r_{t+1} \in R$ obtained from the previous experiences [17, 18, 19].

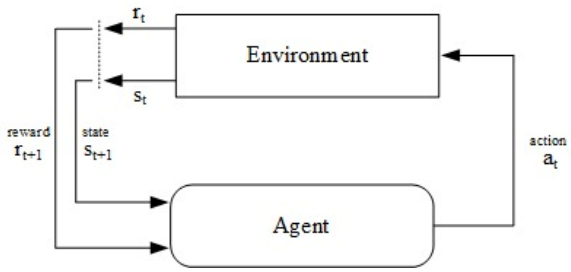


Figure 1. Reinforcement Learning Framework

There are several RL algorithms in the literature; one is the Q-learning algorithm. Q-Learning is used to find the optimal action-value function for an agent in an environment. The Q-Learning algorithm is based on the principle of learning from experience and updating the agent's knowledge based on the rewards it receives from the environment. One of the earliest works on Q-Learning was done in 1992 when they introduced the Q-Learning algorithm for learning the optimal action-value function for an agent in a finite Markov Decision Process (MDP) environment [20]. They showed that the Q-Learning algorithm converges to the optimal action-value function under certain conditions, such as the existence of a unique fixed point and the use of a sufficiently small learning rate. The actor-critic approach has been proposed to improve the Q-Learning algorithm in the following years [21]. State-Action-Reward-State-Action (SARSA) is similar to Q-Learning. Still, it uses the action selected by the agent in the next state, rather than the action with the maximum value, to update the action-value function. This makes SARSA more suitable for online learning and dealing with problems where the optimal action may not be known in advance [22]. The Q-Learning algorithms are used to solve many problems in the literature. For example, robotics are trained to avoid obstacles in unknown environments [23]. Another approach that the algorithm is used to train agents and play complex games such as Chess and Go [24]. Besides, Q-Learning is used in other applications, including finance, healthcare, IoT, and control systems [25, 26, 27, 28, 29].

Q-learning algorithm that uses a Q-table $Q(s,a)$, Where S is the set of states, and A represents the set of actions that the agent can take. In state $s \in S, Q(s,a)$ is updated with the equation (1) when the agent takes an action $a \in A$ [18, 19].

$$Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha(r_t + \gamma max_a Q_t(s_{t+1},a_t) - Q_t(s_t,a_t)) \quad (1)$$

where the agent obtains the instant reward $r_t \in R$ while moving from state s_t to state s_{t+1}. The discount factor $\gamma \in (0,1]$ stabilizes the immediate and future reward. The Learning Rate $\alpha \in (0,1]$ is a hyperparameter correlating the new value with the previous value.

4. SYSTEM MODEL and PROBLEM STATEMENT

In this section, we first introduce the system model of the proposed Q-learning Based Obstacle Avoidance Data Harvesting (QOA-DH) Model for UAV and UGV.

4.1. System Model

We consider a network composed of a Base Station (BS), which is also a Terminal Station, and a grid world ($M(X,Y)$) where the UAV or UGV can start-finish its duty from BS. We consider UAV and UGV responsible for data harvesting from IoT devices, which are yellow circles in Fig. 2a. Our goal is to maximize the total number of visited IoT devices with minimum battery consumption and time interval. For this purpose, we use a grid-based map, as seen in Figure 2a, where the red cell represents BS, yellow circles are IoT devices, and black cells are obstacles (a.k.a. walls). Therefore, we formulate our QOA-DH Model as a tuple $< V, S, A, R, \gamma, \alpha >$ represents the type of the vehicle (agent) V, the set of states S, the set of actions spaces A, reward function $R(s,a)$, the future discount factor γ, and learning rate α, respectively. An agent (UAV or UGV) selects an action $a_t \in A$ in time t based on its state $s_t \in S$, which is the location of the agent in time t. The action set A includes up, down, left, and right in the current state of the agent. In each step, the agent consumes the battery as follows:

$$\text{for } a_t \in A, B_t(V) = B_t(V) - \eta(V) \quad (2)$$

where $B_t(V)$ is the remaining battery of the agent at time t and $\eta(V)$ represents battery consumption in each step a_t. The battery of the agent is 100 at the beginning. The battery consumption differentiates concerning the technology used for the vehicles. Since high-capacity batteries cause excessive load on UAVs, batteries with low capacity are used. Therefore, UAVs consume more battery than UGVs for the same number of actions.

When an agent takes action, it earns a reward (or punishment) based on the Reward function, an essential parameter for optimizing the algorithm. Since we have two different agents, we carefully present the reward function as follows:

$$R_{s,a}(V) = \begin{cases} I(V) * C, if\ s' \epsilon\ G\ \{obstacle\} \\ O(V) * C, \quad Otherwise \end{cases} \quad (3)$$

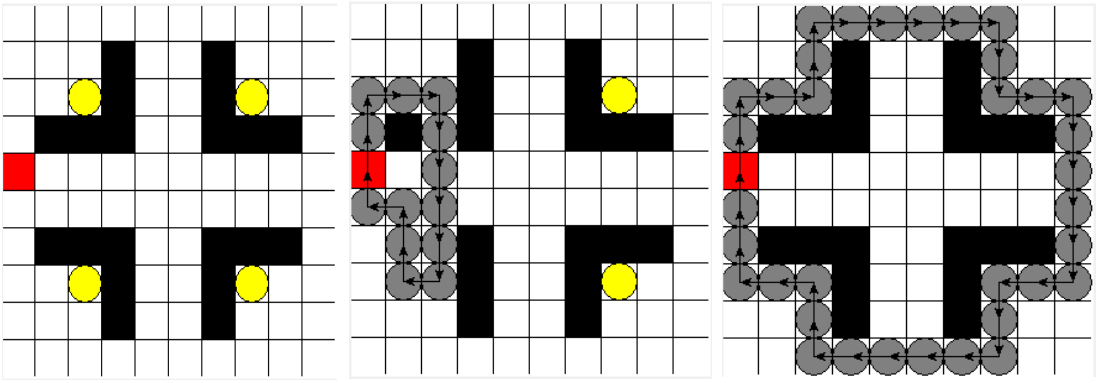
Where $s' \in S$ represents the next state of the agent, C denotes a constant factor, $I(V) \in [1,3]$ and $I(V) \in [-15,0]$ are functions that return a factor concerning the type V of the vehicle for instant reward.

Another two important parameters are the discount factor and learning rate. Discount factor $\gamma \in (0,1]$ balances immediate and future rewards to adjust the greedy of the policy. The learning rate $\alpha \in (0,1]$ is the fitness between the previous and new values [19, 30]. In section 5, we evaluated the algorithms with $\gamma = 0.5$ and $\alpha = 0.15$.

Algorithm 1: Environment setup of Q-learning Based Obstacle Avoidance Data Harvesting (*QOA-DH*) Model.

Initialization: Grid world $M(X,Y)$, Base Station (is also starting position) $BS(X,Y) \in M$, battery $B_t = 100$, Vehicle Type (V), Action set $A = \{up, down, left, right\}$, Reward Table R, Discount Factor γ, Learning Rate α, Initial Q-table $Q_t[s,a] = 0$, Episode Number $E = 1000000$, $\epsilon = 1.0$, and decay value $d = 0.000001$.

```
foreach E do
    Reset R.
    Reset Environment.
    while True do
        t++.
        Reduce battery of the V as in equation (2).
        Obtain reward as in equation (3).
        if random ≤ ε then
            select action a_t ∈ A, randomly .
        else
            Select action a_t = argmax_a(Q(s,a))
        end
        if B_t = 0 ∨ s = BS(X,Y) then
            break.
        end
        Update Q-table as in Equation 1.
        Update Reward Table R as in Equation 3.
    end
    ε = ε − d.
end
```



(a) Initial Small Map (b) UAV (c) UGV

Figure 2. Portraying the initial small map and final trajectories of the vehicles; (a) Initial small map, (b) UAV trajectory, and (c) UGV trajectory. Arrows show the direction of the vehicles, the red square is the base station, and the gray circles present observed areas.

4.2 Methodology

Our primary goal in this study is to visit a maximum number of IoT devices to maximize harvested data. However, there are restrictions and assumptions that Unmanned Vehicles (UAV and UGV) need to adhere to:

- UAV and UGV have limited battery capacity.
- Both UAV and UGV do not charge their battery and fulfill their task with only one battery.
- Battery consumption differs for UAV and UGV. It is calculated with Equation 2, and the parameters are given in Table 1.
- UAV flies at a certain altitude so that it can fly over obstacles. UGV should avoid the obstacles.

4.3 Problem Statement

This section presents our proposed QOA-DH Model setup in Algorithm 1.

The vehicle (agent) starts in the base station $BS(X,Y) \in M(X,Y)$. The algorithm initializes the Q-table $Q(S,A)$ with zero at the beginning and updates it in each step as in Equation 1. The reward table R is filled based on Equation 3. In each step, the vehicle's battery is reduced using Equation 2. A new episode starts over if the battery runs out or the agent reaches the terminal state (same as BS). In each episode, the obtained trajectory and the total reward are stored.

5. SYSTEM MODEL and PROBLEM STATEMENT

In this section, we compare the UAV and UGV on the QOA-DH method considering energy efficiency, number of observed IoT devices, and spent time to accomplish the task.

5.1 Simulation Setting

We developed the QOA-HA with Python 3.9 version using NumPy, Tkinter, gym, and collection libraries. As illustrated in Fig. 2a and Fig. 3a, we have a $10x10$ and a 20x20 grid environment. The red cell is BS, where the agent (UAV or UGV) starts and ends its duty. Black cells are obstacles in the environment.

Since the UAV can fly over the obstacles $O(V)$ parameter is given as 0, it is -15 for forcing the UGV to avoid the obstacles. The yellow circles are IoT devices the agent is responsible for collecting data. $I(V)$ is the parameter to provide a reward for directing the agent through these cells. We train each vehicle on the QOA-DH in 1000000 episodes (E). Battery consumption η(V) is chosen as 6 and 3 for UAV in small and big maps, respectively, while it is 2 and 1. The hyperparameter used in the simulation is summarized in Table I.

TABLE I

Primary hyperparameters for QOA-DH.

	SMALL MAP (10x10)		BIG MAP (20X20)	
Parameter	UAV	UGV	UAV	UGV
Grid G	10x10	10x10	20x20	20x20
Episode E	1000000	1000000	1000000	1000000
Epsilon ε	1.0	1.0	1.0	1.0
Decay Value d	0.000001	0.000001	0.000001	0.000001
Discount Rate γ	0.5	0.5	0.5	0.5
Learning Rate α	0.15	0.15	0.15	0.15
$B_t(V)$	100	100	100	100
$\eta(V)$	6	2	3	1
C	1000	1000	1000	1000
$I(V)$	3	1	3	1
$O(V)$	0	-15	0	-15

The snapshot of the computed trajectories of the UAV and UGV are represented in Fig. 2. Grey circles show the observed cells (trajectories), and arrows represent the direction of the UVs. As seen in Fig. 2b and 2c, the UAV flies over a small area of the environment, while the UGV visits all IoT devices. This is because the UAVs have different battery constraints and assumptions. Therefore, we compare UAV and UGV on

QOA-DH model by using the following performance metrics:

- The ratio of Observed IoT Devices (ROIoT) (%): This is the ratio of the number of observed IoT devices.
- Battery Consumption (BC) (%): This is the ratio of the battery consumption, which is calculated using Equation 2, of the agents.
- Spent Time (ST) (Time Unit): This is the time the agent spends completing its duty for only one battery capacity without replenishment.

5.2 Evaluation

We evaluate the performance of the vehicles with diverse battery consumption parameters in two different size maps. The small map has a 10x10 grid, and the big map has a 20x20 grid.

5.2.1 Small Map Evaluation

As seen in Fig. 2a, four IoT devices are located behind the walls in the 10x10 grid map. The UGV visits and collects data from all IoT devices (100%) in the environment, as seen in Fig. 3. Conversely, the UAV can only fly over half of the IoT devices, even if it can fly over obstacles. The reason behind this is that the UAV consumes more energy for its movement. Accordingly, when we evaluate the battery consumption of the vehicles in Fig. 4, the battery consumption is around 84% for the UAV and around 72% for the UGV.

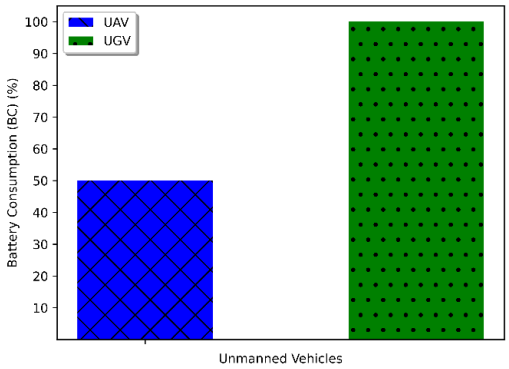


Figure 3. The ratio of Observed IoT Devices (ROIoT) (%) in Small Map.

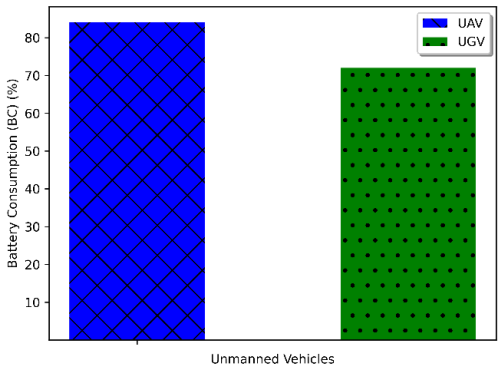


Figure 4. Battery Consumption (BC) (%) in Small Map.

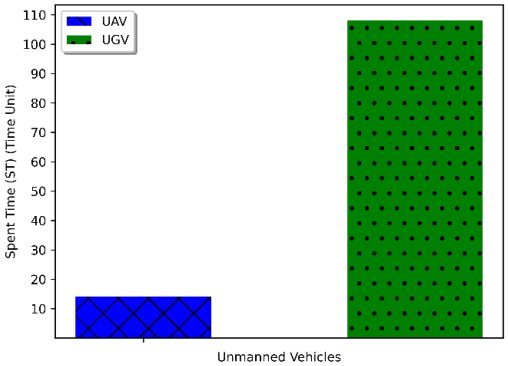
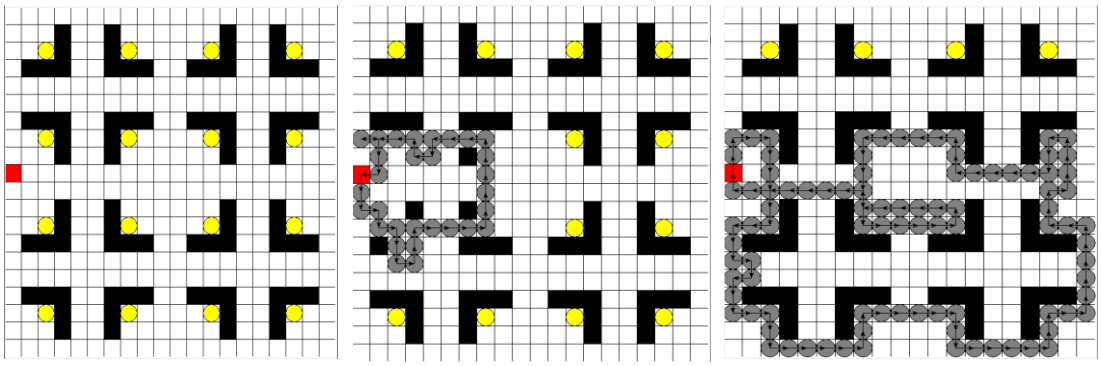


Figure 5. Spent Time (ST) (Time Unit) in Small Map.



(a) Initial Big Map (b) UAV (c) UGV

Figure 6. Portraying the initial big map and final trajectories of the vehicles; (a) Initial big map, (b) UAV trajectory, and (c) UGV trajectory. Arrows show the direction of the vehicles, the red square is the base station, and the gray circles present observed areas.

Obviously, the UGV accomplishes its task with better battery efficiency than the UAV. The UAV should replenish its battery (more than one time) to complete the task.

On the other hand, time efficiency is another metric we should consider. In the scenario of needing fast data collection from IoT devices, the UGV would fail. This is because the UGV completes its duty in around 110 time units, while visiting two IoT devices takes around 15 time units for the UAV. This results in 10 times faster than the UGV. However, if we consider the battery recharging of the UAV, it would take more time than the UGV to achieve the task. Therefore, using multiple batteries or UAVs is not a cost-effective way.

5.2.2 Big Map Evaluation

In this section, we evaluated the performance of the devices in a 20x20 grid map. Assuming that battery usage is 1% of the remaining battery of the UGV and 3% for the UAV on the map. In this map (Fig. 6a), there are 16 IoT devices located behind the walls. The trajectories of UAV and UGV are given in Fig. 6b, Fig. 6c, respectively. Since the map is greater than the previous scenario, both vehicles could not visit all IoT devices. However, as seen in Fig. 7, UGV is able to visit 75% of the devices (12 IoT devices), while UAV can only visit 25% of them (4 IoT devices).

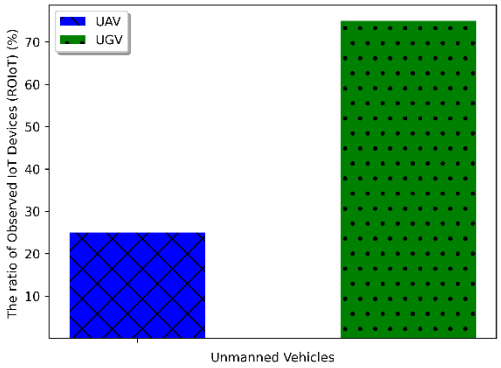


Figure 7. The ratio of Observed IoT Devices (ROIoT) (%) in Big Map.

Due to the large size of the map, both vehicles try to use almost all of their batteries to complete their task, Fig. 8. Since both need to return the base station (BS) before the battery is completely running out, the UAV completes its trajectory with 94% battery consumption. The remaining 6% battery would not be enough to detour to visit another IoT device. On the other hand, UGV uses all its battery to reach the BS.

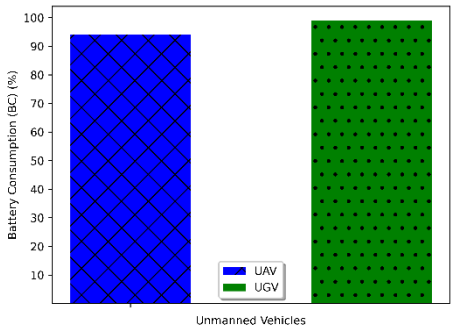


Figure 8. Battery Consumption (BC) (%) in Big Map.

Another important metric is the spent time (ST) to complete the task. As seen in Fig. 9, like the small map, the UAV completes its flight 10 times faster than the UGV with fewer visited IoT devices. Again, if fast data collection is crucial, this issue can be solved by using multiple UAVs. Otherwise, multiple batteries replenishing with one UAV would be time-consuming because the UAV needs to fly to the BS for battery replenishment and take off to visit remaining IoT devices.

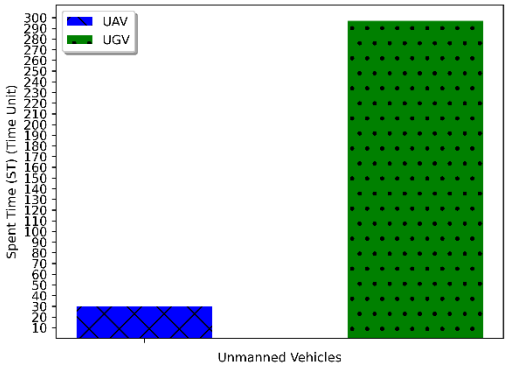


Figure 9. Spent Time (ST) (Time Unit) in Big Map.

6. CONCLUSION

In the event of data harvesting from IoT devices, Unmanned Vehicles (UVs) are widely used. Unmanned Aircraft Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) are well-known vehicles used for this purpose. Accordingly, in this study, we propose a Q-learning Obstacle Avoidance Data Harvesting (QOA-DH) framework, where we train the vehicles concerning their battery restriction and movement velocity. The results demonstrate that the UGV is more battery effective and can visit all IoT devices to harvest data in the environment. On the other hand, the UAV is faster than the UGV, giving it an advantage in achieving minor duties in a short time. However, since the UAV consumes battery faster, it should recharge its battery, which is not time efficient. To overcome this issue, we can use multiple batteries for replenishing or UAVs interchangeably, using each other to accomplish the task. For all that, both solutions are not cost-effective. Therefore, to overcome the issues and achieve the task for our future work, we plan to extend this work to propose a cost, energy, and time-efficient hybrid approach that uses UAV and UGV together.

REFERENCES

[1] Z. Qin, X. Zhang, X. Zhang, B. Lu, Z. Liu, and L. Guo, "The uav trajectory optimization for data collection from time-constrained iot devices: A hierarchical deep q-network approach," Applied Sciences, vol. 12, no. 5, pp. 2546, 2022.

[2] Bithas, P.S., Michailidis, E.T., Nomikos, N., Vouyioukas, D. and Kanatas, A.G., 2019. A survey on machine-learning techniques for UAV-based communications. Sensors, vol. 19, no. 23, pp.5170.

[3] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "Multi-UAV path planning for wireless data harvesting with deep reinforcement learning," IEEE Open Journal of the Communications Society, vol. 2, pp. 1171–1187, 2021.

[4] Y. Yao, Z. Zhu, S. Huang, X. Yue, C. Pan, and X. Li, "Energy efficiency characterization in heterogeneous iot system with uav swarms based on wireless power transfer," IEEE Access, vol. 8, pp. 967–979, 2019.

[5] Z. Wang and J. Cai, "Probabilistic roadmap method for path-planning in radioactive environment of nuclear facilities," Progress in Nuclear Energy, vol. 109, pp.113–120, 2018.

[6] A. Upadhyay, K. R. Shrimali, and A. Shukla, "Uav-robot relationship for coordination of robots on a collision free path," Procedia Computer Science, vol. 133, pp. 424–431, 2018.

[7] F. Yan, Y.-S. Liu, and J.-Z. Xiao, "Path planning in complex 3d environments using a probabilistic roadmap method," International Journal of Automation and computing, vol. 10, no. 6, pp. 525–533, 2016.

[8] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy efficient data collection in wireless sensor networks," Mobile networks and Applications, vol. 11, no. 3, pp. 327–339, 2006.

[9] C.S. Choi and F. Baccelli, "Spatial and temporal analysis of direct communications from static devices to mobile vehicles," IEEE Transactions on Wireless Communications, vol. 18, no. 11, pp. 5128–5140, 2019.

[10] R. Marini, L. Spampinato, S. Mignardi, R. Verdone, and C. Buratti, "Reinforcement learning-based trajectory planning for uav-aided vehicular communications," 2022 30th European Signal Processing Conference (EUSIPCO), IEEE, pp. 967–971, 2022.

[11] A. Kaplan, N. Kingry, P. Uhing, and R. Dai, "Time-optimal path planning with power schedules for a solar-powered ground robot," IEEE Transactions on Automation Science and Engineering, vol. 14, no. 2, pp. 1235–1244, 2006.

[12] M. R. Jabbarpour, H. Zarrabi, J. J. Jung, and P. Kim, "A green ant-based method for path planning of unmanned ground vehicles," IEEE access, vol. 5, pp. 1820–1832, 2017.

[13] E. Almoaili and H. Kurdi, "Path planning algorithm for unmanned ground vehicles (ugvs) in known static environments," Procedia Computer Science, vol. 177, pp. 57–63, 2020.

[14] Y.-C. Wang and K.-C. Chen, "Efficient path planning for a mobile sink to reliably gather data from sensors with diverse sensing rates and limited buffers," IEEE Transactions on Mobile Computing, vol. 18, no. 7, pp. 1527–1540, 2018.

[15] D. Liu, S. Wang, Z. Wen, L. Cheng, M. Wen, and Y.-C. Wu, "Edge learning with unmanned ground vehicle: Joint path, energy, and sample size planning," IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2959–2975, 2020.

[16] A. T. Azar, A. Koubaa, N. Ali Mohamed, H. A. Ibrahim, Z. F. Ibrahim, M. Kazim, A. Ammar, B. Benjdira, A. M. Khamis, I. A. Hameed, et al., "Drone deep reinforcement learning: A review," Electronics, vol. 10, no. 9, pp. 999, 2021.

[17] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 26–38, 2017.

[18] C. J. C. H. Watkins, Learning from delayed rewards. King's College, May 1989.

[19] M. Kusy and R. Zajdel, "Stateless Q-Learning algorithm for training of radial basis function based neural networks in medical data classification," Advances in Intelligent Systems and Computing, vol. 230, pp. 267–278, 2014.

[20] C. J. C. H. Watkins and P. Dayan, "Q-learning," Machine learning, vol. 8, pp. 279-292, 1992.

[21] S. Bhatnagar, R. S: Sutton, M. Ghavamzadeh, and M. Lee, "Incremental Natural Actor-Critic Algorithms," NeurIPS, pp. 729-736, 1993.

[22] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," MIT press, 2018.

[23] Y. Hu, D. Li, Y. He, and J. Han, "Incremental Learning Framework for Autonomous Robots Based on Q-Learning and the Adaptive Kernel Linear Model," in IEEE Transactions on Cognitive and Developmental Systems, vol. 14, no. 1, pp. 64-74, March 2022.

[24] V. Mnih, et al, "Human-level control through deep reinforcement learning. Nature, vol. 518, no. 7540, pp. 529-533, 2015.

[25] J. Zhu, Y. Song, D. Jiang and H. Song, "A New Deep-Q-Learning-Based Transmission Scheduling Mechanism for the Cognitive Internet of Things," in IEEE Internet of Things Journal, vol. 5, no. 4, pp. 2375-2385, Aug. 2018.

[26] X. Zhou, W. Liang, K. I. -K. Wang, H. Wang, L. T. Yang and Q. Jin, "Deep-Learning-Enhanced Human Activity Recognition for Internet of Healthcare Things," in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6429-6438, July 2020.

[27] A. Weissensteiner, "A Q-Learning Approach to Derive Optimal Consumption and Investment Strategies," in IEEE Transactions on Neural Networks, vol. 20, no. 8, pp. 1234-1243, Aug. 2009

[28] M. Ye, C. Tianqing and F. Wenhui, "A single-task and multi-decision evolutionary game model based on multi-agent reinforcement learning," in Journal of Systems Engineering and Electronics, vol. 32, no. 3, pp. 642-657, June 2021

[29] Z. Xiaochuan, W. Wanwan, L. Qin, W. Tianao and S. Hao, "The Design and Realization of Dynamic Evaluation Strategy of Pieces in Military Chess Game System," 2019 Chinese Control And Decision Conference (CCDC) pp. 6287-6292, Nanchang, China, 2019.

[30] T. N. Larsen, H. Ø. Teigen, T. Laache, D. Varagnolo, and A. Rasheed, "Comparing deep reinforcement learning algorithms' ability to safely navigate challenging waters," Frontiers in Robotics and AI 8.

BIOGRAPHIES

Erdal AKIN was born in Ceyhan, Adana, Turkey. He graduated from the Department of Mathematics, Yildiz Technical University, Istanbul, in 2008. In 2009, he received the Republic of Turkey Ministry of National Education scholarship to study in the USA. He received master's and Ph.D. degrees from the Department of Computer Science, The University of Texas at San Antonio (UTSA), in the Spring of 2014 and December 2018, respectively. His research interests are Deep Reinforcement Learning, Software Defined Networks, Security, and Blockchain.

Yakup SAHIN was born in Adana, Turkey, in 1986. He received his B.S. degree in electrical and electronics engineering from İnönü University, Malatya, Turkey, in 2010, and his M.S. and Ph.D. degrees in electrical engineering from Yıldız Technical University, Istanbul, Turkey, in 2013 and 2016, respectively. He worked as a Research Assistant from 2011 to 2016 in the Department of Electrical Engineering at Yıldız Technical University. He served as an Assistant Professor from 2016 to 2021 and is currently an Associate Professor in the Department of Electrical and Electronics Engineering at Bitlis Eren University. He has authored or co-authored more than 50 technical papers published in journals and conference proceedings. Additionally, he has been involved in 9 research projects related to power electronics. His current research interests include DC-DC and DC-AC power converter topologies, high-frequency soft-switching converters, power factor correction, switching power supplies, and bidirectional converters.