MQTPP – Towards Multiple Q-Table based Path Planning in UAV Environments

Michael R. Jones, Soufiene Djahel, and Kristopher Welsh
Department of Computing and Mathematics
Manchester Metropolitan University, UK
{michael.jones9@stu.mmu.ac.uk, (s.djahel, k.welsh)@mmu.ac.uk}

***Abstract*—This paper introduces an original multi destination path planning approach for Unmanned Aerial Vehicles (UAVs) named MQTPP (Multi Q-Table Path Planning). MQTPP aims to reduce the computational burden of cyclical/continuous path planning through a Q-learning planning process whilst overcoming the fixed path origin problem. The preliminary performance evaluation results indicate that MQTPP performs well for longer paths, and allows for more efficient re-planning should mission objectives or environmental topography change.**

***Index Terms*—UAVs, Path Planning, Q-learning**

I. Introduction

The diversity of problems to which advances in Unmanned Aerial Vehicle (UAV) research may be applied drives significant interest in the UAV research space. As UAV technology progresses further from its original military origins, burgeoning commercial and consumer markets display a clear interest in the use of UAVs in scenarios such as: consumer delivery [1], communications [2] and agricultural processes [3]. Tied with the drive for carbon-neutral infrastructure unmanned aerial service platforms prospectively offer the capability of fulfilling roles once handled solely by larger, heavier, and less energy-efficient vehicles, whilst also gaining the significant associated cost reductions in physical manpower and infrastructure for their individual market gains.

The effective path planning of UAVs raises new challenges in control and planning design, with traditional movement planning constraints becoming irrelevant within a UAV's three-dimensional operational space. However, the associated increases in freedom of movement are shared within the environment space by all associated operational objects, resulting in greater challenges in ensuring UAV coordination and collision free movement. In general, we distinguish between two variants of the UAV path planning problem: a lower-level *individual* UAV path planning problem, focused on the generation of an obstacle-free path between to distinct locations within an environment, and a higher-level *collective* planning problem, which is a variation on existing Travelling Salesman and Vehicle Routing Problems focused on the efficient use of a fleet of UAVs to achieve multiple objectives spread within the environment. Solution approaches to the collective planning problem vary greatly [4], requiring consideration of wider mission constraints and optimisation objectives. In this paper, we consider only the individual UAV path planning problem.

A typical path planning process is dependent upon a solid contextual understanding of the operational environment, enabling a complete path to be planned from start to finish as a single planning event. Conversely, many UAV usage scenarios feature potentially unanticipated changes in target destination and/or the potential for the presence, location, size, or shape of environmental obstacles to change during operation. Such dynamic environments present a planning problem that cannot be defined as a single planning event, because the availability of planned paths may change periodically during flight path execution. This does not, however, mean a planning process cannot distinguish between what is static and known and what may be dynamic and unknown. Knowledge gained from static and known objects and obstacles can be considered fixed over time, influencing a planning process beyond a single planning event. Introducing a hybridised off-line and on-line planning approach, [5] initially plans core paths using a static representation of the environment, whilst in-flight a UAV may re-evaluate the core path whenever a dynamic or unknown obstacle is encountered. Thus, the path planning becomes a recurring process based upon the in-flight proximal interpretation of a UAV's environment. Such continuous planning greatly increases the computational processing power required for path planning, which presents a valid concern due to the limited nature of UAVs' computational and battery power.

II. Q-learning based Path Planning: overview and limitations

Following a traditional reinforcement learning based Q-learning methodology [6], the process of learning is conducted upon an environment containing a single origin and single target location. Given a sufficiently populated Q-table (improved through learning iterations) an optimal path can be planned between the two locations as illustrated in Fig. 1.

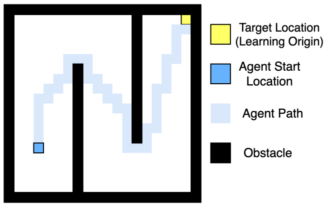

Fig. 1. Single Q-table path generation.

Where a sufficient level of exploration of the environment space exists, a single Q-table can generate a path between any given environment location and its learning origin, without the need to relearn the environment. However, there exists a problem with a single Q-table approach: because all learning is focused through a single environment point, so too must all path planning. Thus, if such an approach is extended to apply to multi-destination problems, all planned paths must transit through the learning origin. This limitation significantly reduces the applicability of single Q-table based planning approaches to collective UAV routing problems. Seeking to overcome this multi-destination problem, [7] first optimises a visitation sequence for target destinations, before applying the Q-learning planning process to each destination pair within that visitation sequence. This allows for point-to-point planning at the expense of computational power and time, with each destination pair requiring an individual Q-learning training process to derive a path solution.

In this paper, we introduce an original multi-destination path planning approach named MQTPP (**M**ulti **Q**-**T**able **P**ath **P**lanning). The approach targets both the efficiency concerns of cyclical/continuous path planning and the fixed path origin problem, and will be the focus of the remainder of this paper.

III. Multiple Q-Table Path Planning

The MQTPP approach provides the ability to reference the knowledge of multiple individual Q-tables, with each table pertaining to the same environment space, upon which an individual learning process has been conducted. However, the environment perspective for each learning process is contrasted. The main phases of MQTPP are described below.

A. Initial Learning Process

To deliver a plannable environment knowledge base MQTPP conducts four separate reinforcement learning processes upon a single grid-based environment. We assume that during the process of learning the environment remains static in nature, consisting of predefined rigid infrastructure i.e., buildings. Upon this environment four separate Q-learning instances are deployed, targeting the NW (North West), NE (North East), SE (South East), and SW (South West) cell of the grid, each forming a separate learning origin. This results in the generation of four individual Q-tables, each with a varying learned perspective of the environment based on their NW, NE, SE, SW cell origin.

1) Exploration vs. Exploitation: Numerous approaches seek to develop an exploration vs. exploitation relationship over the lifetime of the learning process, such as random walks, exponential decay and reward based decay [8]. However, the concept of ε used as a tuneable hyper-parameter influencing the agent's learning is key to environment exploration. One of the most common exploration strategies is ε-greedy [9] which offers a simplistic balance between choosing the current best Q-table value, whilst selecting a random action with some small probabilistic frequency ε. Whilst a Q-table that can exploit the given environment to generate an optimal path is desirable, given the path planning nature of this problem, we must also consider the complete exploration of the environment space as a critical component in allowing multiple paths to be created across the environment. Therefore, we propose an approach that combines both Q-masking and agent location randomisation (ALR). Q-masking seeks to limit early termination of learning episodes through masking the exploitation possibilities of the Q-table when in close proximity to an obstacle, aiming to aid exploration at the edges of an environment grid space. ALR randomises the starting location of the agent for each learning episode, thus seeking to avoid learning "black spots" within the environment grid space, often formed when an agent converges upon an optimal path with limited environment exploration.

B. Path Generation

The use of multiple Q-tables allows opposing NW, NE, SE, SW paths to be formed from the differing environment perspectives. Each path consists of a series of grid locations from target to destination. The paths from opposing Q-tables naturally intersect when considered as operating within a single environment. The natural creation of path intersection points illustrated in Fig.2, facilitates the merging of paths on collision free routes between multiple environment locations.

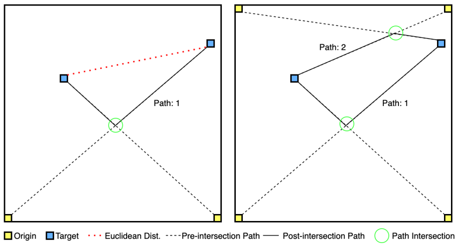

Fig. 2. Multi Q-table path generation

Whilst MQTPP does not guarantee an optimal route (see the red dashed line in Fig.2), the use of four learning origins allows the evaluation of multiple collision free paths. In contrast to single Q-table use, freedom is now given to plan paths between two environment locations without revisiting a learning origin.

Considering a wider environment picture MQTPP also offers a structure of distinct navigation points. The curse of dimensionality within Q-learning's action and state space coined by Bellman [10], ultimately applies a limit to the overall effective dimensions of the environment space. Thus, MQTPP environment construction presents the future ability to conjoin multiple environment cells at points of learning origin, such that UAVs can possess or share only the subsections of Q-table environment knowledge they require for an operational task.

IV. Performance Evaluation

To evaluate the performance of MQTPP we will compare it against an existing path planning algorithm (A*) in terms of the perceived path generation time across a standardised simulated environment.

A. Evaluation Metrics

The evaluation metric chosen for MQTPP is based upon a continuous planning scenario whereby after each step taken within an environment, a UAV would be required to evaluate its current position. Path re-planning would be required if an obstacle blocks the UAVs current path. The worst-case scenario of such a situation would require a UAV to re-plan its path after every step taken. Therefore, this is the assumed testing metric for comparison of MQTPP against the A* algorithm. Given the focus on the limited computational power available to a UAV, the MQTPP learning process is not directly considered for comparison with the A* algorithm. It is recognised, however, that if a single A* planning episode was compared directly against the combined MQTPP learning process and planning episode, A* would significantly outperform MQTPP due to the computational burden of the required learning process. Thus the MQTPP learning process is considered as an off-line event completed *a priori*, with this evaluation considering a UAV's on-line planning abilities.

We focus on two key evaluation experiments, firstly a direct comparison between MQTPP and A*, evaluating the comparative path planning computation times over a single journey within the environment space. Secondly, a comparison of computational time where an increasing number of random locations are allocated to the UAV.

B. Evaluation Setup

The evaluated UAV path planning problem is defined as a 2D stochastic maze environment of grid size 34x34 within which the Q-learning agent can transit through. The maze defines the environment through three reward states. When the Q-learning agent transitions in to *free*, *obstacle* or *goal* space, a reward of *-1*, *-100*, or *200* is issued respectively. Movement of the Q-learning agent within the environment is restricted to a Von Neumann neighbourhood, meaning it maintains only four directions of freedom i.e., North, East, South and West. The MQTPP relies solely on its four Q-tables for path planning after the initial learning process has concluded, Whereas the A* algorithm maintains continuous access to a static representation of the environment.

C. Evaluation Results Analysis

1) Q-masking Comparison: The principal of Q-table is derived from the concept of UAVs being required to work in real world dynamic environments where it is envisaged that LiDAR or ultrasonic [11] sensors would offer a perception mechanism for a UAV's immediate environment. Therefore, when an obstacle is detected in close proximity, Q-table masking serves to block any action within the Q-table that could cause a collision, thus forcing action selection to be made from the remaining unmasked actions within the Q-table.

When masking is applied within the learning process two distinct outcomes can be observed, Fig.3 illustrates a significant increase in the path success rate for epsilon values ε <0.75. Similarly, Fig.4 demonstrates the effective reduction in the obstacle collision rate when Q-table masking is introduced. Masking reduces the collision rate to zero, in doing so this initially leads to an increase in timeout steps exceeded state. However, the action of keeping the agent within the environment for an increased number of steps significantly reduces the number of episodes required to reach the target goal by 60%. Whilst the rate at which the agents learn from the environment had been significantly improved, Fig.3 still shows a lack of completeness with non of the exploration rates applied achieving 100% path success.

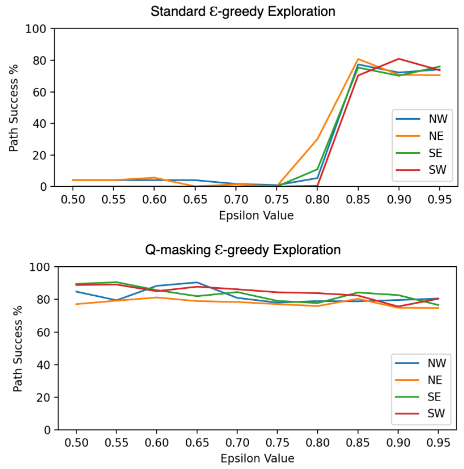

Fig. 3. Q-masking Path Success Rate - 10000 episodes

2) ALR Comparison: The combination of both a randomised starting location and Q-table masking for collision avoidance offers further improvement in learning ability. Applying both approaches together in Fig. 5, along with an arbitrary exploration value $\varepsilon = 0.9$ (10% exploration rate), accelerates the stabilisation of the average learning rate by 30%, compared to when Q-table masking is solely applied in Fig. 3, with exploration achieving 100% path success.

3) Computational Comparisons: To compare the computational efficiency of MQTPP against A*, a single UAV journey is selected from environment location (1,1) to (32,32), with the UAV returning to its base location (1,1). A direct path between the locations encounters four obstacles within the environment. After each path step the UAV must instigate the path planning process to generate its next location step within the environment, the average process time for each MQTPP and A* planning procedure is recorded in Fig.6. The planned path consists of a total 124 steps across the environment, 62 for each path leg. During this period MQTPP maintains a near static path calculation time for all steps taken. In contrast the A* path calculation time decreases as it steps over the environment moving closer to the target location. To evaluate MQTPP in a multi-location scenario we compare an increasing number of locations (1-20) based on the Manhattan distance separation between them. Where the location separation exceeds 20 steps MQTPP serves to reduce overall path processing time. Whilst when the Manhattan

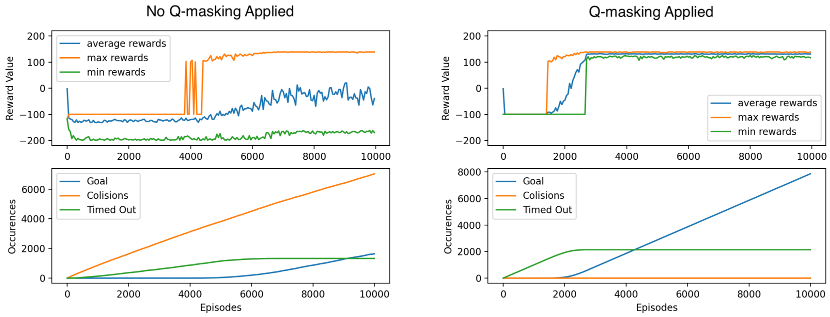

Fig. 4. Q-masking Comparison - 10000 episodes

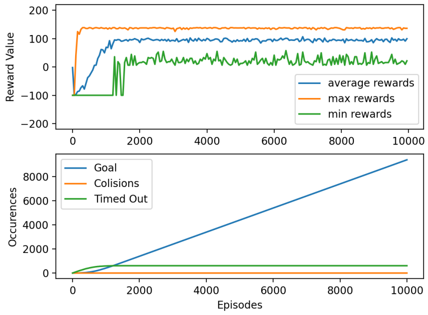

Fig. 5. ALR with Q-masking - 10000 episodes.

distance between locations is less than 20 steps the A* algorithm offers a more efficient processing solution Fig7. Significantly, whilst showing A* is dominant when searching in close proximity to a target (grid environment<10x10) as the environment search space for A* expands so to does the path calculation time. Thus, for environments where paths traverse greater distances MQTPP demonstrates a noteworthy reduction in path calculation time.

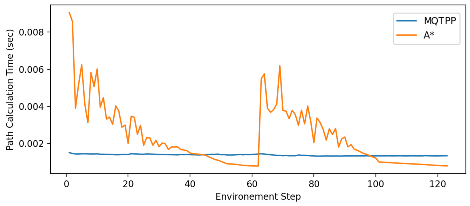

Fig. 6. Path calculation time comparison: MQTPP vs. A*

V. Conclusion

This paper introduced an original path planning method for UAVs named MQTPP. Compared to A*, MQTPP offloads its computational burden to its learning process. Once learning is complete, MQTPP efficiently references multiple Q-table knowledge for path planning tasks. In contrast, A* becomes computationally burdened as the environment space grows and repetitive path planning tasks are encountered. The future of MQTPP offers the scope to mesh the wider environment, whilst improving the Q-masking metric to influence Q-value

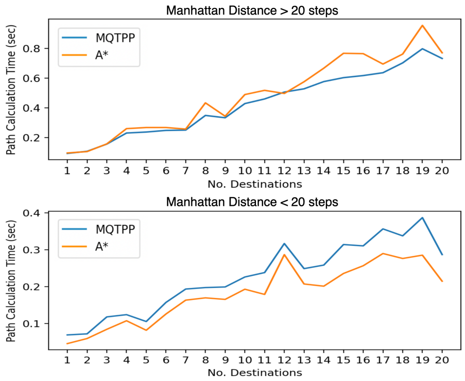

Fig. 7. Multi-destination Location Distance Comparison

selection based upon varying proximity to obstacles. As UAV processing power is limited, MQTPP's use of a prior learning step to offer lightweight and efficient re-planning offers significant potential for use in UAV path planning problems.

References

[1] D. Schermer, M. Moeini, and O. Wendt. A matheuristic for the vehicle routing problem with drones and its variants. *Transportation Research Part C: Emerging Technologies*, 106:166–204, 2019.

[2] H. Qi et al. Energy efficient 3-d uav control for persistent communication service and fairness: A deep reinforcement learning approach. volume 8, pages 53172–53184, 2020.

[3] P. Lottes et al. Uav-based crop and weed classification for smart farming. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3024–3031, 2017.

[4] B. Eksioglu et al. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483, 2009.

[5] C. Yin et al. Offline and online search: Uav multiobjective path planning under dynamic urban environment. *IEEE Internet of Things Journal*, 5(2):546–558, 2018.

[6] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[7] H. Zhuang et al. Multi-destination path planning method research of mobile robots based on goal of passing through the fewest obstacles. *Applied Sciences*, 11(16):7378, 2021.

[8] Aakash Maroti. Rbed: Reward based epsilon decay. *arXiv preprint arXiv:1910.13701*, 2019.

[9] A. D. Tijsma et al. Comparing exploration strategies for q-learning in random stochastic mazes. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 2016.

[10] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[11] C. Wang et al. Autonomous navigation of uavs in large-scale complex environments: A deep reinforcement learning approach. volume 68, pages 2124–2136, 2019.