SHS Web of Conferences **144**, 03020 (2022) https://doi.org/10.1051/shsconf/202214403020
*STEHF 2022*

# Communication Mechanisms Comparison and Decision Algorithm Optimization in Vehicle Edge Computing

Yushen Xun*

Electronic Information Engineering, Nanjing Tech University, Jiangsu province, China, 214000

**ABSTRACT:** In recent years, a new network mode, Vehicle Edge Computing (VEC), has been introduced into vehicle networks to increase their computing power. As modern vehicle applications continue to evolve, the ultimate challenge of meeting communication and computing needs becomes more and more prominent. However, the performance of various commonly used communication mechanisms in transmitting different types of information has rarely been studied and compared. This is the main problem in the development of VEC at present. This paper presents the results of testing the communication modes of WiFi, LTE and DSRC by using existing models. Evaluation of different aspects of their performance is referred to. At the same time, a new VEC offload algorithm is analyzed. The result is that the three different communication mechanisms of indoor VEC each have their own advantages and disadvantages in the transmission of different types of information. Compared with conventional algorithms, the new algorithm can complete tasks faster and better, making up for some shortcomings of existing algorithms.

## 1. INTRODUCTION

With the development of some emerging technologies, such as autonomous driving and augmented reality (AR) technology [1], more and more data from sensors (about 1GB /s) needs to be analyzed and fused, which requires complex data processing capacity and larger storage capacity. Traditional on-board networks face significant storage and communication challenges [2, 3]. In addition, communication between vehicles and cloud servers requires higher bandwidth. With the increasing number of mobile terminals. The demand for bandwidth for information transmission is also getting higher and higher. Therefore, how to meet the needs of data processing and transmission in a vehicle network has become a challenge. The VEC technology, which integrates cloud computing with VANETS, follows. Unlike centralized on-board cloud computing services, in edge computing, data processing and analysis takes place close to the terminal device. Since computing and storage services are located close to the user (at the edge), this can provide better QoS, especially in applications requiring low latency. In addition, we still need strong communication capabilities and computing mechanisms to support the application of contemporary vehicle-mounted networks [4].

In many papers, only WiFi will be discussed in terms of communication mechanisms [5,6]. Many of the less common but predictable communication mechanisms are not discussed. There are also a few comparisons of different communication mechanisms in VEC. This paper describes some prototypes of VEC that support WiFi, LongTerm Evolution (LTE) and Dedicated Short Range Communication (DSRC). The evaluation includes latency, power consumption, and system utilization. This paper summarizes some results on the reliability and efficiency of the three communication methods in the VEC application. When there are multiple vehicles with on-board network offloading tasks simultaneously, the data transfer rate will be low and the delay will be long due to the disregarded for resource allocation. If the vehicle performs part of the task locally and uploads the parts that cannot be processed, the results can be obtained more quickly. Therefore, reasonable task scheduling schemes and communication allocation have an important influence on the performance of the VEC system.

Most of the current studies only carry out simple allocation of resources when performing task offloading, and do not consider the joint decision of communication and computing to make a more reasonable allocation of resources. This paper will analyze a new edge computing algorithm. The paper compares the advantages and disadvantages of LTE, WiFi and DSRC in terms of delay, power consumption and system utilization. Based on the simulation results of the reference model, the new proposed edge computing algorithm is analyzed and evaluated.

## 2. METHOD

To compare different communication mechanisms in the VCE application, the author uses the existing platform in simulation software for data analysis and comparison. Their setup and the equipment required are discussed.

*Corresponding author. Email: s1554081661@163.com

In this research, TPlink routers are used to establish hotspots for WiFi-based communication. It chooses the 5GHz band. To establish the LTE based communication network, two VERT2450 antennas and two Ettus Research USRP B210 boards are used in the paper, as well as uHD software drivers. For DSRC-based communication, a moKar DSRC device that has two antennas was used.

WiFi is one of the most widely used wireless communication methods for data transmission. WIFI can also be used in VEC. The WiFi frequency used in this article is 5GHz. When the OBU and RSU are connected to the router, they can ping each other after obtaining IP addresses. After WiFi connection, the following discussion is based on ROS data communication. ROS is data middleware, specially designed for autonomous vehicle communication transmission and data sharing on the Internet of Things.

Compared with WiFi in mobile environments, LTE has great advantages in terms of communication coverage and performance. With the development of software, the programmability of wireless communication equipment has been greatly improved. Using the SDR board, srsLTE and other open source Software, it is easier to build dedicated LTE communication test stands [7]. Applications can be developed after the RSU and OBU ping each other. The default frequency of srsLTE named EARFCN 3400 was used in this experiment, with 2685MHz for the download link and 2565MHz for the upload link.

# 3. COMPARISON AND CONCLUSION

To evaluate the performance differences among various communication modes, the paper carries out experimental tests and comparisons from three aspects. They are latency, power consumption and system availability.

## 3.1. Latency

Latency is an important metric for evaluating the performance of communication-dependent applications. The section measures end-to-end latency in two different directions: from RSU to OBU and from OBU to RSU. This article measures the time difference in message transfer by using clockdiff in Linux. Latency is then calculated by using the time difference previously measured. The results are shown in table 1.

**Table 1.** The latency in different directions

| | LTE | WiFi | DSRC | |
|---|---|---|---|---|
| BSM | **3.11** | 67.86 | 8.44 | **RSU to OBU（ms）** |
| image | 26394.34 | **3472.87** | / | |
| Ping | **25.13** | 51.48 | / | |
| BSM | 51.13 | 14.93 | **9.79** | **OBU to RSU（ms）** |
| image | 27884.1 | **2593.48** | / | |
| Ping | **28.27** | 67.19 | / | |

When RSU acts as sender and OBU acts as receiver, LTE has significant performance advantages in Ping and BSM compared to the other two. WiFi is more suitable for transmitting image information. From OBU to RSU, the three communication modes have their own advantages and disadvantages. This time DSRC performance is better than LTE in BSM. The other two are basically consistent with the above conclusion. In general, LTE communication outperforms WiFi in the other two aspects except for the poor image performance. Moreover, with the change of transmission direction, LTE and WiFi delays may become unstable when transmitting information from different directions. The DSRC latency is stable regardless of direction.

HydraOne was used in the experiment to test the time latency in a mobile environment [8]. In this paper, BSM messages are selected to measure the performance of three communication modes in terms of latency when RSU is used as the transmitter and OBU as the receiver. As can be seen from the results in table 2, when HydraOne's speed changes, DSRC's difference latency is very stable, while LTE's difference in varies greatly. WiFi is also stable.

**Table 2.** The latency of BSM in different speeds

| Latency（ms） | LTE | WiFi | DSRC |
|---|---|---|---|
| 0m/s | 3.17 | 15.08 | 8.89 |
| 3m/s | 655.42 | 12.93 | 9.08 |
| 5m/s | 1070.79 | 16.01 | 9.19 |

## 3.2. Power consumption

In this paper, power consumption of hardware is measured by using a software named Watt's Up [9]. The power consumption of a processor is measured by rapL. The following four different transmission processes were measured. As can be seen from figure 1, WiFi is the most energy efficient of all the measurement results. DSRC consumes a lot of energy. In addition, compared to other types of information, more power will be consumed when image information is transmitted.

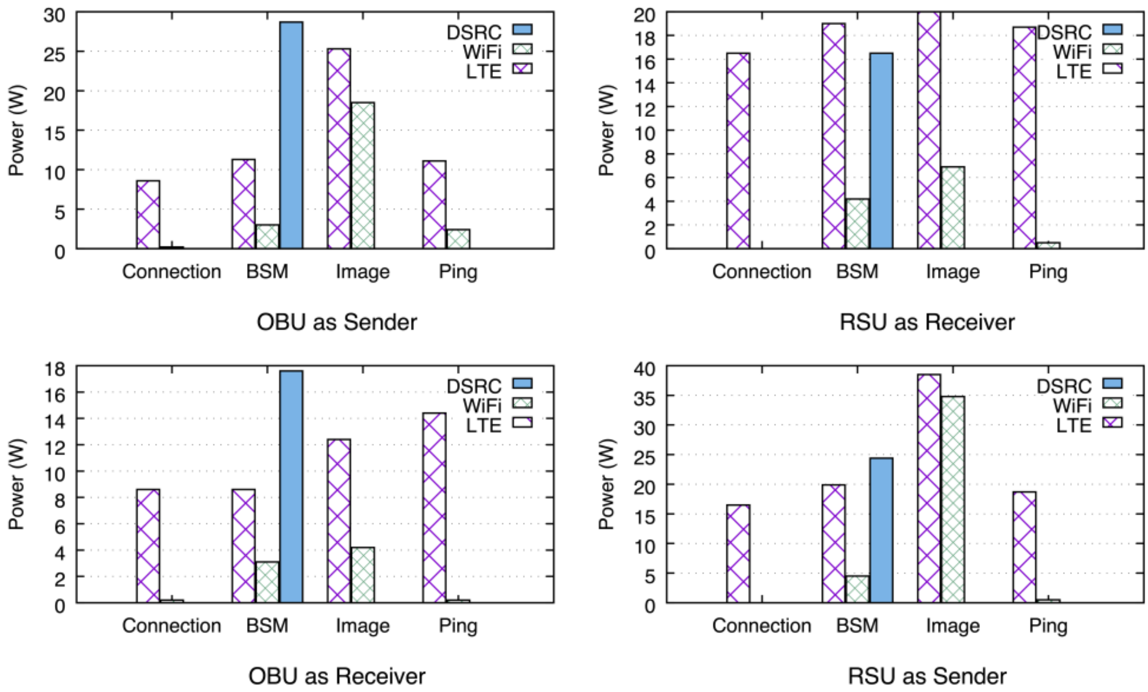


**Figure 1** Power consumption

### 3.3. System Utilization

To measure system utilization, the top is used to obtain CPU utilization. As can be seen from table 3, in most cases, DSRC has the lowest CPU utilization. When transmitting image information, whether WiFi or LTE, the CPU utilization of the sender is higher than the receiver. When using WiFi to transmit information, in most cases, the system utilization is generally higher than with DSRC and LTE.

**Table 3.** The CPU utilization of RSU/OBU as sender and receiver

| | msg | LTE | WiFi | DSRC | |
|---|---|---|---|---|---|
| sender | BSM | 1.00 | 2.61 | 1.03 | RSU(%) |
| | image | 120.48 | 100.63 | / | |
| receiver | BSM | 1.17 | 120.53 | 0.10 | |
| | image | 1.00 | 9.03 | / | |
| sender | BSM | 1.20 | 1.13 | 1.09 | OBU(%) |
| | image | 91.44 | 99.72 | / | |
| receiver | BSM | 1.24 | 1.10 | 0.12 | |
| | image | 1.37 | 1.72 | / | |

### 3.4. Conclusion

DSRC transmits BSM at high speed and excellent performance. WiFi has the smallest delay in transmitting images, but the latency is serious when BSM and ping are transmitted. In the experiment, LTE performed well in the transmission of BSM and Ping, but poorly in the transmission of images. So it can be argued that LTE is better suited for transmitting small messages. The latency of WIFI and DSRC is stable at different speeds. But the latency of LTE varies dramatically when speed changes. The power consumption of LTE is generally high in most cases, while WiFi power consumption is low. The system utilization is very high when sending images, which still needs to be improved.

## 4. MULTI-OBJECTIVE VEC TASK SCHEDULING ALGORITHM

The existing problems with the bat algorithm [10] are as follows. According to the updating method, at the beginning of each iteration, individual bats fly randomly to conduct a global search. When the individuals found a better position, all the bats moved toward the better position. This operation is repeated many times. As the number of bats increased, the closer they got, the uneven distribution of the population as a whole led to an obvious decline in population diversity. At the same time, as the bat individuals are getting closer to the optimal individuals of the population, the convergence rate greatly reduces or even stagnates, and the population loses the ability to undergo further evolution. Therefore, it is difficult for the bat algorithm to find the global optimum distributed in the local optimum neighborhood [11].

Therefore, an elite strategy (NSGA-II) [12] was introduced into the bat algorithm to reduce the influence of local optimization. Elite strategies combine parent and offspring populations, allowing the two populations to compete with each other to produce the next generation. under the circumstance, elite individuals among the parents will not be discarded, which can greatly speed up the convergence of the algorithm.

```
while condition:
    Rt = Pt + Qt
    F = fast_nondominate_sort( Rt )  // Fast non-dominate sort
    Pt+1 = [ ]
    i = 0
    while len(Pt+1) + len( F[i] ) < N:
        crowding_distance_assignment( F[i] )  // Density estimation
        Pt+1 += F[i]
        i += 1
    Pt+1 += F[i][0:N-len(Pt+1)]
    Qt+1 = make_new_generation( Pt+1 )
    t = t+1
```

**Figure 2** Pseudo code of the NSGA-II main circulation section [14]

Based on the above improvements, an improved VEC task offloading algorithm based on bat algorithm will be introduced in figure 3.

**Input:** Task $\{T_i, i \in N\}$, MEC server $\{C_j, j \in M\}$, link duration $T_{ij}$, resource block $KV$
**Output:** Task scheduling scheme $S_p$
1. Calculate time consumption $t_{i0}$ in local computing mode, and construct duration matrix $D$;
2. Determine the objective functions, and set population size $n_p$ and maximum number of iterations $\lambda$;
3. Initialize $n_p$ bat positions $x_k^0$ using $K$-means clustering;
4. **for** $k = 1 \to n_p$ **do**
5. Initialize speed $v_k^0$, loudness $A_k^0$, emission rate $r_k^0$;
6. Generate pulse frequency $f_k$ according to equation (4);
7. Add bat $k$ to initial population $P_R^0$;
8. Calculate the objective function values of all bats, and select Pareto-optimal set $S_p$;
9. **for** $t = 0 \to \lambda - 1$ **do**
10. Randomly select a solution $x^p$ from $S_p$;
11. **for** $k = 1 \to n_p$ **do**
12. Update speed $v_k^{t+1}$ according to equation (10);
13. Generate a random number $rand1$;
14. **if** $rand1 \leqslant r_k^t$ **then**
15. Update position $x_k^{t+1}$ according to equation (6);
16. **else**
17. Generate position $x_k^{t+1}$ according to equation (11);
18. Calculate the objective function values in new positions $x_k^{t+1}$;
19. Generate a random number $rand2$;
20. **if** $rand2 < A_k^t$ and $x_k^{t+1}$ dominate $x_k^t$ **then**
21. Accept $x_k^{t+1}$ temporarily to intermediate population $P_I^t$;
22. Adjust $A_k^{t+1}$ and $r_k^{t+1}$ according to equations (7) and (8);
23. Combine $P_R^t$ and $P_I^t$ into population $P^{t+1}$;
24. Reorder $P^{t+1}$ using fast non-dominated sorting and crowding-distance estimation;
25. Select the first $n_p$ bats to form population $P_R^{t+1}$;
26. Select bats with rank 1 in $P_R^{t+1}$ to form $S_p$.

**Figure 3** An improved VEC task algorithm based on bat algorithm

To comprehensively evaluate the performance of this improved algorithm, this paper will compare it with the other three algorithms. The other three algorithms are shown below:

Local computing (LC): In this case, the vehicles use only local resources for computing tasks. Do not participate in task offload. The following experiments will use this algorithm as a benchmark for comparison.

Fairness-driven allocation (FDA): If the user vehicle cannot meet the requirement through local computing, the system will send the resource equally to the vehicle that sent the offload request. The offload request will be sent to a random MEC vehicle. Computing resources are also distributed equally by MEC servers. So that the needs of user vehicles can be met.

Priority-driven allocation (PDA): The pre-allocation process is similar to the above algorithm. However, this algorithm does not allocate exactly the same resources to all user vehicles. Instead, there will be a more equitable allocation of resources based on the weight of the task. More resources are for more demanding users.

### 4.1. Performance evaluation

Eclipse SUMO and Network Simulator3 are used as Simulation software in this paper. The Eclipse SUMO is used to simulate the operation of city vehicles. Network Simulator3 is used to simulate communication and task offloading. This experiment will comprehensively evaluate the four algorithms in two specific scenarios. The average execution time it takes to complete all the tasks and the task offload ratio are talked about.

In the first scenario, the range of the vehicle cluster is extended to a radius of 100 meters, and the number of vehicles in the cluster is 50. As the number of vehicles increases, the improved algorithm always has the shortest average execution time. LC has the longest average execution time because tasks are slow when local computing resources are not sufficient for computing needs. When the offloading algorithm is used, the user vehicle can send the computing task that cannot be completed in time to the MEC vehicle. The MEC vehicle can process the task with its own rich computing resources and then send the result back after completing the processing. The average time it takes to complete a task this way is less than the time it takes to complete a task by using its own computing resources alone. The specific data results are shown in figure 4.

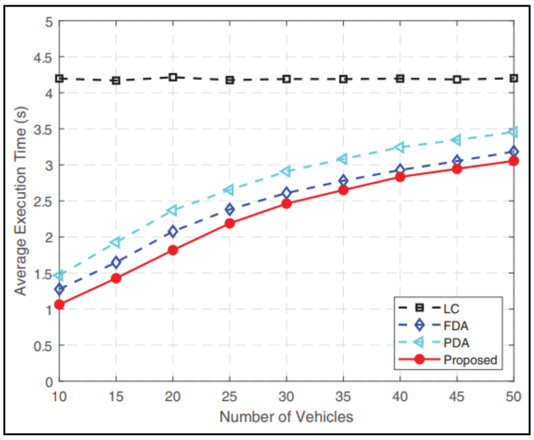


**Figure 4** Average execution time with different number of vehicles

https://doi.org/10.1051/shsconf/202214403020

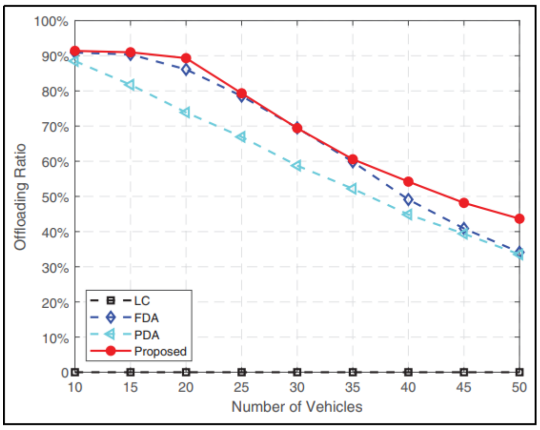


**Figure 5** Offloading ratio with different number of vehicles

With the increase of vehicles, users compete more fiercely for resources. At this point, the PDA allocates resources to different users in different proportions. Heavier users who need more resources get more resources. In this optimization, considering both the average task time and different task weights, PDA performs better than FDA and can obtain a shorter average time. The specific data results are shown in figure 5.

In the second scenario, the number of vehicle clusters remained unchanged at 30 and the radius of vehicle clusters varied between 50 and 250 meters. The results are shown in figure 6 and 7. Similarly, local computing algorithm performed poorly both in execution time and in offloading rate because it just relies on local resources for computing. As the radius increases, the signal-to-noise ratio of vehicle-to-vehicle communication decreases. So the bandwidth needed to transmit information will increase. Spectrum resources will become more strained. The time required to transmit the same information also increases. As shown in figure 6, the new algorithm has very good transmission speed because it takes into account both the bandwidth required for information transmission and the state of computing resources. This will provide users with a better wireless transmission channel. In figure 7, with the increase of radius, the offloading ratio of all the algorithms decrease. It can be seen from the data that the performance of the new algorithm is better than the other three algorithms. When the radius is larger than 150 m, the new algorithm can still maintain a high offloading ratio, which is conducive to the completion of the calculation task.

According to the comparison between Scenario 1 and Scenario 2, it can also be found that the change in average execution time and offload ratio due to the increase of vehicles is more obvious. The increase in radius leads to a more gradual change. This also indicates that the number of offloading tasks has a greater influence on the performance of the offloading algorithm than the signal transmission distance.

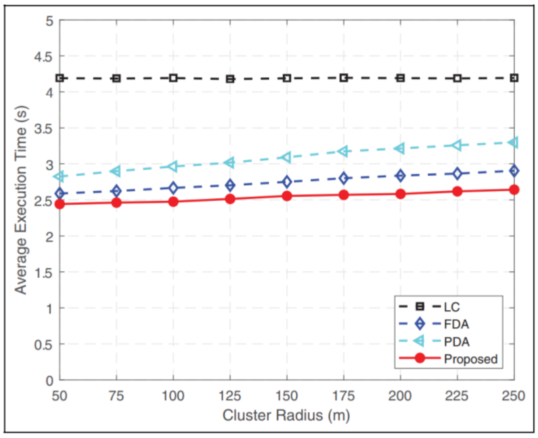


**Figure 6** Average execution time with different cluster radius

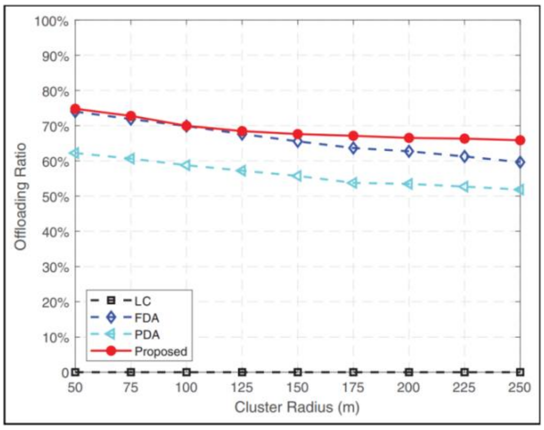


**Figure 7** Offloading ratio with different cluster radius

## 5. CONCLUSION

This paper compares three communication mechanisms of indoor VEC, summarizes three observations for developing practical applications of indoor VEC, and analyzes the information that each communication mode is better at transmitting. At the same time, a new edge computing task offload algorithm is analyzed. In this algorithm, resources are allocated more rationally. Simulation results show that compared with the existing algorithm, the new algorithm can complete the task faster and better, and can make up for the shortcomings of the current algorithm. The paper does not mention the difference between the three communication methods at faster speeds(>30m/s) and the results of previous experiments. Longer communication lengths(>1000m) may also alter some of the results referred to in the paper. Experiments show that the bandwidth difference between download and upload causes a large change in LTE latency. The root cause of this has yet to be studied. LTE and DSRC communication connections have non-negligible power consumption. Image transmission is still expensive. How to save more resources in vehicle communication is still worth studying in the future.

## REFERENCES

1. USRP B210, 2019. Available at https://www.ettus.com/all-products/ub210-kit/.
2. Min Chen and Yixue Hao. Task offloading for mobile edge computing in software defined ultra-dense network. IEEE Journal on Selected Areas in Communications, 36(3) (2018) pp. 587–597.
3. Nima Eshraghi and Ben Liang. Joint offloading decision and resource allocation with uncertain task computing requirement. In IEEE INFOCOM 2019-IEEE Conference on Computer Communications, pages 1414–1422. IEEE, 2019.
4. Navid Nikaein, Mahesh K Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. OpenAirInterface: A flexible platform for 5G research. ACM SIGCOMM Computer Communication Review, 44(5) (2014) pp. 33–38.
5. Liangkai Liu, Xingzhou Zhang, Qingyang Zhang, Andrew Weinert, Yifan Wang, and Weisong Shi. AutoVAPS: an IoT-enabled public safety service on vehicles. In Proceedings of the Fourth Workshop on International Science of Smart City Operations and Platforms Engineering, pages 41–47, 2019.
6. Junjue Wang, Ziqiang Feng, Shilpa George, Roger Iyengar, Padmanabhan Pillai, and Mahadev Satyanarayanan. Towards scalable edge-native applications. In Proceedings of the 4th ACM/IEEE Symposium on Edge Computing, pages 152–165, 2019.
7. Ismael Gomez-Miguelez, Andres Garcia-Saavedra, Paul D Sutton, Pablo Serrano, Cristina Cano, and Doug J Leith. srsLTE: an open-source platform for LTE evolution and experimentation. In Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization, pages 25–32, 2016. https://dblp.org/rec/conf/hotedge/WangLZS19.html
8. Yifan Wang, Liangkai Liu, Xingzhou Zhang, and Weisong Shi. HydraOne: an indoor experimental research and education platform for CAVs. In 2nd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 19), 2019.
9. Hirst, J. M., Miller, J. R., Kaplan, B. A., & Reed, D. D. Watts Up? Pro AC Power Meter for Automated Energy Recording. Behavior Analysis in Practice, 6(1) (2013) pp.82–95. doi:10.1007/bf03391795
10. Yang X. A new metaheuristic bat-inspired algorithm. In: Gonza´lez JR, Pelta DA, Cruz C, et al. (eds) Nature inspired cooperative strategies for optimization (NICSO), Springer, Berlin. Vol. 284. 2010, pp.65–74.
11. Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE T Evolut Comput 2002; 6(2) (2002) pp.182–197.
12. AquilaEAG. NSGAII (Genetic Algorithm for Non-dominated Sorting with Elitism), 2018, https://blog.csdn.net/weixin_43202635/article/details/82708916