In: 2024 International Conference on Unmanned Aircraft Systems (ICUAS), 04-07 June 2024, Crete, Greece
DOI: 10.1109/ICUAS60882.2024.10557095

Disaster Area Coverage Optimisation using Reinforcement Learning

Ciaran Gruffeille, Adolfo Perrusquía, Antonios Tsourdos, and Weisi Guo

***Abstract*— Search and rescue applications using unmanned aerial vehicles (UAVs) also known as drones are becoming a research topic of interest to industry and academia due to its high impact in the ecosystem and people. Exploration of the disaster area is a crucial element in search and rescue to identify the zones that require immediate assistance or with high hazard probability. This paper aims to contribute in the coverage optimisation of a disaster area using drones. We focus on a flood disaster scenario as case of study. The proposed approach consists in two main parts: i) a Siamese-net is used to identify flooded buildings in satellite images, and ii) the points of interest are converted into a suitable maze environment that subsequently is used by any reinforcement learning (RL) architecture for area coverage optimisation. Here, the goal of the RL architecture is to ensure that the complete environment is covered by the drone by optimizing time and previously visited zones. Experiments are conducted to show the benefits and challenges of the current approach.**

I. Introduction

Search and rescue applications using drones are increasing its relevance in the last decade due to climate change and the increasing number of natural disasters across the world [1]. Here, drones provide a low-cost supporting tool in disaster scenarios due to its versatility to fly under inaccessible areas and challenging geological conditions for humans [2]. Additionally, drones are equipped with a variety of sensors and cameras [3], [4] that permit identifying effectively affected areas and/or people that need immediate help. Currently, researchers are interested in providing drones with a high-level of autonomy such that they are capable to cover a disaster area optimally, assist people, and/or request help. To this end, several algorithmic tools have been created to increase the autonomy of the drone [5].

Path planning is a well-known research topic concerning to the autonomous systems research area in the last decades [6]. Path planning algorithms aim to find the shortest path between two separated points [7]. Two different categories can be identified: local and global path planning. Whilst local planning aims to observe the surrounding environment to identify obstacles and dynamically optimize the path, global planning finds the optimal path by assuming complete knowledge of the environment. In this research, we assume knowledge of the complete environment and hence, we focus the research in global path planning algorithms.

This work was supported by the Engineering and Physical Sciences Research Council under Grant EP/V026763/1.

Ciaran Gruffeille, Adolfo Perrusquía, Antonios Tsourdos and Weisi Guo are with the School of Aerospace, Transport and Manufacturing, Cranfield University, MK43 0AL Bedford, UK.
adolfo.perrusquia-guzman@cranfield.ac.uk.

Dijkstra algorithm has been applied in mobile robot 2D planning and transportation applications [8]. Here, the environment is transformed into a graph where two nodes are modelled as the start and end points. Then the graph is explored by selecting the nodes that contain the edges with shortest distance until the end point is reached [9]. This algorithm requires to visit almost the complete graph to define the optimal path, which can be computationally expensive for large graphs. Specifically, this defines a problem for disaster area scenarios where the points of interest can be large.

Alternatively the A^* algorithm has been used to combine the merits of Dijkstra algorithm with heuristic search algorithms to obtain near optimal paths by dividing the area into units with regular shapes that contain the starting and target point [10]. This method has proven to be more efficient compared the Dijkstra algorithm since it does not require to explore every node of the graph. Additionally, in search and rescue scenarios we do not know the location of the target point, which makes global path planning algorithms to be not adequate for these tasks without additional algorithms.

Deep learning approaches has been used to detect points of interest in imagery data with impressive results [11], [12]. Residual network (ResNet) [13] and its variants has been used as a backbone model for several image segmentation architectures [14], [15]. Here, ResNet uses residual connections to avoid the vanishing gradient problem of deep models, whilst allowing the network to learn efficiently deep patterns for better segmentation. Other model widely used for image segmentation is U-Net [16]. This network works with an encoder-decoder structure that permits to identify points of interest with high accuracy [17].

Adaptive dynamic programming (ADP) [18], [19] has been recently used to optimise the area of coverage in [20] using a multi-agent framework. This is done by modelling the environment as a Voronoi diagram where each agent focus on a specific cell independently [21]. Then the coverage problem is converted into a linear regulator control problem [22], [23] using a policy iteration or actor-critic algorithms [24]–[26]. However, the approach focus on a low-level decision-making control whose targets are given by the Voronoi cells which make computationally expensive for a single agent problem under multiple targets. Reinforcement Learning (RL) has been used mainly for path planning problems with successful results [27]–[29]. Similarly to the A^* and the Dijkstra algorithm, the RL agent requires knowledge of the target point. However, it poses a more simple way to model the environment without focusing on diverse target points and distance approaches that are highly computational expensive [30].

© 2024 IEEE. This is the Author Accepted Manuscript issued with: Creative Commons Attribution License (CC:BY 4.0).
The final published version (version of record) is available online at DOI:10.1109/ICUAS60882.2024.10557095. Please refer to any applicable publisher terms of use.

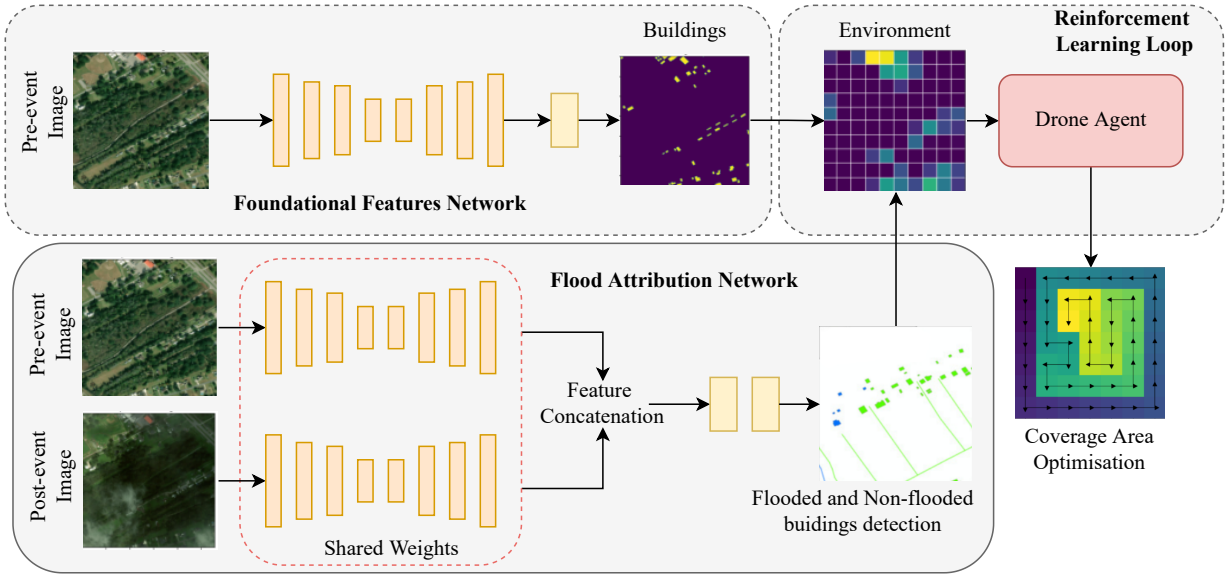


Fig. 1. Proposed Disaster Area Coverage Optimisation high-level diagram

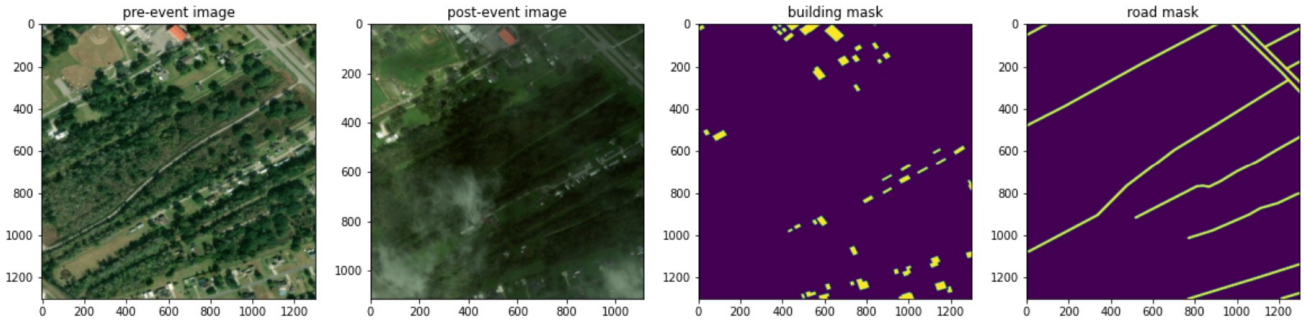


Fig. 2. Louisiana Pre-Disaster Event Masks

In view of the above, the current challenges in disaster area coverage are: i) they are focused on path planning problems with known target point, ii) distance approaches are computationally expensive, and iii) the optimisation of the coverage area is usually modelled by a multi-agent problem where each agent covers a specific area and they are independent to each other, such that they do not share information between them.

In this paper and to alleviate the aforementioned gaps, a coverage area optimisation framework for disaster-event scenarios is proposed. The framework focus on flood satellite imagery data obtained from SpaceNet repository. A image segmentation algorithm is used to determine the points of interest and construct a suitable grid environment. This environment is subsequently used by any reinforcement learning algorithm to optimise the coverage area using a sparse reward function. Here, the requirement of using multiple agents is avoided, whilst reducing the computation time by not being subject to distance approaches. Simulation experiments are conducted under different reinforcement learning agents and grid size. The results show that the proposed algorithm can effectively maximize the area of coverage, whilst minimizing the exploration time.

II. Methodology

The proposed methodology is depicted on Fig. 1. The diagram is composed of three main stages: i) a foundational feature network for building detection using pre-disaster event images. ii) a Siamese-network that uses as inputs the pre-event and post-event images to identify changes in the image caused by the disaster event, and iii) a reinforcement learning loop where an environment is created from i) and ii) to train a reinforcement learning agent (drone) to maximise the coverage area using a sparse reward design. The details of this methodology are given in the subsequent sections.

TABLE I
Data statistics of the SpaceNet dataset

Statistic	Louisiana	Germany
Size (km^2)	741.1	65.5
Buildings	18,892	3,723
Flooded Buildings	2,736	1,175
Flooded Buildings (%)	14	32

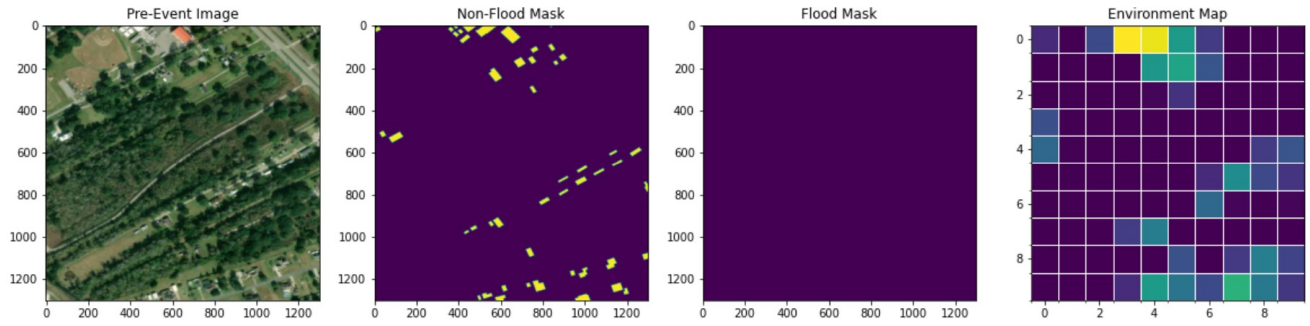


Fig. 3. Environment model from Satellite Images

A. Dataset

In this paper, the SpaceNet dataset from Amazon Web Service is used. The dataset contains approximately 67,000 square km of high-resolution satellite images captured from the hurricane Ida in Louisiana in 2021, and the floods in Dernau, Germany in 2021. Pre-disaster event and post-disaster event images are provided. Additionally, the dataset contains hand-crafted annotations that enable the creation of masks for buildings and roads (see Fig. 2). Table I summarizes some statistics of the SpaceNet dataset.

B. Environment Creation

An improved version of SpaceNet8 (SN8) proposed by the Korea Aerospace Research Institute (KARI) is used as framework for the detection of flood. The algorithm is composed on two main parts: 1) a Foundation Features Network with a U-Net and ResNet34 encoder to extract the building locations from the satellite images, and 2) a Siamese convolutional neural network to detect changes in the imagery data. In this context, one network of the Siamese network takes as input the pre-disaster image, whilst the second network takes as input the post-disaster image. Then, these networks are attached together for comparison of their outputs. This model has an overall accuracy of 66.99% with an inference time of 5 minutes [31].

The output of the SN8 model is used to generate a suitable environment for any reinforcement learning algorithm. This output consists of a square image of 1300×1300 pixels with five channels containing the flooded buildings, the flooded roads, the non-flooded buildings, and the non-flooded roads. We use the flooded and non-flooded buildings channels as main masks since they contain the required information of how the environment changes after a disaster event scenario. We create a grid of $n \times n$ cells that covers the complete size of the image. Here, each cell contains information of the satellite image. The size of the grid will determine how accurate will be the grid model. From the selected masks, we add 1 point for each non-flooded pixel and 2 points for every flooded pixel in a specific cell. Fig. 3 exhibits a 10×10 grid example using a pre-disaster event satellite image.

A two channel environment is proposed. On the one hand, the first channel contains the coverage grid highlighting the previous visited cells by the drone. On the other hand, the second channel contains the reward grid given by the reward values of each cell. The state of the environment is a 3 channel image. The first channel contains the current position of the drone in the grid. The second channel contains the coverage grid that determines which cells have been visited before. The second channel is the one that we aim to optimise by maximizing the coverage area. The third channel contains the one-channel image for visualization purposes.

The reward design consists on a sparse reward function with the following values. In every step of episode, the RL agent loses -0.05 reward value. This penalization is used to encourage the agent to complete the task as fast as possible. For invalid actions (e.g, moving outside the grid), the agent loses -1 reward value. If the applied action is valid and the resulting state is a previously visited state, then a reward penalization of -1 is given. Conversely, if the valid action yields to a new visited state, then the agent is rewarded by 10 times the reward value in that cell and weighted by the inverse of the total number of steps. Here, these reward values encourage the RL agent to maximize the area of coverage in minimum time. If the agent successfully completes the grid, then the agent is rewarded by 200 times the total number of cells minus one and divided by the total number of steps required to complete the task.

C. Reinforcement Learning Models

Three different reinforcement learning models are used in this research: 1) Q-learning, 2) Deep Q-Networks (DQN), and 3) Double Deep Q-Networks (DDQN). The models are assessed by using the average number of steps required to cover the disaster area, and the cumulative reward obtained in each episode. An ϵ-greedy strategy is adopted to trade-off between exploration and exploitation. The models are trained using 1,000 episodes with an exploration rate of 0.3 and discount factor of $\gamma = 0.9$. An early stopping condition is implemented to stop the training when the average steps taken in the last 100 episodes are equal to the optimal solution. For both the DQN and the DDQN uses two convolutional layers with ReLU activation functions followed by a fully connected network of three layers with ReLU activation functions. The output of the network is a dense layer with a softmax activation function.

III. Results

An AMD Ryzen 5 5600x 6-core processor and an Nvidia RTX 3070 are used to train the segmentation and reinforcement learning model.

A. SN8 results

We first train the SN8 model for image segmentation of the flooded buildings. The accuracy of the model is assessed using the Intersection over Union (IoU). The training time was approximately 8-12 hours. Here, the results of the IoU score is approximately 66.3%. Notice that the obtained results give a relative low score due to the complexity of the satellite images, which demonstrate a weakness of the proposed approach for disaster-events applications. Fig. 4 shows the comparison between the ground truth image and the SN8 model.

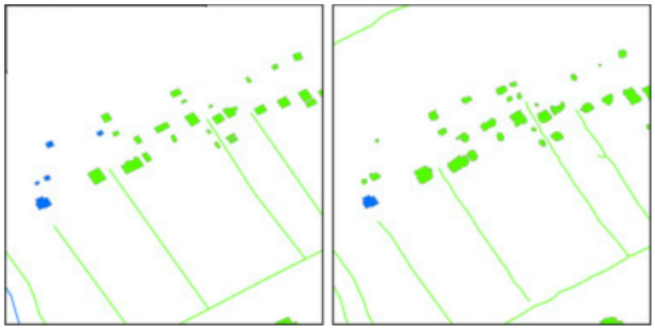


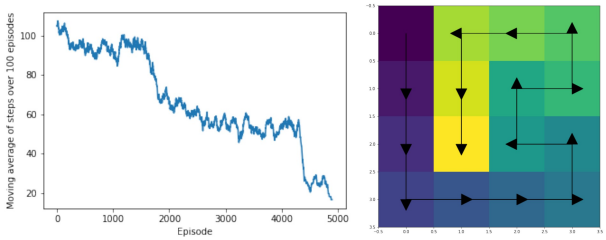
Fig. 4. SN8 results comparisons

The SN8 requires further improvement in order to be used in disaster scenarios. Therefore, in these experiments we use a hand-crafted approach to identify the flooded buildings and create a high accurate environment for the reinforcement learning training.

We create different grids to train the RL agents to optimize the coverage area of a given satellite image sample.

B. Q-learning results

We first create a 4×4 grid for simplicity to train the Q-learning agent. The obtained results are given in Fig. 5.



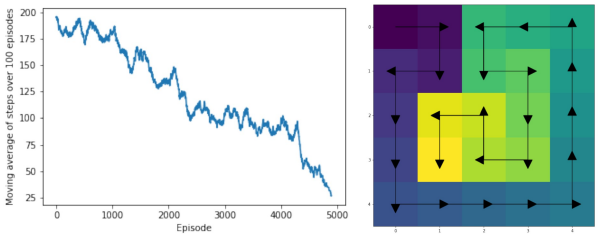
(a) Moving average in a 100 episode window (b) Optimal path founded by Q-learning

Fig. 5. Results of Q-learning under a 4×4 grid.

The moving average result shows a slow convergence to the optimal number of steps required for a 4×4 grid. Here, we use a small decay rate that weights more environment exploration rather than exploitation. The used decay rate is given by

$$\max\left(0.01, \min\left(1, \left(1.1 - \frac{\text{episode}}{\#\text{ of episodes}}\right)\right)\right).$$

The basic idea of the decay rate consists in decaying the exploration probability from 1 to 0.01 as the number of episodes increases. In other words, the current decay factor explores 90% of the total of episodes, whilst only 10% is used to exploit the current knowledge. Here, the final outcome shows that the drone is capable to explore the whole environment effectively without visiting again previous states or applying invalid actions. For this case, the flooded buildings where located in the south-eastern region of the grid. This explains the path followed by the drone by prioritizing flooded regions first instead of non-flooded regions. The size of the grid is increased to a 5×5 grid in order to observe the performance of Q-learning. The final results are observed in Fig. 6.



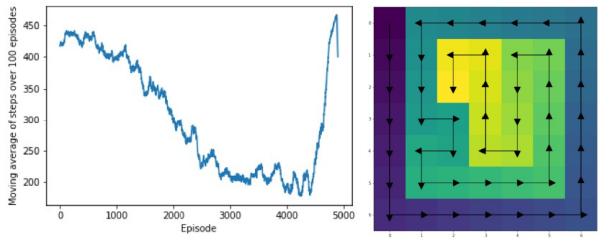
(a) Moving average in a 100 episode window (b) Optimal path founded by Q-learning

Fig. 6. Results of Q-learning under a 5×5 grid

A similar behaviour is observed for this case, where the RL agents obtains the optimal number of steps after 5,000 episodes. Since the grid has been increased, then the path followed of the drone is slightly changed, but it is consistent with the 4×4 grid results.

The proposed approach faces challenges for larger grids since many optimal solutions can be obtained with the same average reward values. For example, for a 7×7 the number of potential solutions are 1,510,446 without visiting previous areas, and for a 10×10 grid the potential solutions grow up o 129 trillion. However, we observe that the proper hyperparameters tuning can yield similar behaviours to the observed on the 4×4 and 5×5 grid. Fig. 7 shows the results for a 7×7 grid.

Notice that the moving average results exhibits a similar behaviour to the previous grids in the first episodes. However, the moving average starts to increase after 4,000 episodes. In this case, the decay rate is small such that if the agent chooses a different path that has not visited before, then the agent cannot learn new things since it will only exploit its current knowledge. In this scenario, the obtained path covers the complete area but it is not optimal since it does not prioritise the flooded buildings over the non-flooded buildings. One important issue of Q-learning is that the agent is trained by starting from the same initial point. The



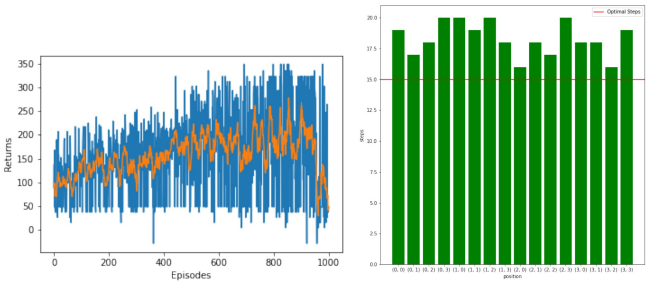
(a) Moving average in a 100 episode window (b) Optimal path founded by Q-learning

Fig. 7. Results of Q-learning under a 7×7 grid

generalization to other cells require to train once again the Q-learning algorithm, which can be regarded as a drawback of Q-learning.

C. Deep Q-Network (DQN) results

We further test the proposed approach using a DQN network. The results are observed in Fig. 8.



(a) Moving average (b) Number of steps required to cover the area

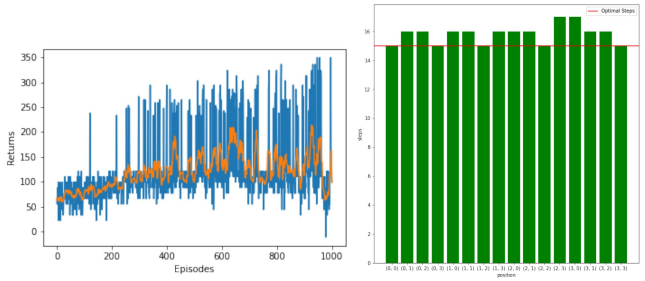
Fig. 8. Results of DQN under a 4×4 grid

The main advantage of DQN is that each episode starts from a random location on the grid, and hence the DQN learns the optimal path from 15 different positions. The moving average result show that the curve drops in approximately 900 episodes, which can be regarded to a lack of exploration given an initial random position. The number of steps obtained by the DQN in random starting points has a mean value of approximately 18.3 steps, which is an acceptable result given the nature of the problem. This also can be attributed to the actions overestimation problem of DQN methods [32].

However, for larger grids the DQN cannot find a suitable solution under the same hyperparameters and decaying rate. Hence, hyperparameter optimisation is required to obtain good results for larger grids.

D. Double Deep Q-Network (DDQN) results

A DDQN model is implemented to deal with the actions overestimation problem of DQN which can provide highest benefit in search and rescue scenarios. The moving average result for a 4×4 grid is shown in Fig. 9. Notice that in contrast to the DQN results, DDQN obtains more stable results by considering the actions that maximize the coverage area and the reward grid. Additionally, the mean number of steps required to cover the complete area starting from random initial cells is approximately 15.81 steps, which clearly improves the DQN results. Fig.10 shows the coverage results of the 4×4 grid starting from a random initial cell.



(a) Moving average (b) Number of steps required to cover the area

Fig. 9. Results of DDQN under a 4×4 grid

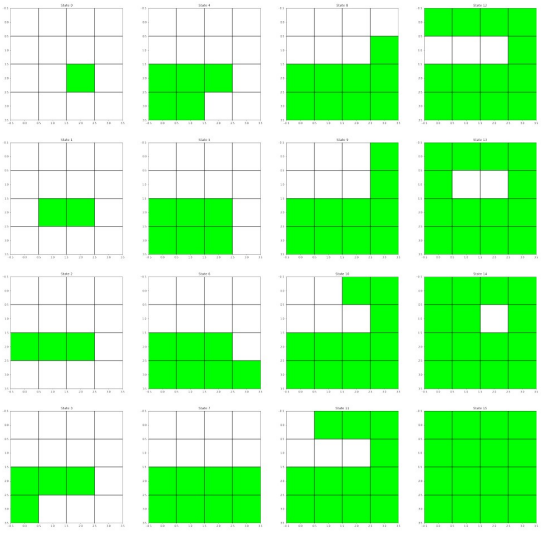
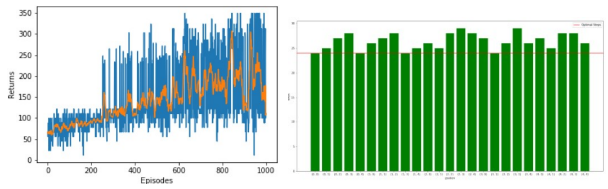


Fig. 10. Optimal path of a 4×4 grid. The path is read from top to bottom and by columns from left to right.

We further test DDQN using a 5×5 grid. The results are observed in Fig. 11. In contrast to the DQN method, the DDQN is capable to handle the 5×5 grid problem by exhibiting a similar performance compared to the 4×4 grid. In this case, the moving average of the rewards exhibit an increasing behaviour as the number of episodes increases. Here, the mean steps value to cover the complete area is approximately 26.4 steps, which is an acceptable value for this environment with higher complexity.

Fig. 12 shows the path followed by the drone starting from an initial random cell. The results clearly show that the area if fully covered by the drone in the minimal number of steps. Here, the DDQN finds the optimal path in some specific starting points. For other points, the path is not optimal since it does not prioritize cells with flooded buildings. This issue can be solved by improving the hyperparameters of the network or by improving the Q-network design.



(a) Moving average (b) Number of steps required to cover the area

Fig. 11. Results of DDQN under a 5×5 grid

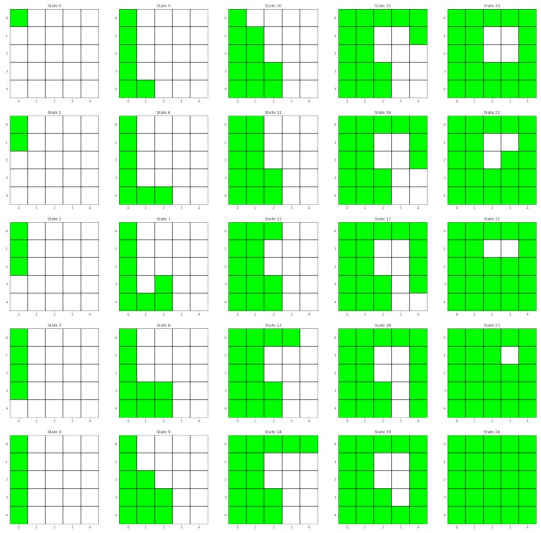


Fig. 12. Optimal path of a 5×5 grid. The path is read from top to bottom and by columns from left to right

For larger grids, the DDQN faces similar challenges to Q-learning, i.e., the number of optimal paths is increased exponentially. This issue can be solved by implementing a multi-agent reinforcement learning [33]–[35] framework that divides the area in different zones, such that each agent can adopt the proposed reinforcement learning approach.

IV. Conclusions

This paper provides a framework to optimise the coverage area of disaster scenarios using reinforcement learning. The approach uses the satellite images of the SpaceNet dataset. The SpaceNet8 framework is used to identify flooded buildings and non-flooded buildings. The output of SpaceNet8 is used to create a suitable environment for reinforcement learning training. The environment is constructed by a user-design grid. The coverage area is maximized by considering the visited states and the reward obtained from the complete cycle of episodes. The results show that the SN8 is not a suitable model to identify the flooded buildings accurately - required for search and rescue applications. This is topic for further work. To alleviate this issue, a hand-crafted approach is used to identify manually the flooded buildings for the accurate creation of the RL environment. Three RL models are used to show the benefits and challenges of the approach in the coverage area optimisation. Future work will consider multi-agent reinforcement learning to distribute better the exploration of the complete environment in minimum time.

References

[1] N. Wang, ""a success story that can be sold"?: A case study of humanitarian use of drones," in *2019 IEEE International Symposium on Technology and Society (ISTAS)*. IEEE, 2019, pp. 1–6.

[2] M. El Debeiki, S. Al-Rubaye, A. Perrusquía, C. Conrad, and J. A. Flores-Campos, "An advanced path planning and uav relay system: Enhancing connectivity in rural environments," *Future Internet*, vol. 16, no. 3, p. 89, 2024.

[3] G. Singh, A. Perrusquía, and W. Guo, "A two-stages unsupervised/supervised statistical learning approach for drone behaviour prediction," in *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2023, pp. 1–6.

[4] B. Fraser, A. Perrusquía, D. Panagiotakopoulos, and W. Guo, "Hybrid deep neural networks for drone high level intent classification using non-cooperative radar data," in *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. IEEE, 2023, pp. 1–6.

[5] A. Perrusquía, W. Guo, B. Fraser, and Z. Wei, "Uncovering drone intentions using control physics informed machine learning," *Communications Engineering*, vol. 3, no. 1, p. 36, 2024.

[6] Y. Yang, X. Xiong, and Y. Yan, "UAV formation trajectory planning algorithms: A review," *Drones*, vol. 7, no. 1, p. 62, 2023.

[7] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, pp. 3–27, 2015.

[8] M. Luo, X. Hou, and J. Yang, "Surface optimal path planning using an extended dijkstra algorithm," *Ieee Access*, vol. 8, pp. 147 827–147 838, 2020.

[9] H. I. Kang, B. Lee, and K. Kim, "Path planning algorithm using the particle swarm optimization and the improved dijkstra algorithm," in *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 2. IEEE, 2008, pp. 1002–1004.

[10] G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou, "Geometric a-star algorithm: An improved a-star algorithm for agv path planning in a port environment," *IEEE access*, vol. 9, pp. 59 196–59 210, 2021.

[11] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3523–3542, 2021.

[12] B. Fraser, A. Perrusquía, D. Panagiotakopoulos, and W. Guo, "A deep mixture of experts network for drone trajectory intent classification and prediction using non-cooperative radar data," in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2023, pp. 1–6.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] T. Kacker, A. Perrusquia, and W. Guo, "Multi-spectral fusion using generative adversarial networks for uav detection of wild fires," in *2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2023, pp. 182–187.

[15] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern Recognition*, vol. 90, pp. 119–133, 2019.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[17] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," *Ieee Access*, vol. 9, pp. 82 031–82 057, 2021.

[18] J. A. Flores-Campos and A. Perrusquía, "Robust control of linear systems: A min-max reinforcement learning formulation," in *2023 20th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*. IEEE, 2023, pp. 1–6.

[19] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 76–105, 2012.

[20] A. A. Adepegba, "Multi-agent area coverage control using reinforcement learning techniques," Ph.D. dissertation, Université d'Ottawa/University of Ottawa, 2016.
[21] A. Perrusquía and W. Guo, "Hippocampus experience inference for safety critical control of unknown multi-agent linear systems," *ISA transactions*, vol. 137, pp. 646–655, 2023.
[22] K. G. Vamvoudakis and F. L. Lewis, "Online solution of nonlinear two-player zero-sum games using synchronous policy iteration," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 13, pp. 1460–1483, 2012.
[23] A. Perrusquía, "Solution of the linear quadratic regulator problem of black box linear systems using reinforcement learning," *Information Sciences*, vol. 595, pp. 364–377, 2022.
[24] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2042–2062, 2017.
[25] A. Perrusquía and W. Guo, "Drone's objective inference using policy error inverse reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
[26] K. G. Vamvoudakis, N.-M. T. Kokolakis *et al.*, "Synchronous reinforcement learning-based control for cognitive autonomy," *Foundations and Trends® in Systems and Control*, vol. 8, no. 1–2, pp. 1–175, 2020.
[27] A. I. Panov, K. S. Yakovlev, and R. Suvorov, "Grid path planning with deep reinforcement learning: Preliminary results," *Procedia computer science*, vol. 123, pp. 347–353, 2018.
[28] H. Bae, G. Kim, J. Kim, D. Qian, and S. Lee, "Multi-robot path planning method using reinforcement learning," *Applied sciences*, vol. 9, no. 15, p. 3057, 2019.
[29] X. Lei, Z. Zhang, and P. Dong, "Dynamic path planning of unknown environment based on deep reinforcement learning," *Journal of Robotics*, vol. 2018, 2018.
[30] A. Perrusquia and W. Yu, "Robust control under worst-case uncertainty for unknown nonlinear systems using modified reinforcement learning," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 7, pp. 2920–2936, 2020.
[31] R. Hänsch, J. Arndt, D. Lunga, M. Gibb, T. Pedelose, A. Boedihardjo, D. Petrie, and T. M. Bacastow, "Spacenet 8-the detection of flooded roads and buildings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1472–1480.
[32] W. Yu and A. Perrusquia, *Human-Robot Interaction Control Using Reinforcement Learning*. John Wiley & Sons, 2021.
[33] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent reinforcement learning: An overview," *Innovations in multi-agent systems and applications-1*, pp. 183–221, 2010.
[34] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of reinforcement learning and control*, pp. 321–384, 2021.
[35] A. Perrusquía, W. Yu, and X. Li, "Multi-agent reinforcement learning for redundant robot control in task-space," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 231–241, 2021.



https://dspace.org/

Disaster area coverage optimisation using reinforcement learning

Gruffeille, Ciaran; Perrusquía, Adolfo; Tsourdos, Antonios; Guo, Weisi

2024-06-04
Institute of Electrical and Electronics Engineers (IEEE)
https://doi.org/10.1109/ICUAS60882.2024.10557095
Gruffeille C, Perrusquía A, Tsourdos A, Guo W. (2024) Disaster area coverage optimisation using reinforcement learning. In: 2024 International Conference on Unmanned Aircraft Systems (ICUAS), 4-7 June 2024, Crete, Greece, pp. 61-67

Downloaded from a DSpace repository. https://dspace.org/