I. Simulation Experiments

We conduct several experiments to investigate the performance of our dynamic ACO-A* algorithm in terms of QoS-satisfaction and time efficiency. These experiments were conducted under different scenarios, considering various urban layouts and population densities. **The source code of the simulation is available at https://github.com/QiyuJiangWuhan/SimulationCode.**

Representatively, we select two districts with distinctive road schema: 1) the regular structure for downtown Queens, New York, USA. 2) the complex structure for Zhongshan, Guangdong, China. The experiments are executed on the simulation scenarios of downtown Queens, New York, USA and Zhongshan, China. Relevant geographic information is collected from Google Maps.

Relevant information about the setting of the Queens area is as follows.

Setting 1 (Geographic information): We select a grid-like regular block area of roughly 1430m * 835m from Google Map and divide it into four types of neighborhoods with different colors shown in Fig.1 : red for residential areas, yellow for commercial areas, blue for the school district, and green for the entertainment area. Each type shares the same user distribution, Poisson Point Process in Eq.(**??**), but with different coefficients λ. Furthermore, population density also varies among blocks. We collect the urban population data from [1], through which it is clear to see the difference in density among block regions. We also consider the settings of data characteristics in [2], which describes the pattern of urban population distribution. Thus we can approximate the population by residential space area.

Setting 2 (The distribution of users): The distribution of users varies in two dimensions, the temporal dimension and the spatial dimension. Complying with the probability distribution of active users in [3], we assume there exists the identical rule and adopt the normal distribution $\mathcal{N}(\mu = 10, \tau = 3.6)$. Meanwhile, we apply the period during 12 a.m., which is the noon peak, and most users are also very active online.

Setting 3 (User walking speed): According to [4] describing, the walking speed distribution of pedestrians follows a normal distribution $\mathcal{N}(\mu = 1.34, s2 = 0.26)$ or remains at a constant speed of 1 m/s, as determined by McNett and Voelker. We determine the average speed of the user to be 1 m/s. In this context, the travel cost time will fulfill the maximum walking time restriction.

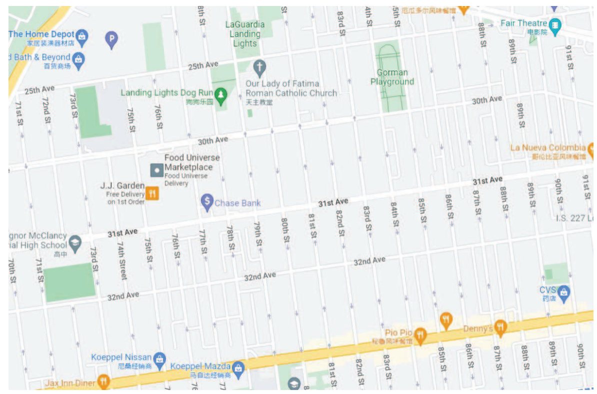
Setting 4 (Base station layout): Following the model proposed in Section II, we dispose the base stations beside the roads uniformly in the area. In this arrangement, the entire block area is certainly covered by 5G signals.

In summary, preciously speaking, the experimental simulation context settings are: we adopt the noontime slot between 12:30 a.m. and 13:30 a.m. The selected study areas will be divided into different sub-regions depending on the region's function and population property. Meanwhile, the population also varies with time.

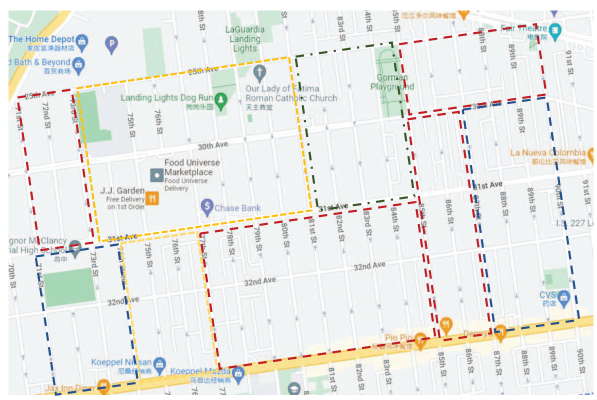
A. QoS Performance Analysis

To study the QoS performance of our proposed algorithm, we will first present a specific navigation scenario as an example to show detailed route information. Then, we will compare our algorithm with the latest proposed algorithm using different metrics.

A.1. Detailed navigation route: The simulation experiment compares our navigation system with Google Maps and is conducted on the regular pattern streets in downtown Queens since the structured layout conveys the route information more explicitly. Detailed selected district and block divisions are shown in Fig.1. We assume the user starts from the top right corner vertex to the left bottom vertex.



(a) Regular Structure Map

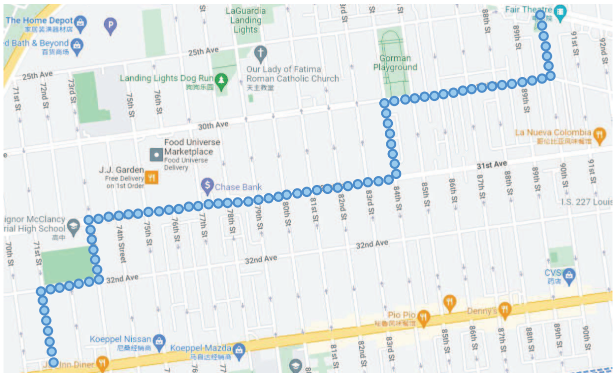


(b) Block Groups

Fig. 1: The Street Map And Subdivisions. (a) presents the specific selected regular region in the experiment. (b) displays the particular divisions according to *Setting 1*.

Fig.2 shows the comparison results. In contrast with our QoS-aware navigation, the solution of the Google Maps shown in Fig.2(b) contains roads in red which are congested during the navigation, thus will cause users a bad experience.

From our division assumption and collected data, we can conclude that the residential areas and the school areas are fairly crowded due to a large number of flows at noon peak time. In order to balance the multiple objectives, our navigation algorithm avoids the congested blocks to ensure



(a) QoS-aware navigation route

(b) Traditional navigation route

Fig. 2: QoS-satisfying routes comparison.

the QoS guarantee and searches for the shortest path route to ensure the minimum cost. From Fig.2, we can see that our navigation bypasses the associated overcrowded districts timely.

A.2. Performance with different metrics: To further investigate the effectiveness of our navigation algorithm, we will compare our algorithm with the newly proposed single source shortest path (sssp) algorithm based on Dijkstra's algorithm using different metrics.

Fig. 3: Complex Structure Map

The complex structure map shown in Fig.3 is adopted as a research scenario due to its complex environment containing various multi-metrics cases. In our experimental conditions, the road states are divided into four gradients, numbered 0, 1, 2, and 3, from low to high, indicating that QoS is not generally satisfied, fairly satisfied, and very satisfied, respectively. The experiments are developed under the following settings:

1. Two types of test cases. The experiments are executed on two types of test cases, one with the starting-end vertex pair as the variable and the other with the time period, i.e., population density, as the variable.

2. Two evaluation metrics. There exist two metrics to evaluate the QoS performance. One is to examine the percentage of roads with a status above 0 in the total number of roads, and another is to determine the respective percentage of status for each gradient.

The comparison results of relevant experiments are presented in Fig.I-A2.

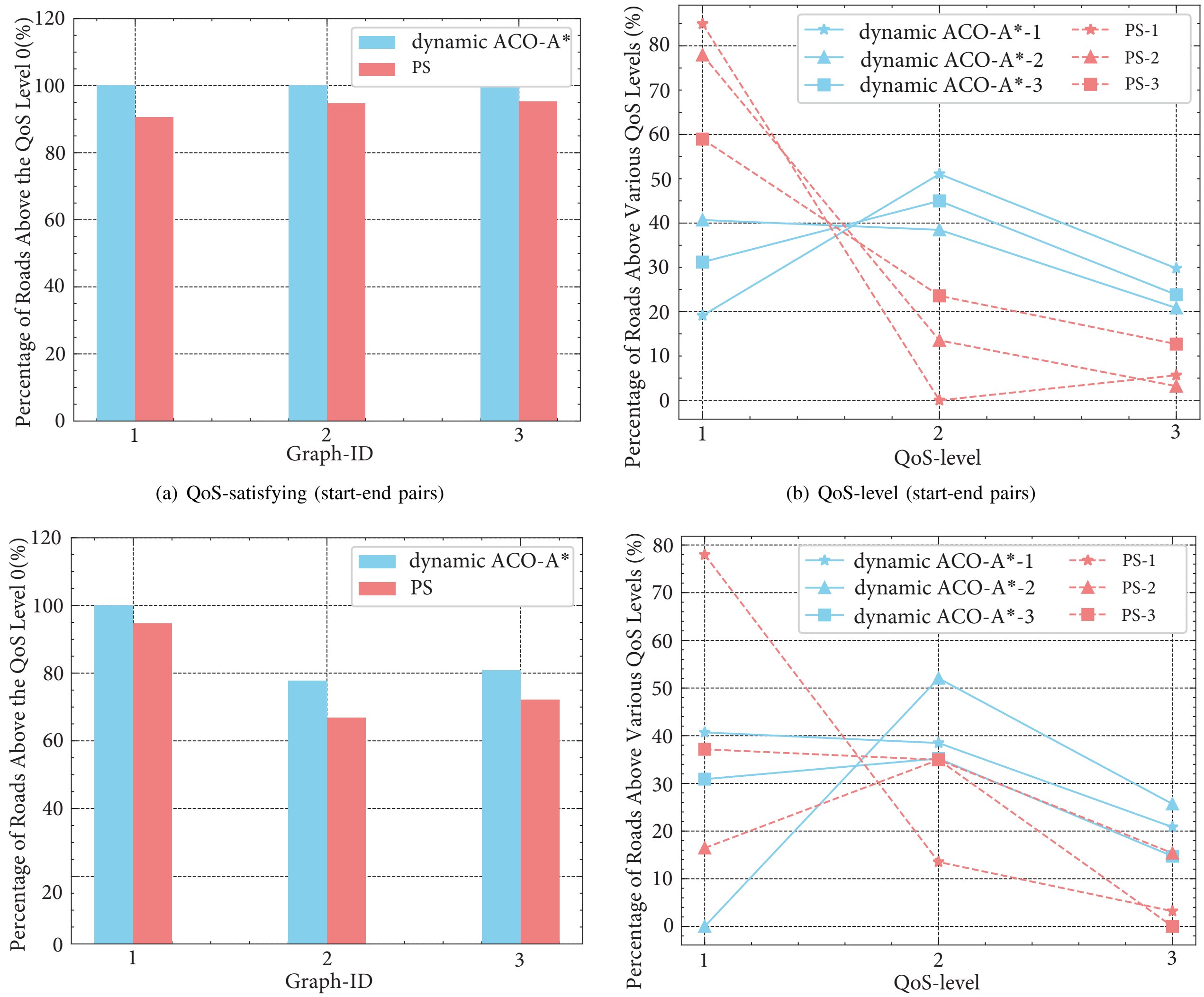
According to the experiments data shown in Fig.I-A2, we can analyze the results and obtain relevant conclusions:

1. The start-end pairs variable: First, in the QoS-satisfying case shown in Fig.I-A2(a), our dynamic ACO-A* algorithm can ensure that the QoS quality above users' criterion throughout the whole route process (ID 1-3), while there always exist some routes below the requirement when navigating with sssp algorithm. Then in the QoS-level case presented in Fig.I-A2(b), the QoS level in our routes is relatively stable (dynamic ACO-A* 1-3) compared with that in sssp solving routes. Besides, the main body of our routes is concentrated in levels 2 and 3, which possess the higher quality. Thus it can be analyzed that the percentage distribution of QoS levels approximates a normal distribution. Therefore, the QoS-aware paths obtained by our algorithm are planned routes with a mean value μ of about level 2, which can generally meet the network quality requirements of users.

2. The time period variable: In the QoS-satisfying case described in Fig.I-A2(c), due to changes in population density, some roads are always congested during rush hours (ID 2-3). However, our algorithm can still ensure more routes meet the demand in contrast with the sssp algorithm. As for the QoS-level case shown in Fig.I-A2(d), its comparison effect is similar to the case depicted in Fig.I-A2(b), and our algorithm still performs better whether some roads are permanently blocked (dynamic ACO-A* 2-3) or not (dynamic ACO-A* 1). Thus, for cases like Graph ID 2 and 3, the paths obtained by our algorithm can still be considered as approximately normally distributed with a mean value μ around level 2, but with larger variance values τ compared to the cases like Graph ID 1.

B. Efficiency Performance Analysis

To investigate the time efficiency of the proposed algorithm, we measure the algorithm on the rather more complex map shown in Fig.3, where the time consumption situations are more complicated and deservedly more convincing. All algorithms were implemented in C++ and run on a Windows 10 platform with a AMD Ryzen 5 4600U with Radeon Graphics processor (2.1GHz) and 16384MB RAM.



(a) QoS-satisfying (start-end pairs)

(b) QoS-level (start-end pairs)

(c) QoS-satisfying (time period)

(d) QoS-level (time period)

Fig. 4: Qos performance with two evaluation metrics.

In the complex structure map, we execute the experiments with several different starting-end vertex pairs and different time ranges as variables, aiming at testing our algorithm in various scenarios. These input data are processed by the primary dynamic A* algorithm and the dynamic ACO-A* algorithm. With the same optimal solution but certainly a gap in time consumption, the comparison of the primary and the optimized can sufficiently demonstrate the optimization effects.

The comparison results are presented in Fig.5.

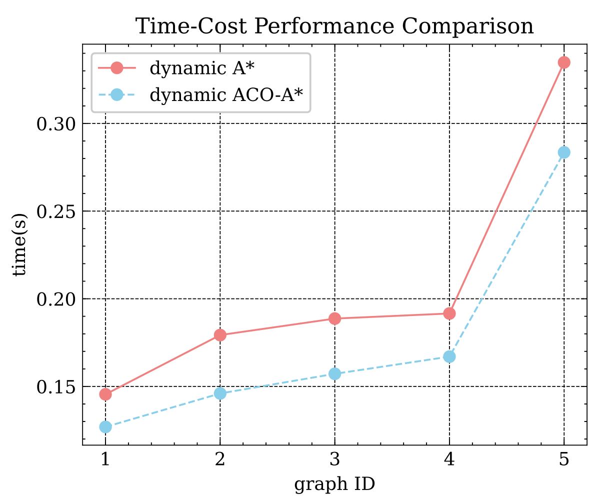


Fig. 5: Efficiency Comparison

In the experiment, the size of the input graph increases. As a result, we can see that the running time also increases. From the comparison of the results, we can see that with the help of the ACO reference values, the performance is improved by a factor of 1 to 1.5 in terms of time consumption.

In summary, the dynamic ACO-A* algorithm can well balance the QoS requirement and obtain the relatively shortest path with low cost in time consumption.

REFERENCES

[1] statisticalatlas, "Population of east elmhurst, new york, new york," https://statisticalatlas.com/neighborhood/New-York/New-York/East-Elmhurst/Populationx.

[2] J. Kim, V. Sridhara, and S. Bohacek, "Realistic mobility simulation of urban mesh networks," *Ad Hoc Networks*, vol. 7, no. 2, pp. 411–430, 2009.

[3] S. Ji, Z. Cai, J. S. He, and R. Beyah, "Primary social behavior aware routing and scheduling for cognitive radio networks," in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2015, pp. 417–425.

[4] F. Legendre, V. Borrel, M. D. De Amorim, and S. Fdida, "Reconsidering microscopic mobility modeling for self-organizing networks," *IEEE Network*, vol. 20, no. 6, pp. 4–12, 2006.