Deep Reinforcement Learning based Path Planning for Multi UAV-assisted Edge Computing Networks

Shalini Lodhi
Dept. of Electronics & Communication Engineering
IIIT Naya Raipur
shalini19101@iiitnr.edu.in

Shreya Masta
Dept. of Electronics & Communication Engineering
IIIT Naya Raipur
shreya19101@iiitnr.edu.in

***Abstract*—Aerial drones (UAVs) have long been utilized in mobile networks as network processors, but they are now being employed as mobile servers in Mobile Edge Computing (MEC). Because of their flexibility, portability, strong line-of-sight communication linkages and low-cost, changeable use, they have become more popular in research and commercial applications. A wide range of civilian services may now be supported by their essential characteristics, including transportation and industrial monitoring and agricultural, as well as forest fire and wireless services. Mobile edge computing networks based on Unmanned Aerial Vehicles are researched in this project, where the unmanned aerial vehicle (UAV) does computations that mobile terminal users supply to it. (TUs). In order to assure each TU's Quality-of-service (QoS), the UAV dynamically selects its course based on mobile TUs' locations.**

***Index Terms*—Aerial Drones, Mobile Edge Computing, Quality-of-service.**

I. INTRODUCTION

The use of mobile data processing technologies in the communications industry is on the rise. IoT devices with large computing capacity may be launched with unique applications and services in a flexible and timely way due to this technology. When edge servers are used to offload computation-intensive tasks, latency is reduced and energy consumption is reduced.

Recent years have seen the utilization of unmanned aerial vehicles (UAVs) in the form of multi-access edge computing servers for end users. As a result of its flexible deployment, comprehensive control, and network performance, UAV-assisted wireless communication has received substantial study interest.

The UAV-assisted edge computing network makes sense and is an intriguing concept when it comes to dealing with the communication and processing demands of enormous devices.

By investing in RD into Unmanned Aerial Vehicles (UAVs), companies can ensure that their mobile edge computing services are reliable and secure. Because of their small size, they may be used in a variety of ways and at a low cost. The upshot is that UAVs are able to fly freely between terminal users and deliver exceptionally efficient calculation services to improve QoS. The QoS of each terminal user is what determines the effectiveness of mobile edge computing. Demands of terminal users may be addressed or served more quickly when QoS is higher.

It's not uncommon for researchers to look at planning as an all-or-nothing endeavor that doesn't have an ongoing improvement process in mind. When the UAVs cannot react to changing environmental variables, the planning might fall into a local optimum. UAV-mounted mobile edge computing research has rarely focused on risk avoidance or collisions between UAVs, owing to the fact that such scenarios are impractical to simulate in real-world scenarios.

For many issues, Reinforcement Learning (RL) looks to be a useful tool. In comparison to the algorithms of A8 and RRT, the RL method is more versatile because of the following factors:

When it comes to edge computing, residual demand from terminal users varies often, necessitating real-time policy updates to keep up. Such time-varying events, on the other hand, cannot be handled effectively by traditional approaches.

When dealing with this problem, algorithms that simply examine geometrical restrictions have a tough time since the map contains both barriers and terminal users.

It is possible to change the policy by modifying the cost function in RL, allowing for flexible policy modification.

This means that the mobile edge computing platform may be built on RL, which is more flexible to a wide range of scenarios.

II. PROJECT PLANNING

This is how the project will be broken down:

Route planning and mobile edge computing are connected together through RL such that cost matrices for UAVs are merged to account for geometric distance, risk, and terminal user demand in a single cost matrix for the drones.

Second, we examine multi-UAV cooperation in the context of mobile edge computing. UAVs may save money and avoid obstacles thanks to the sharing of geometric and terminal user data.

There are a number of ways in which we might improve our quality of service (QoS). The sigmoid-like demand function is more efficient in allocating jobs than the standard linear demand function.

As a final step, we carry out a wide range of tests on the proposed platform and analyse different cost functions. The outcomes have shown the usefulness and viability of our strategy

III. PATH PLANNING FOR UAV- MOUNTED MULTI-UAV MOBILE EDGE COMPUTING WITH REINFORCEMENT LEARNING

In the project there is an agent known as a mobile network processor (UAV). When a UAV is assigned a time slot, it determines the best technique to use in order to maximise reward depending on the conditions in which it is operating. In the course of travelling, a UAV's surroundings change, and a reward matrix A that takes into consideration risk and the geometric distance between the UAV and the terminal user is sent to the UAV. The UAV then use a cost matrix G obtained from A to develop a strategy based on its knowledge of the surrounding environment and the information it has gathered so far. As the planning process progresses, each agent's "memory" becomes stronger and more "trained.".

A. Environmental Modelling

This study analyses the UAV's collision avoidance as well as the need for terminal users on the same platform. The environment is made up of two main elements: barriers and terminal users. In a real-world context, obstacles vary in shape, position, and danger level, and might include buildings, automobiles, or mountains.

We also assume that the distribution of the barriers is Gaussian with different variances for determining their risk exposure probability.

If there are n obstacles independently on the map, the risk ri at the point (x,y) refers to the risk from obstacle Oi at the place (xi, yi) and may be expressed as

$$
\begin{gathered}
r_i(x,y) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{d^2}{2\sigma}}, \\
d = \sqrt{(x - X_i)^2 + (y - Y_i)^2}, \\
i \subseteq (1, 2, ..., n)\,.
\end{gathered} \tag{1}
$$

With n barrier on the map considered, risk to a location (x, y) in the risk exposure probability matrix is defined as:

$$
R\,(x, y) = \prod_{i=1}^{n} [1 - r_i(x, y)] \tag{2}
$$

Exposed risk between two points, p and q is the integration of the risk R at (x, y) for every (x, y) linearly connecting p to q:

$$
\int_{(x,y)C} R\,(x, y) \tag{3}
$$

Second, To begin with, we will assume that the initial demand of each terminal user is d0j that the UAVs must meet while simultaneously supplying terminal users. A second assumption is that unmanned aerial vehicles (UAVs) can only provide demand within a particular radius of servce because unmanned aerial vehicles (UAVs) have limited ability to detect demand signals beyond a set distance. As soon as the UAV reaches the service area of TUj, TUj's services are launched. For the foreseeable future, the remaining demand for TUj will continue to decline at a steady rate. An increased demand for services necessitates more time, and a UAV's time in TUj's area of service increases its ability to deliver service. Distance between two spots on the ground defines TUj's service area. When the amount of time spent by UAVk increases, so does dj. as

$$
d_j^{l+1} = d_j^l - \pi t_k \tag{4}
$$

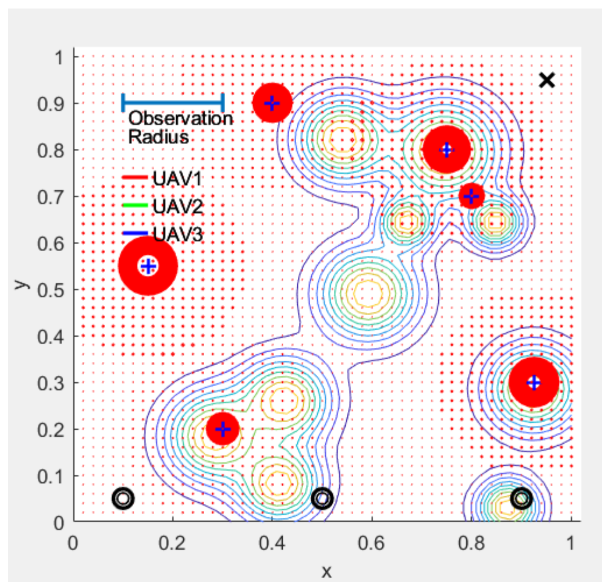


Fig. 1. Background path

B. The Reward Matrix

A reward matrix is offered to assist UAVs in learning and adapting so that they may pick the optimal route for them. Using the reward matrix, it is possible to compute the reward or penalty from any point on the map to any other point on the map, while taking into consideration factors like as terminal user demand, risk and geometric distance.

During the project's development, an NxN lattice represents the map, and the payout Api,pr between each point pi and pr on the map is stated as follows:

$$A_{p_t,p_r} = d_{p_t,p_r} + K \int_C R(x,y)\, ds + \frac{M}{1 + \sum_{j \subseteq s(\pi,e)} U(d_j)} \tag{5}$$

The parts of equation can be explained as:

- The distance between pi and pr.
- The detected risk from pi to pr, which indicates that the higher the detected risk, the greater the penalty.
- The total demand identified at pi, where pi is the UAV's current position. The more the identified demand, the less severe the penalty or the greater the reward.

When a point pr, is located on the map, a reward matrix Apr is formed and is located on the map with relation to the map at all points pi. K and M represent the risk tolerance and service priority, respectively, and have an impact on the approach for route planning. When implemented in real-world scenarios, K and M may be changed to meet the specific needs of the operation.

For example, A relatively high value of K drives the UAVs away from the obstacles, despite the fact that this results in a longer route length for the drones.

In order to better match real-world scenarios, an observation radius for the obstacle is provided for each UAV. An unmanned aerial vehicle (UAV) identifies an obstacle when it enters the monitored area of the UAV and gathers information about the threat.

When constructing the weight matrix, only the impediments that have been noticed will be considered a risk. The observed risk if a UAV is at point P and has an observational radius of R is shown by the color highlighted in the figure.

Below is the reward matrix based on the observed risk distribution of each UAV and the demand of the entire terminal user population. The demand of the terminal user lowers the cost, but the detected danger raises the cost of the terminal. And thus we may conclude that the dark colors represents lower values of terminal user demand and the lighter areas represent higher values of it.

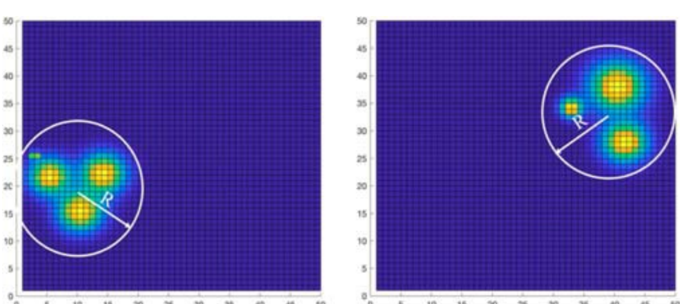


Fig. 2. (a) P = (0.15,0.4) (b) P = (0.75, 0.6) *Observed risk distribution*

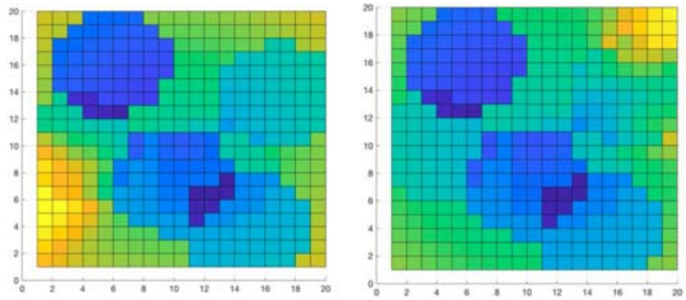


Fig. 3. (a) P = (0.15,0.4) (b) P = (0.75, 0.6) *Weight matrix*

C. The Cost Matrix

To achieve a preliminary best route to the destination, a cost matrix G is added during the path design process for an unmanned aerial vehicle (UAV). An iterative approach is used to generate the cost matrix, which is then evaluated.

When the cost matrix has been iterated multiple times, it will converge and become stable. When there are N computing nodes in a map, the updating method for the cost matrix is defined as follows:

- *Initialize G*: This step involves initialising the cost matrix G0. The target point should have the value 0, while all other points should have the value 1..
- *Make changes to the cost matrix G*: Pick a random location on the map using the arrow keys. Create an update to the point value in G for each PI point on the map by comparing, with consideration for the reward matrix, the current point value with that of the updated one.
- Repeat the above step till maximum number of iterations have completed.

Following the generation of G, an ordinal series of locations is established, which serves as a tentative route for the UAV to follow.

InGare, new points with the lowest prices are being added on a daily basis. The following is a description of the path creation process:

- Create an empty list as the starting point for Path.
- To Path, adding pi lowest in value of G, then assign Gpi to Path.
- Repeat steps 2 and 3 until you reach your destination or reach the maximum length of your rope.

It's worth noting that the products in Path are sorted by price, with the most expensive at the top. The Algorithm combines the calculation of G with the generation of Path to provide the Planning function.

D. Algorithm of UAVs Movement

Algorithm 1: UAV Movement Algorithm.

```
for i in UAVnum do
  Initialize G(i)
  Path_i ← Planning()
end for
for i in UAVnum do
  if pos_i = TargetPoint then
    Stopmovement(i)
  else
    // Remove outdated information from D_i because
    pos_j has changed in last loop
    for j in UAVnum and j! = i do
      delete pos_j from memory D_i
    end for
    ObstacleFound ← ScanEnv(pos_i, R)
    if ObstacleFound then
      Path_i ← Planning()
    end if
    if pos_i = Path_i[1] then
      Path_i ← Path_i[2...end]
    end if
    pos_i ← Move(StepLength, Path_i[1])
    for TU_j within s(pos_i, ε) do
      d_j ← d_j − τ
    end for
  end if
end for
```

IV. Simulation

A. Multiple UAVs Path Planning

In this simulation, we set

K	M	n	B	R
20	1	2	8	0.2

First, we allocate ten obstacles at random places varying randomly, and six terminal users are placed at random requests. In a real-world setting, the demand of the terminal user may be a variable that changes in real time.

The position marked with a black cross serves as the target point for all unmanned aerial vehicles. All unmanned aerial vehicles (UAVs) are charged with serving the terminal users on a map and flying to the designated target location for each operation.

The presence and quantity of terminal users' demand are shown by the red dots, which are located within a service radius. While terminal customers are being supplied, the red spots drop in size, indicating a decrease in the amount of demand that remains. Observe that there is a significant overlap in the service radius, indicating that the demands are cumulative.

Our framework allows us to infer that unmanned aerial vehicles (UAVs) may find a low-risk route to service each terminal user in a complicated environment. UAVs are more appealing to terminal users who have a larger demand for their services.

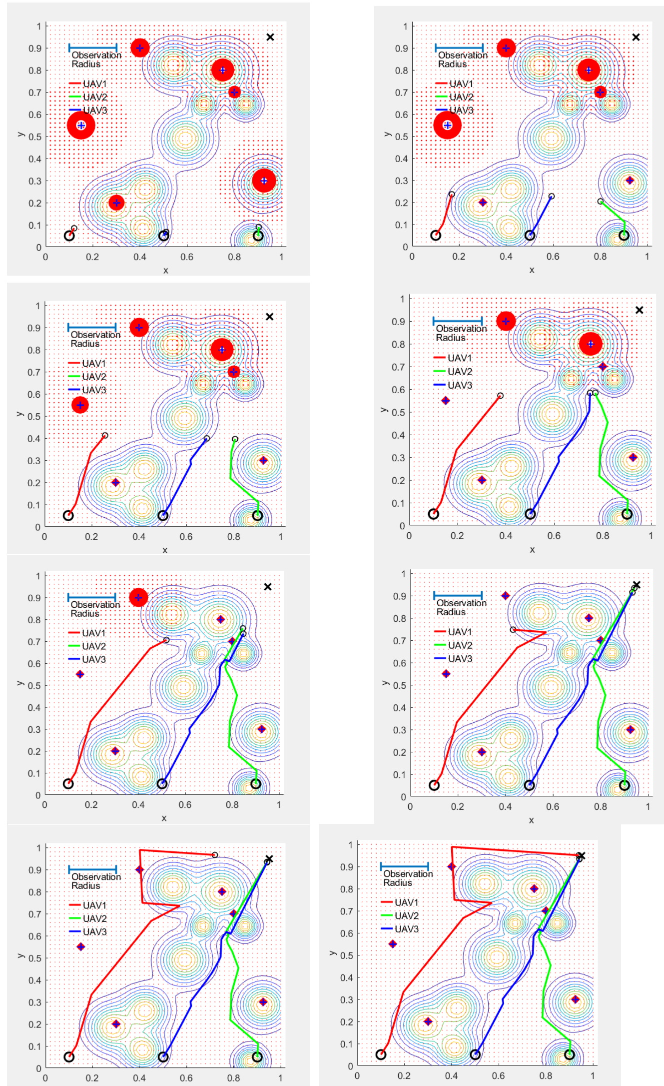


Fig. 4. Steps taken by UAVs

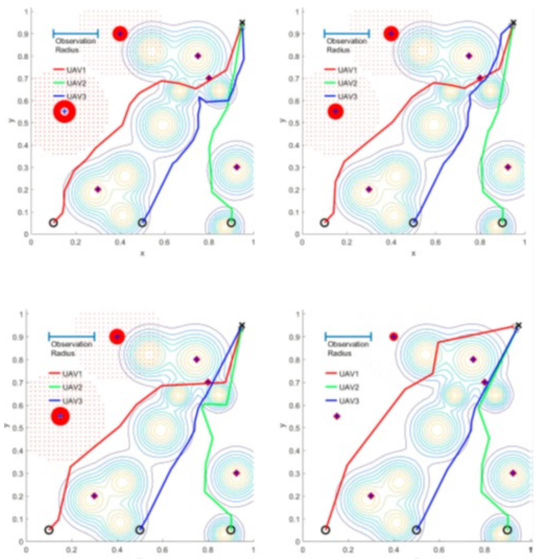
After the demand for one location has been lowered, UAVs will shift their course to other places with strong demand. Meanwhile, information exchange is beneficial in preventing accidents between unmanned aerial vehicles (UAVs) during the planning phase.

B. The Evaluation of M

M specifies the order in which terminal users are prioritised for service. and, as a result, governs the quality of service (QoS). Given a given K, in situations with bigger M, UAVs are more likely to satisfy more terminal user needs, but they do so at the expense of greater risk and reduced travel length (i.e., energy consumption).

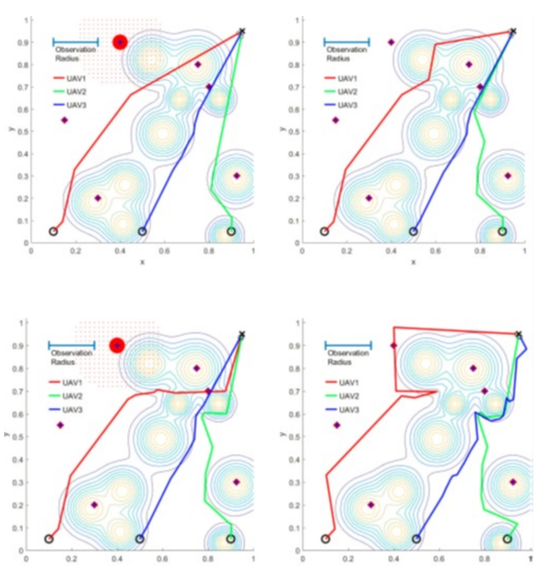
The opposite is true: Under conditions with a lower M, UAVs are unable to serve all terminal users, but they consume less energy because they fly less.

We can satisfy the criteria of various missions with varying service requirements by altering the value of M. Quality of service and risk may be managed in a flexible manner.



C. The Evaluation of K

Changing the parameter K, in a similar way as changing the parameter M, may make the algorithm more adaptable to varied settings. K is in charge of determining one's risk tolerance. UAVs with a higher K in the outcome have a tendency to sacrifice energy cost in order to reduce danger, which has an impact on QoS.



CONCLUSION

We built our project such that there must be a quality of service that is tailored to the individual needs of each terminal user, as well as maximum collision avoidance with the smallest amount of risk, and cooperation amongst unmanned aerial vehicles. As a starting point for future mobile edge computing applications, these tests and simulations illustrate the platform's efficiency and usability.

REFERENCES

[1] Y Peng, Y Liu, H Zhang, "Deep Reinforcement Learning based Path Planning for UAV-assisted Edge Computing Networks", doi: 10.1109/WCNC49053.2021.9417292

[2] PA Apostolopoulos; G Fragkos; EE Tsiropoulou; S Papavassiliou. "Data Offloading in UAV-assisted Multi-access Edge Computing Systems under Resource Uncertainty", doi: 10.1109/TMC.2021.3069911

[3] Q Liu, L Shi, L Sun, J Li, M Ding, F Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning", doi: 10.1109/TVT.2020.2982508

[4] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," 30th AAAI Conference on Artificial Intelligence, pp. 2094–2100, 2016