Unmanned Aerial Vehicle-Mounted Edge Server Deployment for Internet of Vehicles

Zhihai Tang[1], Aiwen Huang[1], and Le Chang[1]

[1]Guangdong University of Technology

April 08, 2024

Abstract

Edge computing has been proved an efficient approach to provisioning computation offloading service to vehicles on road through Road-Side Units (RSUs). However, the traffic volume on road is highly dynamic, while RSU-based edge servers are static in terms of geographical location and computation capacity. To address this problem, this paper proposes a mobile edge server placement strategy using cruising UAVs along the roads based on the genetic algorithm. We first build a mathematical model to characterize the deployment cost of these UAV-mounted servers and their routes. Next, we design a heuristic UAV-mounted edge server deployment scheme based on K-medoid clustering and genetic algorithms. Experimental results verify that our proposed UAV deployment scheme satisfies the offloading demand of IoV nodes while reducing the total deployment cost by 17.05% to 48.94% compared with existing popular approaches.

Unmanned Aerial Vehicle-Mounted Edge Server Deployment for Internet of Vehicles

Zhihai Tang,[1] AiWen Huang,[1] and Le Chang[1]
[1] *School of Automation, Guangdong University of Technology, Guangzhou, China*
Corresponding address: lechang@gdut.edu.cn

Edge computing has been proved an efficient approach to provisioning computation offloading service to vehicles on road through Road-Side Units (RSUs). However, the traffic volume on road is highly dynamic, while RSU-based edge servers are static in terms of geographical location and computation capacity. To address this problem, this paper proposes a mobile edge server placement strategy using cruising UAVs along the roads based on the genetic algorithm. We first build a mathematical model to characterize the deployment cost of these UAV-mounted servers and their routes. Next, we design a heuristic UAV-mounted edge server deployment scheme based on K-medoid clustering and genetic algorithms. Experimental results verify that our proposed UAV deployment scheme satisfies the offloading demand of IoV nodes while reducing the total deployment cost by 17.05% to 48.94% compared with existing popular approaches.

Introduction: The rapid development of the Vehicle-to-Everything (V2X) technology has become an integral part of modern Internet of Vehicles (IoV) [1–3]. Nevertheless, with the emergence of new applications such as autonomous driving and high-precision real-time map navigation, there has been a surge in on-board computing demands where the in-car computation capacity is insufficient [4, 5].

Edge computing has emerged as one of the most suitable solutions to addressing such challenge. Edge computing provides a decentralized processing architecture by deploying computing resources near the IoV nodes (vehicles), i.e., the data source, which significantly lightens the computation load on IoV nodes. This distributed computing model is particularly suitable for those applications that are extremely sensitive to latency [6].

Despite numerous benefits edge computing brings to IoV systems, the traditional static edge computing model still faces great challenges [7] [8]. Static edge servers, located at fixed locations, inherently fail to adapt to the dynamically changing traffic [9] [10]. To this end, mobile carriers have been proposed, such as using Unmanned Aerial vehicles (UAV) equipped with edge servers to follow the dynamic traffic and offer elastic computation services to the vehicles on road. These carriers can dynamically adjust their locations based on real-time traffic conditions and user demand [11].

Efficient route plan of the routes of UAV-mounted edge servers remains a critical opening problem. Existing work usually focus on route planning of a small number of UAVs and static user demands, while ignoring the collision risk of overlapping flight paths of a large number of UAVs and their limited battery capacity [12] [13]. To address these issues, we propose a deployment scheme for UAV-mounted edge servers based on Genetic Algorithms (GA). We make a set of UAVs carry edge servers and fly along a variety of roads, i.e., routes, to serve the offloading requests on the way. Each route is formed as a queue of UAVs to avoid collision. We also place charge stations for these routes to ensure continuous service of these UAV-mounted servers. We carefully optimize the cruising routes of the UAVs to reduce the overlapping chance of different routes, cover all the offloading requests on each road, and reduce the deployment cost of the UAVs. The major contributions of our work can be summarized as follows.

- We develop a mathematical model of the deployment and route planning of UAV-mounted edge servers for an IoV system, aiming at minimizing the total deployment cost while meeting the offloading requirements of the IoV nodes on road.
- We design an efficient heuristic UAV deployment strategy based on K-medoid clustering and the genetic algorithm. We use the K-medoid algorithm to partition the map into regions and place one charging station at the medoid of each region, and design a GA-based algorithm to determine the routes in each region. Each route must go through one charge station. We iteratively explore different combinations of the number of regions and routes to find the best solution.
- We perform trace-driven experiments on real GPS data of the Shanghai traffic trace. Compared with existing popular algorithms, our proposed strategy can reduce the deployment cost by 17.05% to 48.94% in typical scenarios.

Related Works: The strategic deployment of edge servers to effectively serve the computational needs of IoT nodes poses a complex challenge [14]. Static edge servers struggle with adaptability due to inflexible placement and limited capacity. To address this, researchers explore innovative solutions, such as using mobile edge servers on UAVs [14–16].

In addition to addressing adaptability issues, various algorithms have been proposed to efficiently schedule tasks and minimize communication costs in dynamic computational resource provisioning scenarios. Saurez et al. introduced a task graph partition offloading algorithm that takes into account end device capabilities [17], while Lv et al. applied Q-learning techniques to schedule microservice containers, aiming to reduce resource variance and communication overhead [18].

Resource allocation in Mobile Edge Computing (MEC) systems is another prominent area of research. Xiong et al. contributed to enhancing resource utilization with an improved DQN algorithm [19]. This optimization is crucial for the efficient operation of MEC systems.

Furthermore, researchers have explored the integration of UAV-based mobile edge computing in Intelligent Transportation Systems (ITS), particularly in traffic flow monitoring and management [20]. The incorporation of UAVs into MEC networks has demonstrated improvements in computational performance and reductions in execution latency [21]. The utilization of drones for IoT traffic monitoring further underscores the potential of edge computing in intelligent transportation system applications [22, 23].

Notwithstanding these noteworthy advancements, it is imperative to acknowledge that variations in workload within IoT environments can potentially jeopardize the continuity of communication. In addressing this formidable challenge, our research endeavors are dedicated to the meticulous optimization of UAV-based edge computing, with the overarching goals of enhancing energy efficiency and quality of service. Our specific focus lies in the optimization of UAV service routes and the strategic positioning of charging stations to ameliorate the efficiency and efficacy of IoT node services.

The Deployment Model of UAV-Mounted Edge Servers:

System Model: We consider an IoV system with vehicles on road as *IoV nodes*. These IoV nodes continuously generate offloading requests. The road network is modeled as a graph, $\mathcal{M} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_i | i = 1, \ldots, N\}$ is the set of the vertices representing the road intersections, and $\mathcal{E} = \{e_{i,j} | v_i, v_j \in V; i < j\}$ is the set of the edges representing the road sections. For a given time interval, let $l_{i,j}$ denote the total volume of the computation demand of the offloading requests generated within road section $e_{i,j}$, and $d_{i,j}$ denote its length. We assume K bases need to be placed to some intersections, acting as the manage and charge stations. Let $\mathcal{B} = \{b_i | i = 1, \ldots, K; b_i \in V\}$ represent the set of all bases.

We need to determine a variety of cruising routes of the UAV-mounted edge servers to offer offloading services, ensuring that all the demand of all the road sections is covered with sufficient computation capacity. We also need to place a number of charge stations on the map, and each route must go through a charge station. The problem is to find the locations of these UAV charge stations and plan the cruising routes of the UAVs with minimum deployment cost. Fig. 1 illustrates the model of the problem.

The Deployment Cost: We define the deployment cost as the sum of the cost on deploying charge stations and the cost on the routes of the UAV-mounted edge servers. The deployment cost of the charge stations is simply calculated as

$$C^{\text{all charge stations}} = C^{\text{charge station}} K \tag{1}$$

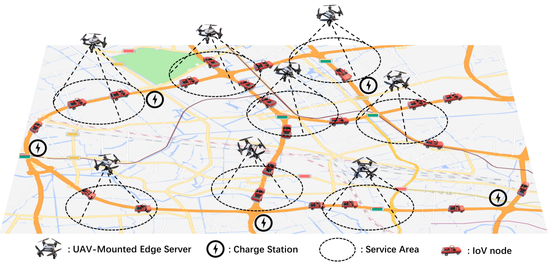

Fig 1 *The IoV system with UAV-mounted edge servers.*

where $C^{\text{charge station}}$ is the deployment cost of deploying a single charge station.

To deploy a route, we consider the number of UAV-mounted edge servers on road with certain density to serve the load of each road section in route. We use $u_i =< e_{x,y} | e_{x,y} \in \mathcal{E} >$, an ordered set of edges, to denote route i. Such deployment cost of route i is calculated as

$$C_i^{\text{route}} = \sum_{e_{x,y} \in u_i} C^{\text{dist}} d_{x,y} n_i^{\text{server}} \tag{2}$$

where C^{dist} is the deployment cost on one unit travelling distance by a single UAV-mounted edge server, and n_i^{server} is the number of UAV-mounted edge servers in route i, calculated as

$$n_i^{\text{server}} = \sum_{e_{x,y} \in u_i} d_{x,y} \epsilon_i^{\text{server}} \tag{3}$$

where $\epsilon_i^{\text{server}}$ is the density of the UAV-mounted edge servers in the route, i.e., number of UAV-mounted edge servers per KM.

To satisfy all the requests along the route, it must follow that the load of any road section should be no greater than the supply by all the routes, i.e.,

$$\forall e_{x,y} \in \mathcal{E} : \frac{l_{x,y}}{d_{x,y}} \leq \sum_{\forall j : e_{x,y} \in u_j} \eta \epsilon_j^{\text{server}} \tag{4}$$

where the computation capacity of a UAV-mounted edge server is η.

Each UAV-mounted edge server has a communication range with diameter D, and the UAV-mounted edge servers must cover all the road sections in length. Therefore,

$$\forall\ e_{x,y} \in u_i : \frac{d_{x,y}}{n_i^{\text{server}}} \leq D. \tag{5}$$

The Formulation of the Optimization Problem: Our goal is to minimize the total deployment cost, so the optimization problem is formulated as

$$\textbf{min} : \quad C = C^{\text{all charge stations}} + \sum_{i=1}^{M} C_i^{\text{route}} \tag{6}$$

$$\textbf{s.t.} \quad \forall e_{(x,y)} \in \mathcal{E} : \frac{l_{x,y}}{d_{x,y}} \leq \sum_{\forall j : e_{(x,y)} \in u_j} \eta \epsilon_j^{\text{server}} \tag{7}$$

$$\forall\ e_{x,y} \in u_i : \frac{d_{x,y}}{n_i^{\text{server}}} \leq D \tag{8}$$

$$\epsilon_j^{\text{server}} \geq 0 \tag{9}$$

$$\forall i : u_i \text{ is a loop passing } v_x \in \mathcal{B} \tag{10}$$

The problem above can be converted to a variant of the Vehicle Routing Problems (VRP), which is NP-hard [24].

Heuristic Algorithms:

Methodology: Given the complexity of Problem 6, we employ heuristic algorithms, combining K-medoid clustering and Genetic Algorithm (GA). Initially, we partition the map into regions, deploying a single charge station in each to break down the problem. Subsequently, a genetic algorithm is utilized for route determination and UAV-mounted edge server density within these regions. Finally, we employ an iterative optimization approach to assess deployment costs for varying numbers of regions/bases (K) and routes (M), identifying the optimal parameter combination to address the charge station deployment and route planning challenge.

Map Partitioning Using K-medoid Clustering: We treat the load of a road section as the number of offloading points uniformly distributed along the section, and use K-medoid clustering to partition these points into clusters, referred to as *regions*. Initially, we select k random points as medoids, assign each point to the nearest medoid's cluster, and update medoids to minimize distances within clusters iteratively. The charge station is placed at the medoid of each region. This map partitioning reduces the scale of Problem 6. We chose K-medoid over K-means for its robustness in handling GPS signal bias and ensuring more reliable results.

Route Planning Using GA: The route planning problem is a complex combinatorial problem which can be difficult even for a small region, so we design a genetic algorithm which efficiently generates desirable solutions, with the key components as follows.

Individual Representation: We use a numerical sequence to represent a route. Each number corresponds to a road intersection or vertex. For example, the sequence $[1, 2, 3, 4]$ represents the route of the UAV-mounted edge servers moving in the order of intersection 1, 2, 3, and 4.

Population Initialization: The initial population consists of multiple sequences of numbers. They are randomly generated by shuffling the indices of road intersections other than the charge station. We set the number of routes in region k to be M_k, and randomly choose $M_k - 1$ breakpoints in a sequence to divide it into M_k subsequences where each subsequence represents a route. To ensure that each route starts and ends at the charge station, we set the first and last number of each subsequence to the charge station's index. We check the sequence and add any missing edges to it to form a solution. The population size is set to 80.

Fitness Function: The fitness function assesses the total deployment cost in the region, encompassing the charge station's deployment cost and route costs using Eqn. 2. Lower deployment costs indicate better solutions.

Selection Operation: Retains individuals with lower deployment costs as parents for the next generation.

Crossover Operation: The selected parent individuals perform two-point crossover operations to generate new offspring individuals. The crossover operation selects two crossover points from the parent individuals and exchanges the sequence parts between these two points.

Mutation Operation: The offspring individuals perform mutation operations to increase the diversity of the population. We achieve individual mutation by randomly swapping the positions of two numbers in the sequence.

Flip Operation: Parent individuals choose two index values, and the sequence between these two index values (excluding the endpoints) is reversed to create a new offspring individual.

Iterative Evolution: The process of selection, crossover, mutation, and flip operations is repeated until the maximum specified number of iterations is reached.

The above GA algorithm can gradually improve the fitness of the individuals, and finally find a desirable solution.

Iterative Optimization: We design an iterative optimization method to find the best combination of the number of regions/bases K and routes M that minimizes the total deployment cost. In each iteration, we evaluate the deployment cost on the current combination of K and M and adjust these parameters for the next iteration accordingly.

We first utilize the K-medoid clustering algorithm to divide the map into K regions and set the medoids as the bases, where K ranges from 1 to 10. Next, for each region, we change the number of cruising routess M from 1 to 50 and employ the genetic algorithm to find the solution routes minimizing the deployment cost of each region.

Finally, we calculate the total deployment cost of the current solution to the charge station deployment and route planning problem and record the corresponding values of K and M. Through multiple iterations, we gradually approach the optimal solution which is the combination of K and M values that yields the lowest total deployment cost.

Performance Evaluation:

Experiment Setup: We obtained experimental data from GPS records of Shanghai buses [25]. We focused on a specific area within latitude 31.20 – 31.30 and longitude 121.40 – 121.50. To analyze the data effectively, we divided the map into 1x1 square kilometer cells. We filtered out cells without roads using a minimum threshold, eliminating GPS signal deviations. We processed the map using OpenCV to identify offloading points, representing GPS records. We extracted road network topology and shape features, identifying 527 road intersections. These intersections served as the basis for dividing the road network into road sections. The load for each road section was determined by the number of GPS records between two intersections, and the length of each section was obtained from the map.

In our evaluation, we compared our K-medoid and GA-based strategy, referred to as `KGA` with two benchmark algorithms, `GA` and `PSO`. While `GA` uses genetic algorithms without map division, `PSO` (Particle Swarm Optimization) employs a different individual update method, focusing on global search but possibly getting trapped in local optima. In contrast, `KGA` maintains population diversity and introduces new solutions to achieve global search.

Performance Results and Analysis: Fig. 2 displays the total deployment cost of different algorithms (KGA, GA, and PSO) as the number of routes changes. KGA outperforms PSO when $K < 7$, and it surpasses GA when K is between 2 and 4. The performance of KGA is influenced to a greater extent by the quantity of regions rather than the quantity of routes.

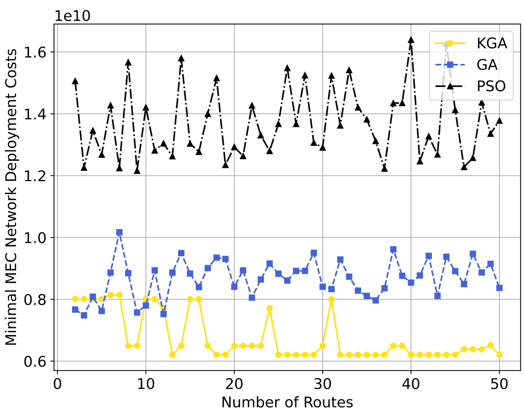

Fig 2 *Total Deployment Cost of Different Algorithms with Varying Number of Regions and Routes.*

In Fig. 3, we see a "v"-shaped trend in the optimal deployment cost and the number of routes for our KGA algorithm as the number of regions varies. Therefore, we claim that the optimal solution is to divide the map into $K = 4$ regions and maintain $M = 13$ routes.

In addition, Fig. 4 illustrates the deployment cost in each region when using KGA to divide the map into 4 regions, as per the optimal solution mentioned earlier. The deployment cost decreases initially with increasing routes but then rises. This trend is consistent with the observations in Figure 3. The optimal solution balances route length and density of UAV-mounted edge servers to minimize deployment cost.

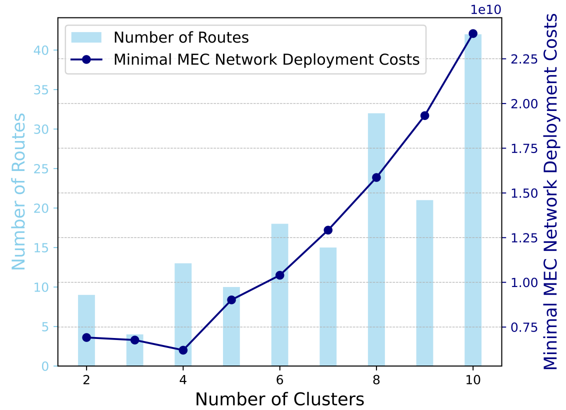

Fig 3 *The Minimum Total Deployment Cost and the Corresponding Number of Routes with Different Number of Regions.*

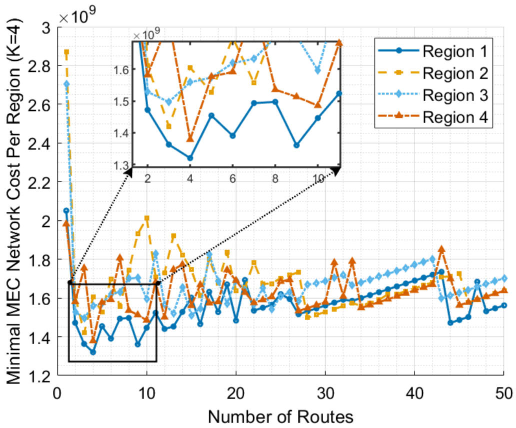

Fig 4 *The Deployment Cost of Each Region with Different Number of Routes (K=4).*

In summary, our KGA algorithm effectively optimizes charge station numbers, locations, and routes to reduce deployment cost. By iteratively adjusting the number of regions and routes, we achieve an optimal solution that reduces the cost by 48.94% compared to PSO and by 17.05% compared to GA.

Conclusion and Future Work: This paper proposed a route planning strategy to optimize the deployment cost of UAV-mounted edge servers, while meeting the computational offloading requirements of IoV end users. We built a model to characterize the deployment cost of deploying multiple UAV-mounted edge servers and planning their routes. A heuristic algorithm based on K-medoid clustering and genetic algorithm was proposed to solve the problem. K-medoid clustering was used to divide the map into multiple regions to reduce the problem scale, and the GA algorithm was utilized to plan the routes within each region. We designed an iterative optimization strategy by iteratively performing the K-medoid clustering and genetic algorithm to find the best combination of the numbers of the regions and routes. Experimental results verified that our proposed algorithm achieved the lowest deployment cost compared with other popular algorithms.

In our future work, we aim to improve IoV research using multi-objective optimization and distributed deep reinforcement learning. This will enhance factors like latency, network congestion, and reliability, enabling autonomous decision-making for UAV-mounted edge servers. Combining these methods will address IoV challenges, enhancing overall system performance and user experience.

Acknowledgment: This work was supported in part by the National Natural Science Foundation of China (Grant No. 62273107).

References

1. F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 10, pp. 1-15, 2014.
2. J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of vehicles: architecture, protocols, and security," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3701-3709, 2017.
3. M. N. O. Sadiku, M. Tembely, and S. M. Musa, "Internet of vehicles: An introduction," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 8, no. 1, pp. 11, 2018.
4. J. Zhang, and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 246-261, 2019.
5. S. M. A. Huda, and S. Moh, "Survey on computation offloading in UAV-Enabled mobile edge computing," *Journal of Network and Computer Applications*, vol. 201, pp. 103341, 2022.
6. Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 2322–2358, 2017.
7. L. Chen, J. Wu, G. Zhou, and L. Ma, “QUICK: QoS-guaranteed efficient cloudlet placement in wireless metropolitan area networks,” *The Journal of Supercomputing*, vol. 74, pp. 4037–4059, 2018, Springer.
8. G. Cui, Q. He, F. Chen, H. Jin, and Y. Yang, “Trading off between user coverage and network robustness for edge server placement,” *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 2178–2189, 2020, IEEE.
9. Y. Zhang, Y. Xu, H. Yin, and W. Wang, “Edge computing for UAV-assisted intelligent transportation systems: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4381–4399, 2020.
10. E. T. Michailidis, K. Maliatsos, D. N. Skoutas, D. Vouyioukas, and C. Skianis, “Secure UAV-aided mobile edge computing for IoT: A review,” *IEEE Access*, vol. 10, pp. 86353–86383, 2022, IEEE.
11. K. Sadatdiynov, L. Cui, L. Zhang, J. Z. Huang, S. Salloum, M. S. Mahmud, "A review of optimization methods for computation offloading in edge computing networks," *Digital Communications and Networks*, 2022, Elsevier.
12. K. Kim and C. S. Hong, "Optimal task-UAV-edge matching for computation offloading in UAV assisted mobile edge computing," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1-4, 2019, IEEE.
13. Q. Wei, Z. Zhou, and X. Chen, "DRL-Based Energy-Efficient Trajectory Planning, Computation Offloading, and Charging Scheduling in UAV-MEC Network," in *2022 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 1056-1061, 2022, IEEE.
14. W. S. Shi, J Cao, Q. Zhang, Y. H. Li, and L. Y. Xu, “Edge computing: Vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
15. A. R. Al-Ali, T. Y. Al-Naffouri, M. S. Alouini, and T. Başar, “Edge Computing and UAVs for 5G-Enabled Vehicular Networks: A Review, Some Research Issues, and Challenges,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 760–776, 2020.
16. A. Alshamrani, M. Alshehri, H. Alshaikhi, S. Alghamdi, and G. Fortino, “An edge computing system for vehicular networks using UAVs,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5156–5167, 2020.
17. E. Saurez, H. Gupta, A. Daglis, U. Ramachandran, "Oneedge: An efficient control plane for geo-distributed infrastructures," in *Proceedings of the ACM Symposium on Cloud Computing*, 2021, pp. 182-196.
18. W. Lv, Q. Wang, P. Yang, Y. Ding, B. Yi, Z. Wang, C. Lin, "Microservice deployment in edge computing based on deep Q learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2968-2978, 2022, IEEE.
19. X. Xiong, K. Zheng, L. Lei, L. Hou, "Resource allocation based on deep reinforcement learning in IoT edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1133-1146, 2020, IEEE.
20. H. Wang, J. Xiong, S. Chen, H. Huang, Y. Liang, X. Li, “A UAV-based edge computing system for smart transportation,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6578–6586, 2020.
21. Y. Wu, H. Li, B. Li, Y. Liang, “Mobile edge computing for unmanned aerial vehicle-assisted intelligent transportation systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3755–3766, 2019.
22. M.A.A. Al-Haidari, S.A. Al-Sumaidaee, H. Zedan, “UAV-enabled intelligent transportation systems: Opportunities and challenges,” *IEEE Access*, vol. 8, pp. 44247–44267, 2020.
23. M. Chen, Y. Zhang, W. Wu, and S. Mao, “Edge computing for intelligent transportation systems: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4135–4154, 2020.
24. P. Toth and D. Vigo, *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, 2002.
25. SUVnet Trace Data[OL]. http://wirelesslab.sjtu.edu.cn