

# Multi-objective Optimal Roadside Units Deployment in Urban Vehicular Networks

Weian Guo *Member, IEEE*, Zecheng Kang, Dongyang Li, Lun Zhang, and Li Li *Member, IEEE*

arXiv:2402.18581v1 [cs.NE] 14 Jan 2024

**Abstract**—The significance of transportation efficiency, safety, and related services is increasing in urban vehicular networks. Within such networks, roadside units (RSUs) serve as intermediates in facilitating communication. Therefore, the deployment of RSUs is of utmost importance in ensuring the quality of communication services. However, the optimization objectives, such as time delay and deployment cost, are commonly developed from diverse perspectives. As a result, it is possible that conflicts may arise among the objectives. Furthermore, in urban environments, the presence of various obstacles, such as buildings, gardens, lakes, and other infrastructure, poses challenges for the deployment of RSUs. Hence, the deployment encounters significant difficulties due to the existence of multiple objectives, constraints imposed by obstacles, and the necessity to explore a large-scale optimization space. To address this issue, two versions of multi-objective optimization algorithms are proposed in this paper. By utilizing a multi-population strategy and an adaptive exploration technique, the methods efficiently explore a large-scale decision-variable space. In order to mitigate the issue of an overcrowded deployment of RSUs, a calibrating mechanism is adopted to adjust RSU density during the optimization procedures. The proposed methods also take care of data offloading between vehicles and RSUs by setting up an iterative best response sequence game (IBRSG). By comparing the proposed algorithms with several state-of-the-art algorithms, the results demonstrate that our strategies perform better in both high-density and low-density urban scenarios. The results also indicate that the proposed solutions substantially improve the efficiency of vehicular networks.

**Index Terms**—Vehicular Networks, Roadside Units Deployment, Data Offloading, Multi-objective Optimization, Constraints.

## I. INTRODUCTION

WITH the accelerated expansion of urbanization, traffic networks have grown significantly. To provide a safe and efficient traffic environment, Vehicle-to-Infrastructure (V2I) framework in the Intelligent Transportation System (ITS) gains prominence. Since roadside units (RSUs) provide data transfer services for vehicles in such a framework, it is essential to deploy RSUs at optimal locations to assure quality of service (QoS) in complex urban environments [1].

This work is supported by the National Natural Science Foundation of China under Grant Number 62273263, 72171172 and 71771176; Shanghai Municipal Science and Technology Major Project (2022-5-YB-09); Natural Science Foundation of Shanghai under Grant Number 23ZR1465400. (Corresponding author: Li Li).

Weian Guo, Zecheng Kang, and Dongyang Li are with Sino-German College of Applied Sciences, Tongji University, Shanghai, China (email:{guoweiyan, kangzecheng, lidongyang0412}@163.com). Lun Zhang is with School of Transportation, Tongji University, Shanghai, China (email:lun\_zhang@tongji.edu.cn). Li Li is with the Department of Electronics and Information Engineering, Tongji University, Shanghai, 201804, China (email:lili@tongji.edu.cn).

To provide optimal RSU deployment in urban areas, the challenges can be summed up in two aspects. First, many factors with potential conflicting perspectives, such as quality of service and economics of deployment, should be taken into account simultaneously in real-world implementations [2]. Second, the urban environment is characterized by a large number of vehicles, various obstacles and structures, and a variety of road types. Hence, it is vital to contemplate the appropriate placement of RSUs to alleviate signal interference resulting from intricate topographies [3]. Therefore, an optimal RSU deployment should provide effective and cost-efficient communication services to guarantee the reliability, and stability of connectivity [4].

In existing studies, a part of the focus is on optimizing several single objectives using mathematical programming methods, heuristics [5], [6] or data-driven approaches [7], which may not be competent to simultaneously balance the trade-offs between objectives from different perspectives. In another part of the research, multi-objective optimization modeling is used, but it disregards the complexity of actual implementations in urban environments, such as physical landscapes and excessive RSU density [8]. In order to address these challenges, this study incorporates techniques for addressing constraints into a multi-objective optimization algorithm. This approach aims to consider the impact of actual urban terrain on the RSU deployment. Furthermore, during deployment, the data offloading is also modeled to take into account the stability and reliability of the connection between vehicles and RSUs [9].

To investigate optimal RSU deployments, we consider both communication quality and development cost together in the problem modeling. Since the landscape is gridded into hundreds of map pieces, the problem belongs to large-scale optimization issues. Meanwhile, the road terrain and buildings also bring constraints. To deal with these problems, this paper proposes an improved NSGA-III algorithm that uses a multiple-population strategy and an adaptive balancing strategy to balance exploration and exploitation in a large-scale optimization search space. Furthermore, it incorporates an offspring calibration mechanism to prevent excessive RSU density. The contributions of this paper can be summarized as follows:

- First, we propose a multi-objective deployment model for RSUs in urban scenarios. To evaluate the QoS in communications, we have set time delays as objectives. In the meantime, the construction budget is also an objective. For the constraints, we consider the elements in urban environments, including road maps, building

landscapes, RSU density, and so forth. In vehicle-RSU connections, a strategy named iterative best response with sequential game (IBRSG) is proposed for data offloading.

- Second, we improved the NSGA-III algorithm by adding a multi-population strategy and an adaptive balancing strategy for exploring and exploitation in the search space. This helps solve the proposed multi-objective optimization problem with constraints. To further optimize the density of RSUs, we also present a distance-inhibition method for calibrating solutions during the optimization process.

The remaining sections of this paper are structured in the following manner: Section II presents a comprehensive overview of the existing literature pertaining to the RSUs deployment. In Section III, a multi-objective problem with constraints is modeled, specifically targeting urban scenarios. In order to handle the problems, an enhanced multi-objective optimization algorithm is elaborated upon in Section IV. The experiments are conducted in Section V, where comparisons and related analyses of the results are also performed. Finally, the conclusions and future work are presented in Section VI.

## II. RELATED WORK

The RSUs deployment in urban environments involves modeling and algorithm design. The related works are introduced as follows.

### A. Optimization of RSUs Deployment

The growing demand for vehicle network connectivity has resulted in the widespread deployment of RSUs. The deployment scenarios have been categorized as static deployment [10] and dynamic deployment [11], [12] accordingly. For deployment modeling, it is common to encounter two types of objectives. The first one concerns communication profit, while the second addresses expenses. The profit-related factors encompass RSU network coverage [12], response time delay [13], [14], service profit [15], and so on. For the expenses issue, construction and maintenance costs for RSUs tend to be included [16].

First, researchers combine several optimization objectives into one target and use single-objective optimization algorithms to solve problem. The methods include mathematical programming, heuristic algorithms, data-driven methodologies, etc. The authors in [17] weight two factors, say time delay and cellular bandwidth cost, as one objective to model RSU deployment to build a nonlinear integer programming problem, then use a two-step centralized heuristic approach for solving the problem. In [10], researchers consider RSU coverage and the number of RSUs to build a single-objective optimization model and propose an improved genetic algorithm to solve the problem. In [18], an enhanced genetic technique is improved to address a single-objective optimization model by prefixing weights for network coverage, communication interference, and deployment costs, respectively. In current research of deployment modeling, the data-driven method is popular to examine real-world traffic flows to better reflect communication needs. In [19], the authors train a traffic forecasting

model using Google Maps data, providing accurate data for RSU deployment. The study in [2] emphasizes the importance of RSU deployment in VANETs and proposes a mechanism to prioritize investments based on real traffic features. The authors of [20] suggest a traffic data-based RSU deployment strategy to reduce costs and increase communication coverage using traffic data from Maghrebian cities. Additional research on real-world data is available in [14], [15], [21]. These algorithms anticipate RSU demand and QoS using historical data, traffic flow data, and vehicle trajectories to determine appropriate RSU deployment locations. However, in the above studies, the single-objective method cannot find a pareto set of solutions since each objective's weight is prefixed. Thus, the solutions rarely provide a full view of RSU deployment in practice. In addition, mathematical programming also takes a long time to solve optimization problems when confronted with large-scale variables.

Second, several studies have employed multi-objective optimization techniques for the purpose of deploying RSUs. However, it is worth noting that the research conducted in this particular area is very limited. In [22] and [23], the authors consider three objectives, namely RSU deployment cost, coverage area, and expected data transmission delay respectively, to propose a multi-objective optimization model, then employ NSGA-II related algorithms to address the problem. An improved multi-objective particle swarm optimization technique optimizes RSU deployment cost and communication coverage simultaneously [24]. Olia et al. optimize communication efficiency while minimizing RSU count by using NSGA-II [25]. Wang et al. propose a novel multi-objective differential evolutionary method that minimizes RSU count and maximizes communication coverage [8]. In the above studies, it has been observed that multi-objective optimization techniques have the capability to generate Pareto-optimal solutions. However, these methods often encounter difficulties when dealing with intricate constraints, such as those found in urban environments. Additionally, the optimization efficiency may be compromised, particularly when encountering a large-scale decision-variable space.

### B. Some Related Issues

1) *Constraints Handling:* Constraints are an unavoidable aspect of modeling the deployment of RSUs in urban conditions. Hence, the researchers in [13] and [5] incorporate budgetary limitations, geographical constraints, and road construction into their approaches for addressing the deployment of RSUs, which brings challenges in problem modeling and algorithm design. In order to overcome these constraints, [13] and [5] have put forth enhancements, namely the constrained Delaunay triangulation (CDT) approach and the enhanced greedy algorithm, respectively. By effectively considering and dealing with the constraints, the optimization strategies for the RSUs deployment will become more viable and realistic.

2) *Data Offloading Problem:* Many academics have developed techniques and strategies for data offloading in vehicular networks, which is influential on RSUs' efficiency. In [26]–[28], the authors focus on optimizing specific metrics such as

maximizing V2I system utility, minimizing offloading latency [29]–[31], or reducing system budgets [32]–[34]. However, these approaches often lean on static or predetermined models, which might not efficiently handle dynamic traffic flows. To address this issue, heuristic approaches combine multiple algorithms to provide flexible solutions, such as hybrid intelligent optimization algorithms [27] or load balancing schemes [35], [36]. However, it also brings challenges with large computational demands and low efficiency in complex scenarios. Besides, based on historical and real-time data, supervised learning and reinforcement learning approaches are also favored in optimizing data offloading strategies. Liu et al. present two reinforcement learning algorithms to optimize data offloading and resource allocation, taking into account vehicle traffic, and dynamic requests under different communication environments [37]. These algorithms are also computationally intensive, and their performances are dependent on the training procedures as well. To investigate the interactions among vehicles and RSUs, there also exist game theory methods to assign RSUs resources for vehicle data offloading [38], [39]. By formulating the RSU deployment as a game, the authors can analyze the strategic behavior of RSUs and derive optimal deployment strategies that maximize their individual benefits while considering the overall network performance.

### III. PROBLEM FORMULATION

In this section, we mainly take into account two aspects of the RSUs deployment. First, delay time is involved, which includes minimizing the total delay and maximizing the maximum delay in latency-sensitive areas. Second, to control the construction cost, minimizing the number of RSUs is also set as an optimization objective. In the modeling, the constraints consist of the areas covered by various kinds of obstacles, which are essential factors in real-world urban scenarios.

#### A. Objective Functions

In this paper, we assume there are two ways of transferring data. A vehicle prefers a connection to a RSU, while cellular networks are connected if the vehicle is out of RSUs' coverage.

The first goal is to decrease the overall latency delay that all vehicles experience at all time-periods because this helps improve the effectiveness of network communication as a whole. By minimizing the total latency time, it ensures that the network meets the requirements of as many vehicles as possible. The objective is given in (1).

$$\min \sum_{i=1}^V \sum_{t=1}^T d_i^t \quad (1)$$

where  $d$  is the total latency time,  $V$  is the number of vehicles,  $T$  is the number of time-periods,  $i$  and  $t$  are the corresponding indexes of vehicle and time-period, respectively. For  $d_i^t$ , it consists of several components and the details can be found in Appendix A.

In (1), the latency time is evaluated for the whole target region. However, there are several geographical areas that

are latency-sensitive, such as accident-prone areas, crossroads surrounding buildings, the central business district, and other similar locales, where the latency time should be strictly limited. Therefore, the second objective aims to minimize the maximum delay-time in latency-sensitive areas. By this means, we can ensure that the vehicular network satisfies the stringent requirements of latency-sensitive responses. The objective is given in (2).

$$\min \max_{i \perp A} \sum_{t=1}^T d_i^t \quad (2)$$

where  $d_i^t$  can be obtained as the same as in (1),  $i$  is the index of vehicle,  $A$  is the set of latency-sensitive areas,  $i \perp A$  means the vehicle  $i$  is located at the area  $A$ , and  $T$  is all time periods. Based on (2), it limits the maximum time-delay for the latency-sensitive areas, which further strengthens the communication efficiency in such areas.

Lastly, the third objective focuses on minimizing the quantity of deployed RSUs to control the investment costs of construction and further reduce maintenance expenses. The objective is described in (3).

$$\min \sum_{k=1}^K x_k \quad (3)$$

where  $K$  is the number of all potential positions,  $x_k = 1$  means the location with index  $k$  is positioned by a RSU, while  $x_k = 0$  otherwise.

#### B. Descriptions of Constraints

In urban areas, the constraints for RSU deployment are mainly from two aspects. The first kind of constraint is from the urban environment, which is related to obstacles such as buildings, gardens, rivers, and many others, where RSUs cannot be deployed within these obstructions. Therefore, RSUs must not be deployed in such locations. For the obstacles set  $O = \{o_1, o_2, \dots, o_K\}$ ,  $o_i = 1$  means there exists an obstacle at location  $i$ , while  $o_i = 0$  otherwise, where  $i \in \{1, \dots, K\}$  and  $K$  is the number of all potential positions. Meanwhile, the locations of RSUs are defined as a set  $R = \{r_1, r_2, \dots, r_K\}$ , where  $r_i = 1$  means there exists a RSU at location  $i$ , while  $r_i = 0$  otherwise. Then, the constraint can be expressed by an inequality shown in (4).

$$r_i \wedge o_i \neq 1, \quad i \in \{1, \dots, K\} \quad (4)$$

In (4),  $\wedge$  is the logic symbol “and”. It means that  $r_j$  and  $o_j$  cannot simultaneously be the value of 1. If a RSU is deployed inside an obstacle, the violation is calculated by the distance from the RSU to the nearest edge of the obstacle. Second, signal interference may occur when RSUs are deployed too closely, deteriorating communication quality and network performance [40]. Therefore, the distance limitation is implemented to prevent communication interference between RSUs. By presetting a minimum distance  $D_{min}$  between RSUs, it helps reduce interference risk and improve network stability. Therefore, the constraint is expressed in (5).

$$\text{dis}(r_i, r_j) \geq D_{min}, \quad \forall i, j \in \{1, \dots, K\}, j \neq i \quad (5)$$

where  $r_i = 1$  and  $r_j = 1$  are any two positions deploying RSUs,  $\text{dis}()$  is a function to calculate the distance between two positions,  $D_{min}$  is a preset minimum distance. For (5), it is used to control the density of RSUs. However, during the optimization process, this constraint on the RSU density may prevent algorithms from yielding feasible solutions. Therefore, in Section IV, two versions of algorithms are proposed. The one employs this RSUs-density constraint in calibrating candidate solutions during the whole optimization process, while the other one does not.

### C. Problem Statement

Based on the established optimization objectives and constraints, the RSUs deployment in urban environment can be stated as a multi-objective optimization problem with constraints, which is formulated in (6).

$$\min \left\{ \begin{array}{l} f_1 = \sum_{i=1}^V \sum_{t=1}^T d_i^t \\ f_2 = \max_{i \perp A} \sum_{t=1}^N d_i^t \\ f_3 = \sum_{k=1}^K x_k \end{array} \right. \quad (6)$$

subject to:

- (1)  $r_i \wedge o_i \neq 1, \quad i \in \{1, \dots, K\}$
- (2)  $\text{dis}(r_i, r_j) \geq D_{min}, \quad \forall i, j \in \{1, \dots, K\}, j \neq i$

Based on the (6), we have built a multi-objective optimization problem with constraints. To address this problem, the algorithms are designed in Section IV.

## IV. ADAPTIVE MULTI-POPULATION NSGA-III

According to the modeling in (6), the RSUs deployment is a multi-objective optimization with constraints. In addition, in the urban scenario, the map is gridded into hundreds of pieces, which makes this problem exhibit large-scale optimization properties. To simultaneously take into account the multiple objectives, an improved NSGA-III is considered in this paper. In the improvements, it is crucial to design a balancing strategy for exploitation and exploration because of the large scale of the decision-variable space. To be specific, the innovations are generally made in two aspects. First, we employ a multi-population strategy to enhance the exploration capability of the proposed algorithm. Meanwhile, adaptive parameter tuning for both mutation and crossover is proposed to dynamically balance exploitation and exploration. Second, to handle the constraints, an  $\epsilon$ -level comparison method cited from [41] is used to compare solutions even if they violate the constraints. In addition, to make RSUs retain a certain distance when generating offspring, a calibrating strategy is also proposed in this paper. All the details are illustrated in the following subsections.

### A. Rules for Comparing Solutions

For multi-objective optimization algorithms, the rules for comparing solutions are crucial to determine the selection

mechanism. In our design, we compare solutions in three cases. First, for solutions  $\mathbb{A}$  and  $\mathbb{B}$ , if  $\mathbb{A}$  satisfies the constraints, but  $\mathbb{B}$  does not, then  $\mathbb{A}$  is better than  $\mathbb{B}$ . Second, if  $\mathbb{A}$  and  $\mathbb{B}$  both satisfy the constraints, then pareto-dominance rule is preferred in comparing the two solutions [42]. Third, if neither  $\mathbb{A}$  nor  $\mathbb{B}$  satisfies the constraints, we employ an improved  $\epsilon$ -level rule to compare the two solutions [41]. The details about the  $\epsilon$ -level rule are explained as follows. For two solutions,  $\mathbb{A}$  and  $\mathbb{B}$ , their overall constraint violations are defined as  $\phi^A$  and  $\phi^B$  respectively. Then, for any  $\epsilon \geq 0$ , the epsilon level comparison  $\preceq_\epsilon$  is defined in (7).

$$(\mathbb{A}, \phi^A) \preceq_\epsilon (\mathbb{B}, \phi^B) \leftrightarrow \begin{cases} \mathbb{A} \preceq \mathbb{B}, & \text{if } \phi^A, \phi^B \leq \epsilon \\ \mathbb{A} \preceq \mathbb{B}, & \text{if } \phi^A = \phi^B \\ \phi^A < \phi^B, & \text{otherwise} \end{cases} \quad (7)$$

where  $\mathbb{A} \preceq \mathbb{B}$  means that  $\mathbb{A}$  dominates  $\mathbb{B}$ . The symbol “ $\preceq_\epsilon$ ” means the comparison is conducted with a  $\epsilon$ -level relaxation of constraints. The detail explanations on  $\epsilon$ -level rule are presented in Appendix B.

### B. Strategies in Balancing Exploitation and Exploration

In the urban environment, the regions are partitioned into hundreds of grids for RSUs deployment, which brings a multitude of variables and therefore forms a large-scale optimization issue. To address such problems, it is important to emphasize the algorithm's capability to balance exploration and exploitation. In this paper, we employ a multi-population strategy so that the whole population is divided into  $N$  sub-populations. In each sub-population, the algorithm runs independently. Among sub-populations, solutions migrate after each generation. In this way, superior solutions in one sub-population will replace inferior solutions in other sub-populations. This migration strategy helps improve the diversity of solutions in each population and, meanwhile, accelerates convergence by sharing superior solutions among sub-populations. The pseudo-codes of the migration operator are given in Algorithm 1.

---

#### Algorithm 1 Migration Operator among Sub-populations

---

##### Require:

- Total amount of sub-populations  $m$
- Sub-populations  $SP_1, SP_2, \dots, SP_m$
- Amount of emigrants  $N_{EM}$  in each sub-population

##### Ensure:

- Updated  $SP_1, SP_2, \dots, SP_m$
- 1: **for**  $i = 1$  to  $m$  **do**
  - 2:     Select the best  $N_{EM}$  individuals in  $SP_i$
  - 3:     Store the  $N_{EM}$  individuals in  $D_i$
  - 4: **end for**
  - 5: **for**  $i = 1$  to  $m$  **do**
  - 6:      $EM_i = \text{NULL}$
  - 7:     Merge all  $D_j$  ( $j \in \{1, \dots, m\} \& j \neq i$ ) into  $EM_i$
  - 8:     In  $SP_i$ , select the worst  $(m - 1) \times N_{EM}$  individuals and mark them as  $IM_i$
  - 9:     In  $SP_i$ , instead  $IM_i$  by  $EM_i$
  - 10: **end for**
  - 11: **return** Updated  $SP_1, SP_2, \dots, SP_m$
-

Besides the multi-population strategy, in the proposed algorithm, during iterations, an adaptive strategy in both crossover and mutation parameter tuning is applied. According to each sub-population's performance in the current generation, mutation and crossover probabilities are adjusted dynamically so as to balance exploration and exploitation. If the current best fitness in a sub-population has not been improved for a predefined number of iterations, the mutation probability increases while the crossover probability decreases. In this way, exploration is promoted. Conversely, if the current best fitness improves after the same predefined number of iterations, the mutation probability decreases and the crossover probability increases. Then exploitation is emphasized. The pseudo-codes are provided in Algorithm 2. It is noteworthy that due to the independent execution of the algorithm in each sub-population, there may be variations in the crossover rate and mutation rate across distinct sub-populations.

#### Algorithm 2 Adaptive Crossover Rate and Mutation Rate

##### Require:

- A sub-population  $SP$
- The iteration index  $g$ , where  $g > 1$
- Initial crossover rate  $C_r(g - 1)$  and mutation rate  $M_r(g - 1)$
- Value limitations of crossover rate  $C_{max}$ ,  $C_{min}$ , and limitations of mutation rate  $M_{max}$  and  $M_{min}$
- Range-ability for changing crossover rate  $\Delta_c > 0$  and for mutation rate  $\Delta_m > 0$

##### Ensure:

- Updated  $C_r(g)$  and  $M_r(g)$

- 1: Evaluate the current sub-population  $P$
- 2: Mark the current best fitness as  $F(g)$
- 3: **if**  $F(g)$  is better than  $F(g - 1)$  **then**
- 4:    $C_r(g) = C_r(g - 1) + \Delta_c$
- 5:    $M_r(g) = M_r(g - 1) - \Delta_m$
- 6: **else**
- 7:    $C_r(g) = C_r(g - 1) - \Delta_c$
- 8:    $M_r(g) = M_r(g - 1) + \Delta_m$
- 9: **end if**
- 10: Regulate the values of  $C_r(g)$  and  $M_r(g)$  by limitations
- 11: **return**  $C_r(g)$ ,  $M_r(g)$

#### C. Offspring Calibrating Strategy

In this problem, the locations of RSUs cannot be too close in order to prevent interference in data transmission. Therefore, an inhibition distance is preset as  $D_{min}$ . The offspring naturally generated by multi-objective optimization algorithms may not satisfy the constraints, and it should be calibrated after generating offspring. The steps are described as follows.

- 1) Calculate the distance between all RSU pairs.
- 2) Identify the RSU pairs that violate the inhibition distance constraint  $D_{min}$  according to (5).
- 3) For each violating RSU pair, rank the RSUs according to the traffic volume in the range of RSU's coverage.

- 4) Retain the RSU with higher traffic volume and remove the deployment of the other RSU.

This calibrating strategy guarantees that the retaining RSUs satisfy the inhibition distance constraint and therefore avoid communication interference. Meanwhile, the RSU retention mechanism in this strategy also helps maintain large-scale coverage for traffic volume.

#### D. Data Offloading Strategy

In transportation networks, vehicles offload data to RSUs and request services from RSUs [43]. When vehicles have more than one RSU to be connected, the data offloading strategy is much more influential in evaluating network performance, e.g., the calculation of latency time [13]. In this section, we propose a novel strategy named iterative best response sequential game (IBRSG). For  $V$  vehicles and  $R$  RSUs, the strategy for the  $i^{th}$  vehicle is marked as  $s_i$ . The value of  $s_i$  is an index value of a certain RSU. Then the establishment of data offloading can be described as follows.

- 1) Establish a random strategy  $S_{old} = \{s_{o1}, s_{o2}, \dots, s_{oV}\}$ , where  $s_{oi} \in \{1, \dots, R\}$ ,  $i \in \{1, \dots, V\}$
- 2) Set up a **NULL** strategy  $S_{new} = \{s_{n1}, s_{n2}, \dots, s_{nV}\}$ , where  $s_{ni} = \text{NULL}$ ,  $i \in \{1, \dots, V\}$
- 3) Initialize an index value **index** = 1
- 4) For all vehicles, their index is  $i \in \{1, \dots, V\}$ , update their connection strategy based on the following rule:  
For  $i > \text{index}$ , the  $i^{th}$  vehicle uses the strategy  $s_{oi}$ ;  
For  $i < \text{index}$ , the  $i^{th}$  vehicle uses the strategy  $s_{ni}$ ;  
For  $i$  equals to **index**, the  $i^{th}$  vehicle is connected to a RSU, which can minimize the total latency time for all  $V$  vehicles. Store the  $i^{th}$  vehicle's strategy to  $s_{ni}$ .
- 5) If **index** <  $V$ , then **index** = **index**+1 and goto "Step 4"; Otherwise, update  $S_{new} = \{s_{n1}, s_{n2}, \dots, s_{nV}\}$
- 6) If  $\|S_{old} - S_{new}\| \geq \text{error}$ , instead  $S_{old}$  by  $S_{new}$ , then goto "Step 3"; Otherwise, stop the iterations and output  $S_{new}$

In Step 6, **error** is a predefined threshold to evaluate the difference between two data offloading strategies. The whole loop runs until there is no big change between two iterative strategies.  $S_{new}$  is the final output strategy for data offloading. The use of a sequential strategy based on iterative best responses ensures the convergence and robustness of the algorithm. Since different ways to implement data offloading will affect the transmission efficiency and the load balance of RSUs, we also compare the proposed strategy with several other data offloading strategies in Section V.

#### E. Proposed Algorithm

According to the above designs, we propose adaptive multi-population NSGA-III, which has two versions. For the first version, it is with offspring calibrating strategy (marked by AM-NSGA-III-c), while the other version (named as AM-NSGA-III) is not. The pseudo-codes of AM-NSGA-III-c are shown in Algorithm 3. For the version of AM-NSGA-III, its pseudo-codes just delete the part of the offspring calibrating strategy that is marked in Line 7 of Algorithm 3. As mentioned

in Section IV-C, the offspring calibrating strategy is used to mitigate the RSU density by reducing the number of RSUs during algorithm executions. For the version without the offspring calibrating strategy, the number of RSUs may increase, potentially resulting in violations due to the overcrowded RSUs deployment. The violations persist consistently and are passed down from iteration to iteration, ultimately undermining the algorithm's ability to achieve feasible solutions. In addition, the number of RSUs obtained by AM-NSGA-III may be greater than that obtained by AM-NSGA-III-c. In the subsequent experiments, we will investigate the influence of offspring calibration on the algorithm's performance in detail.

---

**Algorithm 3** AM-NSGA-III-c

---

**Require:**

- Objective functions  $f_1$ ,  $f_2$  and  $f_3$
- Obstacle locations  $O_c$ , Distance constraint  $D_{min}$
- Population size  $N$
- Amount of sub-populations  $m$
- Initial crossover rate  $Cr$ , initial mutation rate  $Mr$ , initial violation tolerance  $\epsilon$
- Maximum number of iterations  $G$

**Ensure:**

- A set of pareto solutions **Pareto**

**1: Initialization:**

- 1) Randomly initialize a population  $P$  with size of  $N$ .
- 2) Partition  $P$  into  $m$  sub-populations  $SP_i$ , where  $i \in \{1, \dots, m\}$ , the size of each sub-population is  $\frac{N}{m}$
- 3) In each sub-population  $SP_i$ , the initial crossover rate  $c_{ri} = Cr$ , mutation rate  $m_{ri} = Mr$ , violation tolerance  $\epsilon_i = \epsilon$

**2: Main Loop:**
**3: for**  $g = 1$  to  $G$  **do**
**4: for**  $i = 1$  to  $m$  **do**
**5:** Evaluate the fitness of  $SP_i$  based on the objectives  $f_1$ ,  $f_2$ ,  $f_3$ , where the IBRSG data offloading strategy is employed in the evaluations.

**6:** Run NSGA-III for each sub-population  $SP_i$  to get offspring

**7:** Calibrate offspring according to  $D_{min}$ , where the rules are given in Section IV-C.

**8:** Rank and select solutions to update  $SP_i$  based on (7)

**9:** Update  $c_{ri}$  and  $m_{ri}$  according to Algorithm 2

**10:** Update  $\epsilon_i$  according to (12)

**11: end for**
**12:** Migrate solutions among sub-populations according to Algorithm 1.

**13: end for**
**14:** Collect all  $SP_i$  to update  $P$ 
**15:** Select non-dominated solutions from  $P$  to compose the pareto solutions **Pareto**
**16: return** **Pareto**


---

## V. EXPERIMENTS AND ANALYSIS

In this section, experiments are conducted to investigate the proposed algorithms' performance. The public datasets of

vehicle transportation in Chengdu City, China, are obtained from the “DiDi Gaia Open Data Program”. The data includes time stamps, coordinates, original locations, and destination locations. According to the data, we construct the traffic volumes by calling the Gaode Map API to perform route planning so that urban traffic scenarios can be simulated. In the urban environment, we select areas with high-density and low-density scenarios, respectively, which are explained in Appendix C.

### A. Parameter setting

In this subsection, two categories of parameters are introduced. The first kind of parameters are about scenarios, while the second kind of parameters are set for algorithms. The parameters on scenarios are presented in Appendix C. For the parameters in algorithms, we summarize them as follows. The population size ( $N$ ) is set as 360 and the number of sub-populations ( $m$ ) is set as 3. For (12), the initial value of parameters are set as follows:  $\theta = 0.05 \times N = 18$ ,  $\alpha = 0.95$  and  $\tau = 0.1$ . The initial crossover rate is set as 50%, while the mutation rate is set as 0.05. The crossover rate ranges from 20% to 100%, and the mutation rate varies between 0 and 0.1. The range-ability for crossover rate and mutation rate are set as  $\Delta_c = 0.1$  and  $\Delta_m = 0.01$ , respectively. For the migration operator, each sub-population will select the top 10% best solutions to emigrate, while the worst 20% ( $10\% \times (m - 1)$ ) solutions will be instead by immigrants. The maximum number  $G$  for algorithm's iteration is limited to 50.

### B. Evaluations of Algorithms Performance

In this subsection, we conduct experiments of AM-NSGA-III and AM-NSGA-III-c respectively. Their performances are used to compare with MOEA/D [44] and NSGA-III [45]. The pareto-solutions obtained by each algorithm is given in Fig. 1, where the red points are the reference point. By this figure, it means the four algorithms can be implemented to the RSUs deployment problem to obtain pareto solutions.

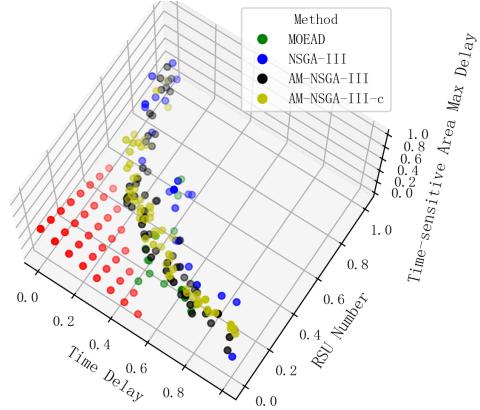


Fig. 1. Pareto fronts obtained by four algorithms

To obtain a straightforward comparison, we present Table I with performance metrics including the number of Pareto-optimal solutions (NPS), the number of feasible solutions (NFS), the Inverted Generational Distance (IGD) indicator, the Hypervolume (HV) indicator and the S-Metric indicator.

TABLE I  
PERFORMANCE COMPARISON OF ALGORITHMS

Metric	MOEA/D	NSGA-III	AM-NSGA-III	AM-NSGA-III-c
NPS	14	26	<b>59</b>	58
NFS	0	4	15	<b>22</b>
IGD	0.39171	0.36199	0.25565	<b>0.24437</b>
HV	2.10352	3.03098	11.60189	<b>12.07229</b>
S-Metric	<b>0.0242</b>	0.0623	0.0302	0.0296

1) *Performances in High Density Scenarios:* From Table I, it can be observed that both AM-NSGA-III and AM-NSGA-III-c achieve a high score for NPS and NFS, which significantly outperforms MOEA/D and NSGA-III. Moreover, according to NFS, AM-NSGA-III-c dramatically performs the best. In addition, it is worth noting that MOEA/D does not generate any feasible solutions.

Upon analysis of the IGD indicator and careful examination of the pareto front depicted in Figure 1, it becomes apparent that the pareto front produced by the AM-NSGA-III-c algorithm exhibits the highest proximity to the reference points, which serves as a clear indication of the superiority of AM-NSGA-III-c's performance. In addition, the HV indicator provides further evidence for this result, indicating that AM-NSGA-III-c has superior performance compared to other algorithms. The S-Metric indicator reveals that MOEA/D achieves the most favorable outcome, while the AM-NSGA-III-c algorithm performs the second best. However, considering the absence of feasible solutions obtained by MOEA/D, it can be concluded that the AM-NSGA-III-c demonstrates superior overall performance for high-density scenarios in the comparisons.

The reasons behind this performance can be attributed to the improved exploration and exploitation capabilities of the AM-NSGA-III-c algorithm, which enable it to effectively navigate the search space and discover more feasible and high-quality solutions that satisfy the given constraints. Additionally, the adaptive mechanism for calibrating crossover and mutation rates helps balance exploration and exploitation, further contributing to performance. Moreover, by incorporating the calibrating offspring strategy, AM-NSGA-III-c can effectively balance the trade-off between satisfying constraints and optimizing multiple objectives. As a result, the algorithm generates high-quality and more feasible solutions, which are crucial for the real-world implementation of RSU deployments.

All the feasible solutions are presented in Appendix D. According to Table I, since MOEA/D fails to obtain any feasible solution, we only list the results obtained by NSGA-III, AM-NSGA-III, and AM-NSGA-III-c, which are shown in Table A.II. The three objectives in this table are explained in (6). It is noted that **Objective 1** is the total latency time, **Objective 2** is the maximal delay in the latency-sensitive areas, and **Objective 3** is the amount of RSUs. According to Table A.II, we know that besides MOEA/D, the algorithm NSGA-III obtained the fewest number of feasible solutions.

To evaluate the three algorithms, we use the dominance rule to compare all the feasible solutions and obtain Table A.III. According to Table A.III, it can be observed that AM-

NSGA-III-c has the largest number of pareto front solutions, followed by AM-NSGA-III, while all feasible solutions from NSGA-III were eliminated, which demonstrates that the two versions of the proposed algorithms outperform both MOEA/D and NSGA-III. In addition, in Table A.III, compared with AM-NSGA-III-c, solutions obtained by AM-NSGA-III usually have a higher number of RSUs. The reason is that, during the algorithm's iteration, AM-NSGA-III does not employ the offspring calibrating strategy. Therefore, in the AM-NSGA-III algorithm's run, violations about minimal distance between RSUs usually exist in generating offspring. However, the same violations will cause RSU deletion in the AM-NSGA-III-c algorithm, and therefore the number of RSUs in AM-NSGA-III-c will be less. On the other hand, the latency time in AM-NSGA-III is generally less than that in AM-NSGA-III-c. Based on the analysis, we know that if decision makers prefer the construction cost, AM-NSGA-III-c is advisable. On the contrary, if decision makers would like to reduce latency time, AM-NSGA-III is appropriate.

To visualize the situations of RSU deployment, we show four feasible solutions of each algorithm to draw the diagrams, which are given in Fig.A.2, Fig. A.3 and Fig. A.4. In the figures, we mark the latency-sensitive areas with cyan diamonds. The dark blue points are the locations for RSU deployment. In such areas, the maximal latency time should be minimized to pursue fast responses. However, for NSGA-III, it does not result in an overcrowded RSU deployment but also fails to deploy RSUs in narrow traffic hotspot areas. For example, in Fig. A.2, many RSUs are located very closely for the four deployment strategies. On the other hand, for the four diagrams in Fig. A.2, especially in the bottom right corner, there are many narrow roads with traffic hotspots, but very few RSUs are deployed at the areas. For AM-NSGA-III, shown in Fig. A.3, although some RSUs are still deployed closely, the narrow areas with traffic hotspots have several RSUs deployed. For AM-NSGA-III-c, shown in Fig. A.4, the deployed RSUs are relatively uniformly distributed. Meanwhile, the deployment strategy also gives consideration to narrow areas. To sum up, compared with other algorithms, in a high-density environment, AM-NSGA-III-c is competent to obtain enough feasible optimal deployment strategies aiming at the objectives in (6).

2) *Performances in Low Density Scenarios:* In urban environment, both high-density and low-density areas exist. In this subsection, we investigate the RSUs deployment in low-density scenarios. In the performance comparisons, we also employ MOEA/D, NSGA-III, AM-NSGA-III, and AM-NSGA-III-c to investigate the RSUs deployment. The metrics

TABLE II  
PERFORMANCE COMPARISON OF ALGORITHMS IN LOW DENSITY

Metric	MOEA/D	NSGA-III	AM-NSGA-III	AM-NSGA-III-c
NPS	22	<b>24</b>	23	19
NFS	6	16	<b>18</b>	17
IGD	0.413803	0.41570	0.36539	<b>0.19456</b>
HV	3.58877	4.09742	2.65386	<b>2.2750</b>
S-Metric	0.06197	0.08181	<b>0.03986</b>	0.05879

include NPS, NFS, IGD, HV and S-Metric, while the results are provided in Table II.

Based on the data shown in Table II, the quantities of both NPS and NFS achieved by MOEA/D exhibit an increment when compared to its performance in high-density scenarios. Nevertheless, the most significant performance on NFS is still achieved by the AM-NSGA-III and AM-NSGA-III-c algorithms. The distribution of pareto solutions obtained by the proposed algorithms is satisfactory in terms of the metrics of IGD, HV, and S-Metric. This result further supports the competence of the proposed algorithms in handling low-density scenarios.

All feasible solutions are presented in Appendix E. The solutions of the algorithms to the three optimization objectives are listed in Table A.IV. In this table, we find that the number of feasible solutions in NSGA-III increases compared with the case in high-density scenarios, which means that the low-density area is easier for algorithms to handle. On the other hand, in the solutions of AM-NSGA-III-c, the number of RSUs is less than that in other algorithms. The reason is the same as when dealing with high-density scenarios. Correspondingly, the latency time in AM-NSGA-III-c is still larger than that in other algorithms. To analyze all algorithms' performance, we use the dominance rule to compare the feasible solutions and obtain a pareto set summarized in Table A.V. In Table A.V, we will find that all feasible solutions obtained by MOEA/D are dominated by other algorithms, which means the quality of MOEA/D's solutions is not competitive. In addition, AM-NSGA-III-c (with 10 solutions) still has more solutions than AM-NSGA-III (with 8 solutions) and NSGA-III( with 3 solutions). This indicates that, in low-density scenarios, AM-NSGA-III-c can still generate more feasible solutions than the other algorithms. The number of feasible solutions in NSGA-III is the lowest, which means that NSGA-III's ability to pursue enough feasible solutions is still incompetent. For AM-NSGA-III, the number of feasible solutions is evenly matched with the number in AM-NSGA-III-c, which means that the constraints in low-density obstacle areas are relatively loose. The number of RSUs in AM-NSGA-III ranges from 19 to 29. The corresponding latency time will neither be too large nor too small. Although the number of RSUs in AM-NSGA-III-c is relatively small, the burden of RSUs will be heavy and therefore cause a high delay in communications. Based on the analysis, the solutions proposed by AM-NSGA-III will be appropriate choices. However, if the decision makers give priority to construction cost in their considerations, AM-NSGA-III-c is preferred. Meanwhile, in time-sensitive areas, the solutions of AM-NSGA-III-c also achieve a relatively

low delay. The reason for this is that AM-NSGA-III-c does not deploy RSUs too densely, which avoids the decrement in utilization rate of RSUs and waste of communication resources caused by overcrowded RSU deployment. With a small number of RSUs, AM-NSGA-III-c is competent to cover a large area and provide communication services for more vehicles.

To provide visualization comparisons, we also select four results obtained by each algorithm to draw diagrams, which are given in Fig. A.5, Fig. A.6 and Fig. A.7 for NSGA-III, AM-NSGA-III, and NSGA-III-c, respectively. In the figures, there are two latency sensitive areas marked by cyan diamonds, while the dark-blue points are the RSU deployment locations. Based on the deployment locations, it is obvious that the solutions provided by all algorithms can generally cover the traffic flow. In Fig. A.5 and Fig. A.6, the number of RSUs is relatively higher than that in Fig. A.7 and therefore obtains a relatively low time delay according to Table A.IV. Nevertheless, in Fig. A.5 and Fig. A.6, several RSUs are deployed too closely. In Fig. A.7, the distance between RSUs is uniformly located in general, and the number of RSUs is relatively low, although each RSU will afford relatively heavy burdens in communications.

3) *Analysis of Algorithms with Increased Delay Sensitive Areas:* We conducted a comparative study of the algorithms MOEA/D, NSGA-III, AM-NSGA-III, and AM-NSGA-III-c in the same high-density scenario, where the number of delay-sensitive areas increased to 6 and 10, respectively. For the case with 6 latency-sensitive regions, the results are given in Table III. The table reveals that neither MOEA/D nor NSGA-III can yield feasible solutions, indicating that an increment in the number of latency-sensitive areas poses a greater challenge for algorithmic resolution. Our proposed algorithms continue to demonstrate proficiency in achieving feasible solutions. Correspondingly, the pareto solutions of AM-NSGA-III and AM-NSGA-III-c can be found in Table III, which showcases the performance of the proposed algorithms. In addition, we include four diagrams illustrating the spatial arrangements in Appendix F. We also select four figures for each algorithm to demonstrate the spatial deployment, shown in Fig. A.8 and Fig. A.9 for AM-NSGA-III and AM-NSGA-III-c, respectively.

For the case with 10 latency-sensitive regions, among the algorithms, only AM-NSGA-III-c obtains feasible solutions shown in Table IV, while MOEA/D, NSGA-III, and AM-NSGA-III failed to find any feasible solutions. The results clearly indicate the superiority of AM-NSGA-III-c in handling multi latency-sensitive areas. In Appendix F, the spatial

TABLE III  
THE PERFORMANCE OF FEASIBLE SOLUTIONS OF AM-NSGA-III AND AM-NSGA-III-C ON THREE OBJECTIVES.

Algorithm	Objective 1	Objective 2	Objective 3
AM-NSGA-III	225631.26	0.845629	35
	232364.41	0.93568	30
	284269.73	0.98531	27
	346548.37	0.94433	20
AM-NSGA-III-c	200765.36	0.79352	34
	226954.83	0.73525	29
	254693.25	0.80863	25
	286536.79	0.80239	22
	302658.57	0.80161	20
	306598.74	0.83094	19
	334569.42	0.85365	17

deployment of AM-NSGA-III-c for this case are shown in Fig. A.10. Since there only obtains 5 feasible solutions, we show all of them in the figure. According to the above analysis, when

TABLE IV  
THE PERFORMANCE OF FEASIBLE SOLUTIONS OF AM-NSGA-III-C ON THREE OBJECTIVES WITH 10 DELAY SENSITIVE AREAS.

Algorithm	Objective 1	Objective 2	Objective 3
AM-NSGA-III-c	347348.6	1.06849	13
	283862.3	0.92926	19
	280072.1	0.82802	22
	271401.7	0.87875	24
	227543.7	0.80335	27

evaluating algorithms' performance across different numbers of latency sensitive areas, a trend was observed: the number of feasible solutions decreases as the number of latency sensitive areas increases. This is due to the increased complexity and restrictions brought about by the greater number of latency sensitive areas.

### C. Evaluations on Data Offloading Strategy Performance

To verify the effectiveness of the proposed data offloading strategy, we compare IBRSG with five other different data offloading strategies in terms of total delay for all vehicles and the load of all RSUs. The experimental scenario and related parameter settings are based on the case of the high-density urban environment. The illustrations for the offloading strategies are presented as follows.

- 1) IBRSG data offloading strategy: The proposed strategy in this paper.
- 2) Location-based data offloading strategy (marked by “Mindis”): Offloading vehicle tasks to the nearest RSU [46].
- 3) Signal strength-based data offloading strategy (marked by “MinPL”): Offloading vehicle tasks to the RSU with the strongest current signal strength [47].
- 4) Random data offloading strategy (“marked by Random”): Offloading vehicle tasks to a random RSU [48].
- 5) Genetic algorithm-based data offloading strategy (“marked by GA”): Allocating vehicle offloading data to an RSU according to a genetic algorithm [49].

- 6) Multi-metric criteria-based data offloading strategy (“marked by MCDM”): Offloading vehicle tasks to the most optimal RSU based on a weighted combination of multiple metrics [13].

The metrics to compare the six data offloading strategies include total latency-time, network load balance, and convergence speed, which are explained in Appendix G. The comparison results are given as follows.

1) *Comparison of Total Latency*: The latency comparison of data offloading options over iterations is illustrated in Fig. 2. The IBRSG improves with more iterations, closing the gap with GA and nearing GA. The GA approach initially had an advantage, but the IBRSG eventually caught up. The consistently higher performance of the IBRSG technique suggests it could be a dependable and efficient vehicular network data offloading strategy.

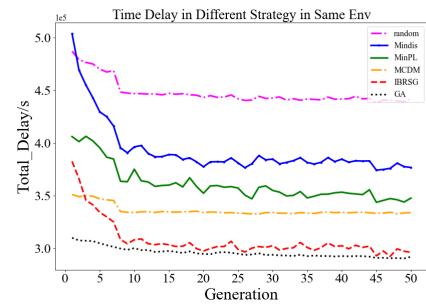


Fig. 2. Time delay in different strategy in same environment.

2) *Comparison of Load Balance*: Fig.3 reflects a load balance comparison of data offloading options. The MCDM method regularly balances load better than the others, which optimizes load balance best, followed by the IBRSG strategy. The IBRSG technique outperforms the other four.

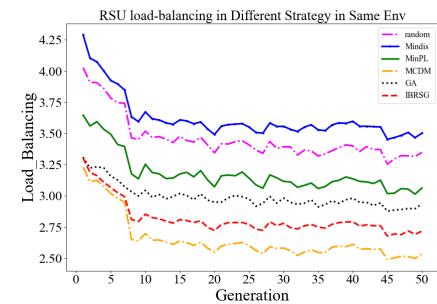


Fig. 3. Load balancing in different strategy in same environment.

3) *Comparison of Time Consumption*: The comparison of time consumption is shown in Fig. 4, where the values are marked by logarithmic forms,  $\log_{10} \text{Time\_consumption}$ , for an intuitive comparison (this is the reason for the negative values in the figure). Although the random technique has the lowest calculation overhead, its performances on both total latency and load balance deteriorate. The IBRSG strategy computes the second fast.

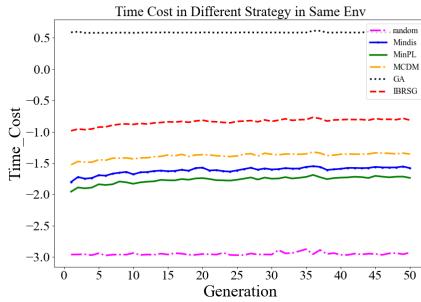


Fig. 4. Time cost in different strategy in same environment.

4) *Summary of the Data Offloading Strategies:* In summary, the IBRSO strategy offers a well-balanced performance across key metrics. It is consistently better at minimizing delays and balancing loads than most other strategies, which shows that it has the potential to be a reliable and efficient data offloading strategy. This shows that the IBRSO strategy is a better choice for real-world use in vehicular networks.

## VI. CONCLUSIONS AND FUTURE WORK

This research examines the difficulties associated with the multi-objective optimal deployment of roadside units in urban vehicular networks, with the aims of enhancing communication efficiency and reducing the deployment cost of RSUs. This study proposes two versions of multi-objective optimization algorithms that build on the improvements made to NSGA-III. These algorithms are meant to deal with the problems of conflicting optimization objectives, obstacles in urban environments, and exploring a large-scale optimization space. The algorithms proposed in this study incorporate a multi-population method, an adaptive exploration technique, and an offspring calibration mechanism within the NSGA-III framework. These components are utilized to modify the distribution and density of RSUs. A unique data offloading mechanism is also proposed by designing an iterative best response-based sequential game (IBRSO) approach. The experimental performances of the proposed algorithms are compared and analyzed in both high-density and low-density urban scenarios. A comparison of results is made with state-of-the-art algorithms, and the results demonstrate the superiority and feasibility of the proposed algorithms. In future work, we will explore the integration of other emerging technologies, such as edge computing and machine learning, to further enhance the performance of vehicular networks in urban environments.

## ACKNOWLEDGMENTS

Thanks for Mr. Wanli CAI for his helps in providing experiment environments to run and debug the algorithms. This work also get guidance and comments from Professor Lei WANG in Tongji University, Shanghai, China.

## REFERENCES

- [1] Donghyun Kim, Yesenia Velasco, Wei Wang, RN Uma, Rasheed Hussain, and Sejin Lee. A new comprehensive rsu installation strategy for cost-efficient vanet deployment. *IEEE Transactions on Vehicular Technology*, 66(5):4200–4211, 2016.
- [2] H. Yu, R. Liu, Z. Li, Y. Ren, and H. Jiang. An rsu deployment strategy based on traffic demand in vehicular ad hoc networks (vanets). *IEEE Internet of Things Journal*, 9(9):6496–6505, 2021.
- [3] Manuel Fogue, Julio A Sanguesa, Francisco J Martinez, and Johann M Marquez-Barja. Improving roadside unit deployment in vehicular networks by exploiting genetic algorithms. *Applied Sciences*, 8(1):86, 2018.
- [4] Lixia Xue, Yuchen Yang, and Decun Dong. Roadside infrastructure planning scheme for the urban vehicular networks. *Transportation Research Procedia*, 25:1380–1396, 2017.
- [5] Moyukh Laha and Raja Datta. A budgeted maximum coverage based mmwave enabled 5g rsus placement in urban vehicular networks. In *2021 International Conference on COMmunication Systems & NETworks (COMSNETS)*, pages 387–395. IEEE, 2021.
- [6] Zhilong Li, Jing Liu, and Qi Zhang. A ga-based strategy for deploying cable connected roadside units in vanets. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pages 1–5. IEEE, 2018.
- [7] Jianping Liu, Lei Cai, Xinke Wang, Chenyang Lyu, and Li Zhang. Roadside unit deployment of cooperative vehicle-infrastructure system based on digital measurable image method. In *Journal of Physics: Conference Series*, volume 1626, page 012112. IOP Publishing, 2020.
- [8] Feng Wang, Chenle Wang, Kan Wang, Qiaoyong Jiang, Bin Wang, and Wenjuan He. Multiobjective differential evolution with discrete elite guide in internet of vehicles roadside unit deployment. *Wireless Communications and Mobile Computing*, 2021:1–13, 2021.
- [9] Gunasekaran Raja, Aishwarya Ganapathisubramaniyan, Sudha Anbalagan, Sheeba Backia Marry Baskaran, Kathiroli Raja, and Ali Kashif Bashir. Intelligent reward-based data offloading in next-generation vehicular networks. *IEEE Internet of Things Journal*, 7(5):3747–3758, 2020.
- [10] Abderrahim Guerna and Salim Bitam. Gica: An evolutionary strategy for roadside units deployment in vehicular networks. In *2019 International Conference on Networking and Advanced Systems (ICNAS)*, pages 1–6. IEEE, 2019.
- [11] Ribao Cai, Yijia Feng, Dazhi He, Yin Xu, Yu Zhang, and Wei Xie. A combined cable-connected rsu and uav-assisted rsu deployment strategy in v2i communication. In *ICC 2020-2020 IEEE international conference on communications (ICC)*, pages 1–6. IEEE, 2020.
- [12] Ribao Cai, Yijia Feng, Dazhi He, Yin Xu, Yu Zhang, and Wei Xie. Trajectory optimization for large-scale uav-assisted rsus in v2i communication. In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pages 1–6. IEEE, 2020.
- [13] Chinmoy Ghora and Indrajit Banerjee. A constrained delaunay triangulation based rsus deployment strategy to cover a convex region with obstacles for maximizing communications probability between v2i. *Vehicular Communications*, 13:89–103, 2018.
- [14] Sudha Anbalagan, Ali Kashif Bashir, Gunasekaran Raja, Priyanka Dhanasekaran, Geetha Vijayaraghavan, Usman Tariq, and Mohsen Guizani. Machine-learning-based efficient and secure rsu placement mechanism for software-defined-iov. *IEEE Internet of Things Journal*, 8(18):13950–13957, 2021.
- [15] Yuanzhi Ni, Jianping He, Lin Cai, Jianping Pan, and Yuming Bo. Joint roadside unit deployment and service task assignment for internet of vehicles (iov). *IEEE Internet of Things Journal*, 6(2):3271–3283, 2018.
- [16] Abderrahim Guerna, Salim Bitam, and Carlos T Calafate. Ac-rdv: a novel ant colony system for roadside units deployment in vehicular ad hoc networks. *Peer-to-Peer Networking and Applications*, 14:627–643, 2021.
- [17] Weijing Qi, Björn Landfeldt, Qingyang Song, Lei Guo, and Abbas Jamalipour. Traffic differentiated clustering routing in dsrc and c-v2x hybrid vehicular networks. *IEEE Transactions on Vehicular Technology*, 69(7):7723–7734, 2020.
- [18] Yunyi Liang, Ning Ma, Xin Li, and Jia Hu. Stochastic roadside unit location optimization for information propagation in the internet of vehicles. *IEEE Internet of Things Journal*, 8(17):13316–13327, 2021.
- [19] Sreya Ghosh, Iti Saha Misra, and Tamal Chakraborty. Optimal rsu deployment using complex network analysis for traffic prediction in vanet. *Peer-to-Peer Networking and Applications*, 16(2):1135–1154, 2023.
- [20] Seif Ben Chaabene, Taoufik Yeferny, and Sadok Ben Yahia. A roadside unit deployment framework for enhancing transportation services in maghrebian cities. *Concurrency and Computation: Practice and Experience*, 33(1):e5611, 2021.
- [21] Yuhao Liu, Zhibin Chen, Siyuan Gong, and Han Liu. Reducing congestion and emissions via roadside unit deployment under mixed traffic flow. *International Journal of Coal Science & Technology*, 10(1):1, 2023.

- [22] Liangjie Yu, Zihui Zhang, Jiajian Li, Jing Ma, and Yong Wang. A multi-objective roadside unit deployment model for an urban vehicular ad hoc network. *ISPRS International Journal of Geo-Information*, 12(7):262, 2023.
- [23] Zihui Zhang, Chao Li, Liangjie Yu, Yiqiang Zhao, and Yanqiang Li. A multi-objective roadside units deployment method in vanet. In *2021 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pages 390–394. IEEE, 2021.
- [24] Lingyu Zhang, Li Wang, Lili Zhang, Xiao Zhang, and Dehui Sun. An rsu deployment scheme for vehicle-infrastructure cooperated autonomous driving. *Sustainability*, 15(4):3847, 2023.
- [25] Arash Olia, Hossam Abdelgawad, Baher Abdulhai, and Saiedeh N Razavi. Optimizing the number and locations of freeway roadside equipment units for travel time estimation in a connected vehicle environment. *Journal of Intelligent Transportation Systems*, 21(4):296–309, 2017.
- [26] Shichao Li, Siyu Lin, Lin Cai, Wenjie Li, and Gang Zhu. Joint resource allocation and computation offloading with time-varying fading channel in vehicular edge computing. *IEEE Transactions on Vehicular Technology*, 69(3):3384–3398, 2020.
- [27] Jianan Sun, Qing Gu, Tao Zheng, Ping Dong, Alvin Valera, and Yajuan Qin. Joint optimization of computation offloading and task scheduling in vehicular edge computing networks. *Ieee Access*, 8:10466–10477, 2020.
- [28] Chung-Ming Huang and Chi-Feng Lai. The delay-constrained and network-situation-aware v2v2i vanet data offloading based on the multi-access edge computing (mec) architecture. *IEEE Open Journal of Vehicular Technology*, 1:331–347, 2020.
- [29] Liping Qian, Yuan Wu, Ningning Yu, Fuli Jiang, Haibo Zhou, and Tony QS Quek. Learning driven noma assisted vehicular edge computing via underlay spectrum sharing. *IEEE Transactions on Vehicular Technology*, 70(1):977–992, 2021.
- [30] Zhenyu Zhou, Pengju Liu, Junhao Feng, Yan Zhang, Shahid Mumtaz, and Jonathan Rodriguez. Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach. *IEEE Transactions on Vehicular Technology*, 68(4):3113–3125, 2019.
- [31] Pengju Liu, Junluo Li, and Zhongwei Sun. Matching-based task offloading for vehicular edge computing. *IEEE Access*, 7:27628–27640, 2019.
- [32] Nan Cheng, Feng Lyu, Wei Quan, Conghao Zhou, Hongli He, Weisen Shi, and Xuemin Shen. Space/aerial-assisted computing offloading for iot applications: A learning-based approach. *IEEE Journal on Selected Areas in Communications*, 37(5):1117–1129, 2019.
- [33] Chao Yang, Yi Liu, Xin Chen, Weifeng Zhong, and Shengli Xie. Efficient mobility-aware task offloading for vehicular edge computing networks. *IEEE Access*, 7:26652–26664, 2019.
- [34] Xin Li, Yifan Dang, Mohammad Aazam, Xia Peng, Tefang Chen, and Chunyang Chen. Energy-efficient computation offloading in vehicular edge cloud computing. *IEEE Access*, 8:37632–37644, 2020.
- [35] Jie Zhang, Hongzhi Guo, Jiajia Liu, and Yanning Zhang. Task offloading in vehicular edge computing networks: A load-balancing solution. *IEEE Transactions on Vehicular Technology*, 69(2):2092–2104, 2019.
- [36] Shih-Yang Lin, Chung-Ming Huang, and Tzu-Yu Wu. Multi-access edge computing-based vehicle-vehicle-rsu data offloading over the multi-rsu-overlapped environment. *IEEE Open Journal of Intelligent Transportation Systems*, 3:7–32, 2022.
- [37] Yi Liu, Huimin Yu, Shengli Xie, and Yan Zhang. Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks. *IEEE Transactions on Vehicular Technology*, 68(11):11158–11168, 2019.
- [38] Ke Zhang, Yuming Mao, Supeng Leng, Sabita Maharjan, and Yan Zhang. Optimal delay constrained offloading for vehicular edge computing networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2017.
- [39] Xiaolong Xu, Qinting Jiang, Peiming Zhang, Xuefei Cao, Mohammad R Khosravi, Linss T Alex, Liyanpong Qi, and Wanchun Dou. Game theory for distributed iov task offloading with fuzzy neural network in edge computing. *IEEE Transactions on Fuzzy Systems*, 30(11):4593–4604, 2022.
- [40] Ming Mao, Peng Yi, Zhen Zhang, Liang Wang, and Jinchuan Pei. Roadside unit deployment mechanism based on node popularity. *Mobile Information Systems*, 2021:1–11, 2021.
- [41] Zhun Fan, Wenji Li, Xinye Cai, Han Huang, Yi Fang, Yugen You, Jiajie Mo, Caimin Wei, and Erik Goodman. An improved epsilon constraint-handling method in moea/d for cmops with large infeasible regions. *Soft Computing*, 23:12491–12510, 2019.
- [42] Dongyang Li, Lei Wang, Li Li, Weian Guo, Qidi Wu, and Alexander Lerch. A large-scale multiobjective particle swarm optimizer with enhanced balance of convergence and diversity. *IEEE Transactions on Cybernetics*, 2022.
- [43] Yasir Saleem, Nathalie Mitton, and Valeria Loscri. Divine: Data offloading in vehicular networks with qos provisioning. *Ad Hoc Networks*, 123:102665, 2021.
- [44] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.
- [45] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4):577–601, 2013.
- [46] M Carmen Lucas-Estañ and Javier Gozálvez. Mode selection for multi-hop cellular networks with mobile relays. In *2013 IFIP Wireless Days (WD)*, pages 1–6. IEEE, 2013.
- [47] Zhuo Yi, Xuehui Du, Ying Liao, and Lifeng Cao. A quality-of-service-aware dynamic evolution model for space-ground integrated network. *International Journal of Distributed Sensor Networks*, 13(8):155014771728649, 2017.
- [48] Chang Guo, Demin Li, Xuemin Chen, and Guanglin Zhang. An adaptive v2r communication strategy based on data delivery delay estimation in vanets. *Vehicular Communications*, 34:100444, 2022.
- [49] Keshav Jha, Akhil Gupta, Abdulatif Alabdulatif, Sudeep Tanwar, Calin Ovidiu Safirescu, and Traian Candin Mihaltan. Csvag: Optimizing vertical handoff using hybrid cuckoo search and genetic algorithm-based approaches. *Sustainability*, 14(14):8547, 2022.
- [50] Xiaolong Xu, Bowen Shen, Xiaochun Yin, Mohammad R Khosravi, Huaming Wu, Liyanpong Qi, and Shaohua Wan. Edge server quantification and placement for offloading social media services in industrial cognitive iov. *IEEE Transactions on Industrial Informatics*, 17(4):2910–2918, 2020.
- [51] Chen Chen, Jinna Hu, Tie Qiu, Mohammed Atiquzzaman, and Zhiyuan Ren. Cvcg: Cooperative v2v-aided transmission scheme based on coalitional game for popular content distribution in vehicular ad-hoc networks. *IEEE Transactions on Mobile Computing*, 18(12):2811–2828, 2018.
- [52] Yujie Tang, Nan Cheng, Wen Wu, Miao Wang, Yanpeng Dai, and Xuemin Shen. Delay-minimization routing for heterogeneous vanets with machine learning based mobility prediction. *IEEE Transactions on Vehicular Technology*, 68(4):3967–3979, 2019.
- [53] Banoth Ravi, Anmol Gautam, and Jaisingh Thangaraj. Stochastic performance modeling and analysis of multi service provisioning with software defined vehicular networks. *AEU-International Journal of Electronics and Communications*, 124:153327, 2020.
- [54] Tsenka Stoyanova, Fotis Kerasiotis, Aggeliki Prayati, and George Papadopoulos. A practical rf propagation model for wireless network sensors. In *2009 Third International Conference on Sensor Technologies and Applications*, pages 194–199. IEEE, 2009.



**Weian Guo** received the M.Eng. degree in navigation, guidance, and control from Northeastern University, Shenyang, China, in 2009, and the doctor of engineering degree from Tongji University, Shanghai, China, in 2014. From 2011 to 2013, he was sponsored by China Scholarship Council to carry on his research at the Social Robotics Laboratory, National University of Singapore. He is currently an associate Professor with the Sino-German College of Applied Science, Tongji University. His interests include computational intelligence, optimization, artificial intelligence, and control theory.



**Zecheng Kang** received the B.S. degree in automotive engineering from Tongji University, Shanghai, China, in 2021, and he is currently pursuing a Master's degree in intelligent science and technology at Tongji University. His research interests are focused on heuristic algorithms, deep learning, and their applications in intelligent transportation system.



**Dongyang Li** received the M.S. and Ph.D degrees in school of electronics and information engineering, Tongji University, Shanghai, China, in 2017 and 2022, respectively. From 2019 to 2021, he was sponsored by China Scholarship Council to carry on his research at Georgia Institute of Technology. He is now an engineer with the Sino-German College of Applied Science, Tongji University; His research interest includes computational intelligence, deep learning and their applications.



**Lun Zhang** received the B.S. and Ph.D degrees in computer communications, transportation information engineering and control, Tongji University, Shanghai, China, in 1992 and 2005, respectively. He is currently a professor with School of Transportation, Tongji University. His interests include intelligent transportation, computational intelligence, and deep learning.



**Li Li** received the B.S. and M.S. degrees in electrical automation from Shengyang Agriculture University, Shengyang, China, in 1996 and 1999, respectively, and the Ph.D. degree in mechatronics engineering from the Shenyang Institute of Automation, Chinese Academy of Science, Shenyang, in 2003.

She joined Tongji University, Shanghai, China, in 2003, where she is currently a professor of control science and engineering. She has over 50 publications, including 4 books, over 30 journal papers, and 2 book chapters. Her current research interests include production planning and scheduling, computational intelligence, data-driven modeling and optimization, semiconductor manufacturing, and energy systems

## APPENDIX

This is the appendix for the manuscript of “Multi-objective Optimal Roadside Units Deployment in Urban Vehicular Networks”.

### A. Explanations on time delay

Eqs. (8) to (11) are complementary to Sect. III-A for computing the commutation delay.

In general, the communication delay consists of transmission delay, propagation delay, and queuing delay [50]. However, in this paper, by gridding an urban map, the communication range is typically less than a few hundred meters, which does not exceed the RSU’s communication capability. Therefore, the propagation delay is negligible. Correspondingly, the total delay in this paper is expressed in (8).

$$d_{ij}^t = d_{ij}^{t,trans} + d_{ij}^{t,queue} \quad (8)$$

where  $d_{ij}^{t,trans}$  is the transmission delay, while  $d_{ij}^{t,queue}$  is the queuing delay. It is worth mentioning that from  $d_{ij}^t$  to  $d_i^t$ , the setting of  $j$  is based on the data offloading strategy, which should be designed in the communication protocol [43]. In vehicular networks, vehicles offload data to RSUs and request services from RSUs. When vehicles have a chance to connect to more than one RSU, the design of data offloading strategy is much more influential in evaluating network performance, e.g. the calculation of latency time [13]. The details in designing data offloading strategy will be presented in Section IV.

In (8),  $d_{ij}^{t,trans}$  expressed in (9) is affected by packet size  $L_p$  and transmission rate  $Tr_{ij}^t$ .

$$d_{ij}^{t,trans} = \frac{L_p}{Tr_{ij}^t} \quad (9)$$

where  $L_p$  is the size of data packet,  $Tr_{ij}^t$  is the transmission rate. In this paper, we set  $L_p$  as a constant, which means the data is divided into the same length for transmission [51]. The way to calculate  $R_{ij}^t$  is obtained from [52], which is expressed in (10).

$$Tr_{ij}^t = B \log_2 \left( 1 + \frac{P_t}{(L_{fs} + L_{shad} + N_0) B} \right) \quad (10)$$

where  $B$  is the bandwidth,  $P_t$  is the transmission power,  $L_{fs}$  is the free-space path loss,  $L_{shad}$  is the loss caused by shadows,  $N_0$  is the noise power spectrum density. In (8), for the queuing delay  $d_{ij}^{t,queue}$  expressed in (11), it is obtained under a M/M/1 queuing model [53].

$$d_{ij}^{t,queue} = \frac{1}{\mu_j - \lambda_j^t} \quad (11)$$

where  $i$  and  $j$  are the indexes of vehicle and RSU respectively,  $t$  is the index of time-period,  $\mu_j$  is the service rate of RSU  $j$ ,  $\lambda_j^t$  is the arrival rate of packets at RSU  $j$  at time period  $t$ .

### B. $\epsilon$ -level rules

The  $\epsilon$ -level rules is cited from [41], which is used for comparing two solutions with violations. The value of  $\epsilon$  varies with iterations, which can be obtained by (12).

$$\epsilon(g) = \begin{cases} \text{rule1} : \phi_{\sum \theta}, \text{ if } g = 0 \\ \text{rule2} : (1 - \tau)\epsilon(g - 1), \text{ if } \rho_g < \alpha \text{ and } g < G \\ \text{rule3} : (1 + \tau)\phi_{max}, \text{ if } \rho_g \geq \alpha \text{ and } g < G \\ \text{rule4} : 0, \text{ if } g \geq G \end{cases} \quad (12)$$

where  $g$  is the index of iterations,  $G$  is the maximum limitation of iterations,  $\phi_{\sum \theta}$  is the violation sum of top- $\theta$  ranking individuals in the initial population ( $g = 0$ ),  $\phi_{max}$  is the maximum overall constraint violation found by far,  $\tau \in [0, 1]$  plays the role to reduce the constraints relaxation and to control the scale factor multiplied by maximum overall constraint violation.  $\rho_g$  is the ratio of feasible solutions at the  $g^{th}$  iteration.  $\alpha \in [0, 1]$  is to control the searching preference between the feasible and infeasible regions.

Based on (12), it is feasible to compare two solutions with constraints violations. Especially when  $g < G$ , the value of  $\alpha$  can be used to control the tolerance to violations. If  $\alpha$  is a large value, **rule2** tends to happen. In this case,  $\epsilon(g)$  will be smaller than  $\epsilon(g - 1)$ , which means the violation tolerance goes strict. On the contrary, it is probable to select **rule3** if  $\alpha$  is a small value. In this case, violation tolerance will be promoted. Therefore, it is crucial to design a suitable value of  $\alpha$  in the comparisons.

### C. Scenarios and Parameters

The high-density and low-density scenarios are given in Fig. A.1. In the high-density scenario, the utilization of Tian-Fu Square is observed, where RSUs cater to a total of 56,704 vehicles. Conversely, in the low-density scenario, the service provision near Du Fu’s thatched cottage accommodates 36,405 vehicles. In the figures, the gray part means the obstacles, and the red part denotes the traffic heat marked by vehicles’ frequency. The traffic volume is high when the red color is dark, while the light red color means the traffic volume is low. For each scenario, it covers one square kilometer and is uniformly divided into  $50 \times 50$  map pieces, thereby creating 2500 decision variables in the experiments.

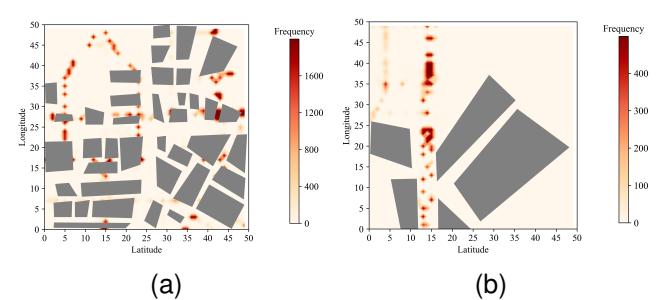


Fig. A.1. (a) Scenarios for high-density area. (b) Scenario for low-density area.

For the parameters in urban scenarios, we present Table A.I to calculate (9), (10), and (11). In this table, the free-space path

TABLE A.I  
SIMULATION PARAMETER SET-UP

Parameter	Value
Packet size( $L_p$ )	1Mb
Bandwidth ( $B$ )	10 MHz
Transmitting power of RSU ( $P_t$ )	23 dBm
Noise power spectrum density ( $N_0$ )	-174 dBm
Path loss model $L_{fs}$	Free-space path loss model
Shadowing model $L_{shad}$	Log-normal shadowing model
RSU service rate ( $\mu$ )	20
Arrive rate ( $\lambda$ )	Number of arrived vehicles

loss mode  $L_{fs}$  and the log-normal shadowing model  $L_{shad}$  are introduced by (13) and (14) respectively, which are cited from [54] and [13].  $L_{fs}$  is used to access signal attenuation when no obstacle exists between vehicles and RSUs, while  $L_{shad}$  is used to evaluate the signal attenuation when obstacles exist between vehicles and RSUs. In addition, in the table, for the arrive rate  $\lambda$ , it is obtained according to the real traffic data.

$$L_{fs} = 20 \log_{10}(dis) + 20 \log_{10}(f_{signal}) - 147.55 \quad (13)$$

where  $dis$  is the distance between vehicle and RSU by meters, which is calculated by the real traffic data.  $f_{signal}$  is the frequency of the signal, which is set as 5.9GHz in this paper.

$$L_{shad} = 10 \cdot \sigma \cdot \mathcal{N}(0, 1) \quad (14)$$

where the  $\sigma$  is the standard deviation of the shadowing attenuation, which is set as 4dB in this paper,  $\mathcal{N}(0, 1)$  represents a normal distribution. For other parameters in scenarios, we summarize them as follows. In the communication network described in this paper, vehicles transfer or receive data every 30 seconds. The radius of a latency-sensitive area is 20 meters. For the constraints of RSU density, the minimal distance  $D_{min}$  between RSUs is set as 30 meters. In addition, if a vehicle connects to a cellular network, the time delay is set as a constant of 2 seconds<sup>1</sup>.

#### D. Experimental Results for the High-density Scenario

In high-density scenario, the feasible solutions obtained by NSGA-III, AM-NSGA-III, and AM-NSGA-III-c are presented in Table A.II. Based on the data in such table, we compare all the feasible solutions to compose a pareto front shown in Table A.III. For each algorithm, we present four diagrams to show the RSU deployment in Fig. A.2, Fig. A.3 and Fig. A.4.

#### E. Experimental Results for the Low-density Scenario

In low-density scenario, the feasible solutions obtained by NSGA-III, AM-NSGA-III, and AM-NSGA-III-c are presented in Table A.IV. Based on the data in such table, we compare all the feasible solutions to compose a pareto front shown in Table A.V. For each algorithm, we present four diagrams to show the RSU deployment in Fig. A.5, Fig. A.6 and Fig. A.7.

<sup>1</sup>Since the packet size is 8Mb and the transferring speed of the cellular network in an urban environment is about 4Mbps, the latency of transmitting data through the cellular network is set to a constant value of 2 seconds.

TABLE A.II  
THE FEASIBLE SOLUTIONS OBTAINED BY NSGA-III, AM-NSGA-III AND AM-NSGA-III-c FOR THE HIGH-DENSITY SCENARIO

Algorithms	Objective 1	Objective 2	Objective 3
NSGA-III	268975.4	0.74705	52
	270625.3	0.78494	47
	292602.1	0.82984	43
	347545.7	0.84349	24
AM-NSGA-III	205928.2	0.76515	44
	223522.9	0.74558	41
	224745.4	0.76874	39
	235735.0	0.74113	39
	241672.7	0.78226	36
	262040.6	0.82293	33
	267599.3	0.79763	31
	269848.4	0.81164	30
	284045.1	0.80584	30
	288118.1	0.80405	30
AM-NSGA-III-c	288685.4	0.82349	29
	289546.3	0.81212	28
	290385.5	0.82069	27
	293218.1	0.82359	24
	318575.6	0.85887	22
	202880.4	0.80161	34
	204962.1	0.78154	34
	207677.2	0.79127	31
	209613.3	0.81132	30
	212552.1	0.80095	30
	215207.2	0.82127	29
	222515.8	0.84418	28
	225287.5	0.81077	28
	230424.5	0.84525	27
	236999.8	0.83394	27
	237892.9	0.84922	26
	243671.7	0.82374	26
	245787.9	0.81008	26
	269620.9	0.80388	25
	275152.2	0.79682	25
	278087.6	0.82612	24
	278601.3	0.82032	24
	283443.7	0.82831	22
	285710.3	0.81678	20
	286750.5	0.81344	20
	288313.5	0.85788	18
	289497.0	0.84867	18

#### F. Experimental Results for Increased Number of Latency-sensitive Areas

The number of latency-sensitive areas may greatly influence the practical implementation of RSU deployment. To investigate the influences, it will be helpful to explore algorithms' capabilities in dealing with complex urban environments. In this experiments, the principle to select the areas is based on the intersections of roads. Two situations are conducted in this subsection, which are with 6 and 10 latency-sensitive areas. For the 6 latency-sensitive areas, results are given in Fig. A.8 and Fig. A.9 for AM-NSGA-III and AM-NSGA-III-c, respectively. For the 10 latency-sensitive areas, results are presented in Fig. A.10.

#### G. Comparison results for Different Data Offloading Strategies

To evaluate the performance of different offloading strategies, the metrics are presented as follows.

- 1) Total delay: The sum of the delays experienced by all vehicles across all time periods based on (1).

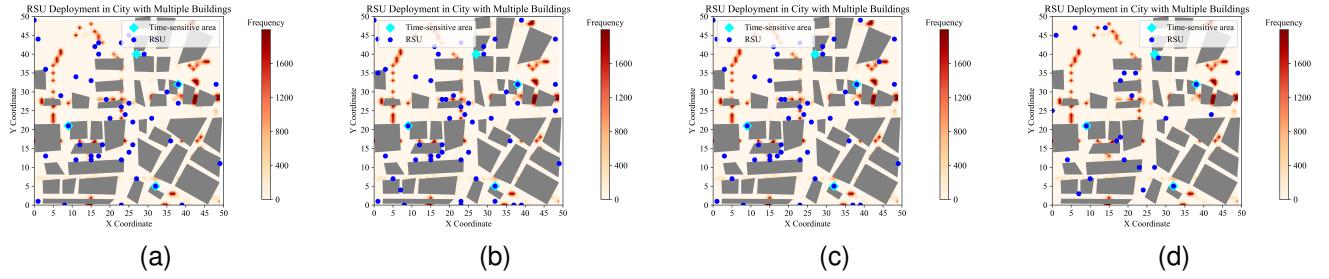


Fig. A.2. RSUs deployment by NSGA-III in high-density scenario.

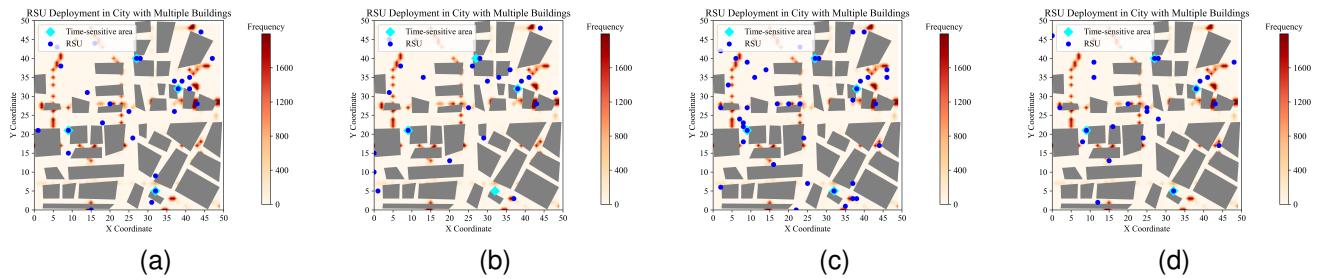


Fig. A.3. RSUs deployment by AM-NSGA-III in high-density scenario.

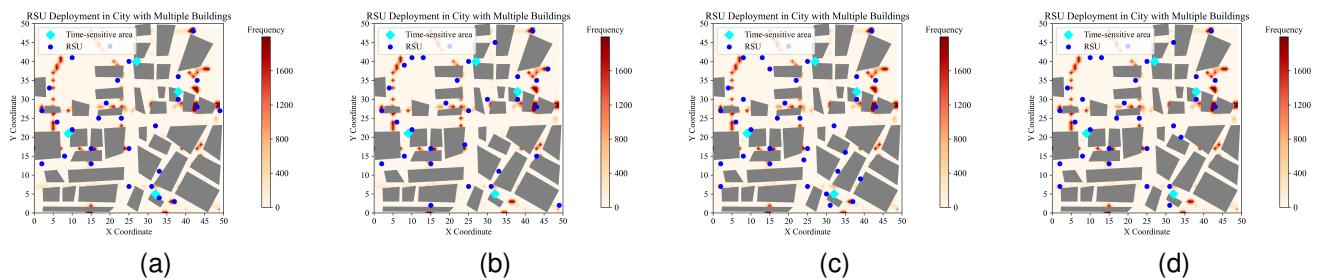


Fig. A.4. RSUs deployment by AM-NSGA-III-c in high-density scenario.

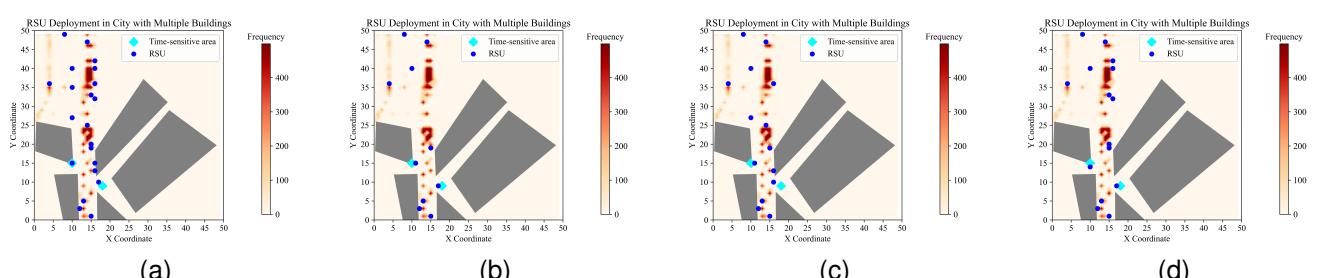


Fig. A.5. NSGA-III deployment in low-density scenario.

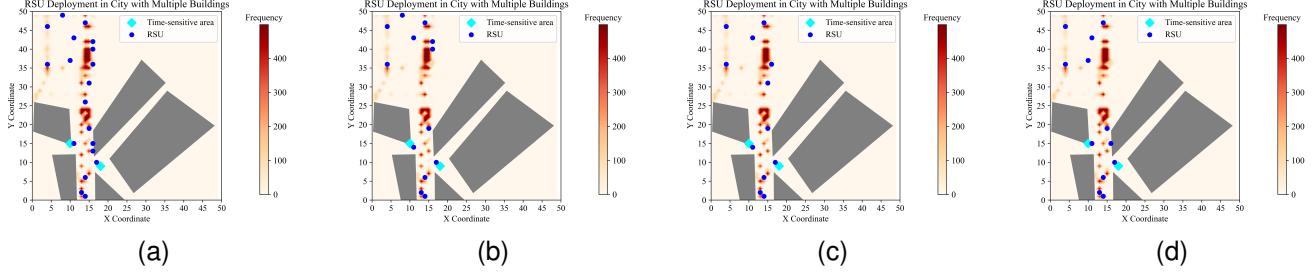


Fig. A.6. AM-NSGA-III deployment in low-density scenario.

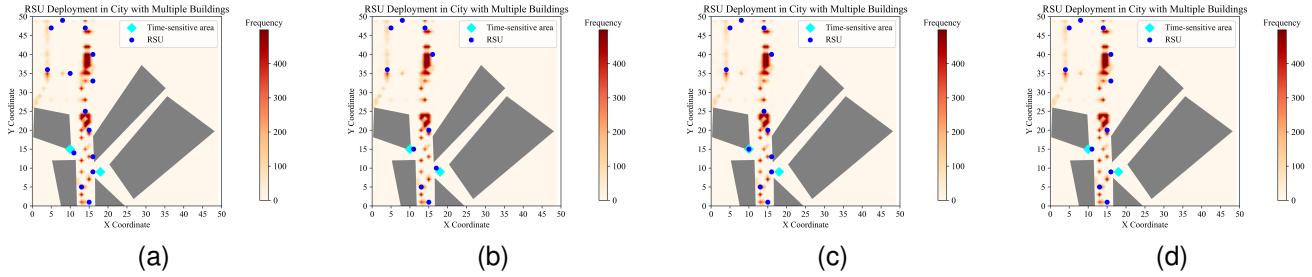


Fig. A.7. AM-NSGA-III-c deployment in low-density scenario.

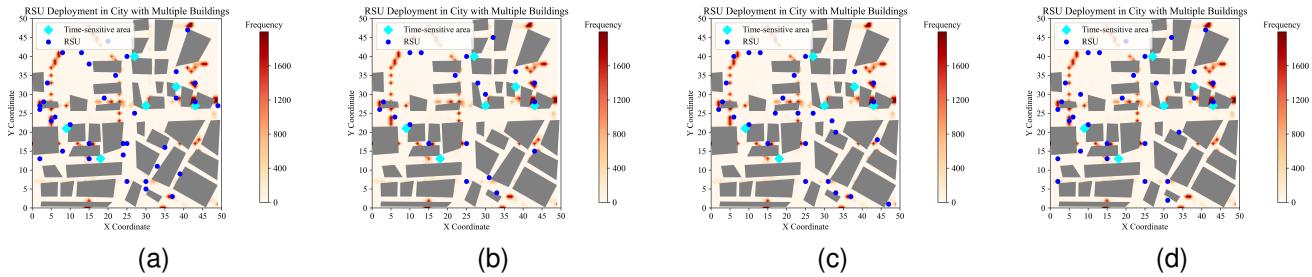


Fig. A.8. AM-NSGA-III deployment of with 6 latency-sensitive areas.

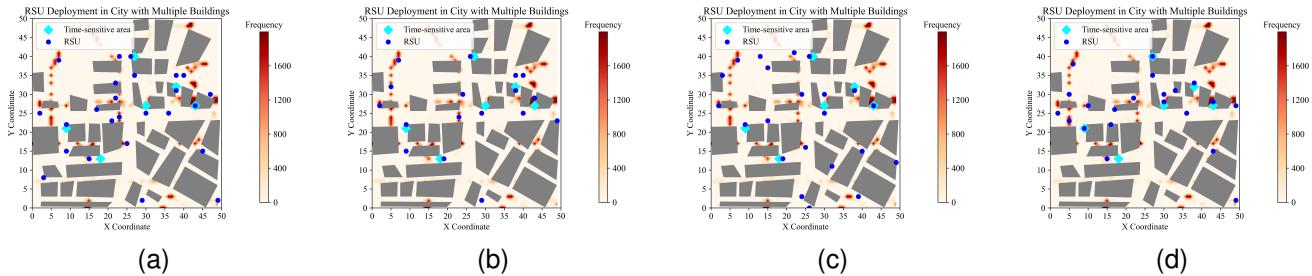


Fig. A.9. AM-NSGA-III-c deployment of with 6 latency-sensitive areas.

- 2) The load balance: The load balance ensures that the vehicles' offloading requests are distributed evenly among the available RSUs. This helps prevent overloading a certain number of RSUs and promotes RSU utilization, which is defined in (15).

$$\text{Balancing} = \sqrt{\frac{\sum_{i=1}^R (\text{Load}_i - \text{Load}_{\text{mean}})^2}{R}} \quad (15)$$

where  $\text{Load}_{\text{mean}}$  is the average load across all RSUs and can be calculated by  $\text{Load}_{\text{mean}} = \frac{\sum_{i=1}^M \text{Load}_i}{M}$ ,  $\text{Load}_i$  represents the load (number of connected vehicles) on the  $i$ -th RSU within a time period, and  $R$  denotes the total number of RSUs in the network.

- 3) Time consumption: It is evaluated by the time consumption for a strategy to converge. This metric indicates the efficiency of different data offloading strategies.

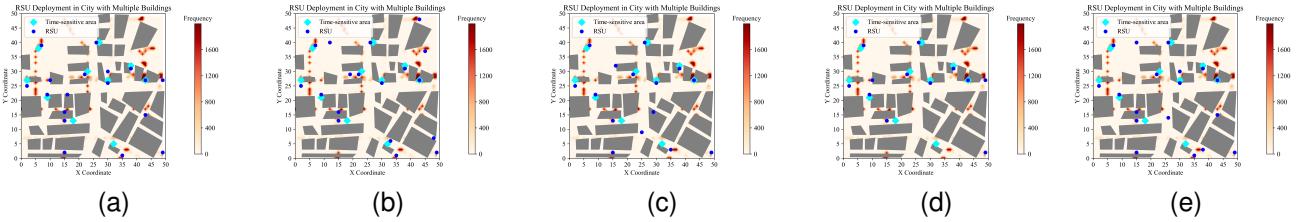


Fig. A.10. AM-NSGA-III-c deployment of with 10 latency-sensitive areas.

TABLE A.III

PARETO FRONT BY COMPARING ALL FEASIBLE SOLUTIONS OBTAINED BY  
NSGA-III, AM-NSGA-III AND AM-NSGA-III-c FOR THE  
HIGH-DENSITY SCENARIO.

<b>Algorithms</b>	<b>Objective 1</b>	<b>Objective 2</b>	<b>Objective 3</b>
AM-NSGA-III	205928.2	0.76515	44
	223522.9	0.74558	41
	224745.4	0.76874	39
	235735.0	0.74113	39
	241672.7	0.78226	36
	267599.3	0.79763	31
AM-NSGA-III-c	202880.4	0.80161	34
	204962.1	0.78154	34
	207677.2	0.79127	31
	209613.3	0.80132	30
	212552.1	0.81095	30
	215207.2	0.82127	29
	222515.8	0.84418	28
	225287.5	0.81077	28
	230424.5	0.83525	27
	236999.8	0.83394	27
	237892.9	0.80922	26
	243671.7	0.84374	26
	245787.9	0.81008	26
	269620.9	0.80388	25
	275152.2	0.79682	25
	278087.6	0.82612	24
	278601.3	0.82032	24
	283443.7	0.82831	22
	285710.3	0.81678	20
	286750.5	0.81344	20
	288313.5	0.84788	18
	289497.0	0.84867	18

TABLE A.IV  
THE FEASIBLE SOLUTIONS OBTAINED BY NSGA-III, AM-NSGA-III AND  
AM-NSGA-III-C FOR THE LOW-DENSITY SCENARIO.

<b>Algorithms</b>	<b>Objective 1</b>	<b>Objective 2</b>	<b>Objective 3</b>
MOEA/D	12854.36	0.829952	17
	13772.94	0.80525	15
	13869.27	0.79663	13
	14804.98	0.81433	11
	14693.35	0.854056	11
	15560.63	0.815891	10
NSGA-III	10616.77	0.749952	33
	10772.94	0.73525	29
	10911.69	0.768727	25
	11369.27	0.77663	24
	11474.81	0.702563	24
	11604.98	0.74433	23
	11704.99	0.701207	22
	11893.35	0.754056	21
	12560.63	0.795891	19
	12665.17	0.717512	18
	12788.85	0.730365	18
	12965.76	0.726708	17
	13128.92	0.72856	17
	13881.26	0.71809	14
AM-NSGA-III	13990.79	0.796187	14
	14136.02	0.720669	13
	10243.95	0.781843	29
	10661.78	0.724627	28
	10880.98	0.776299	27
	10985.55	0.786457	25
	11129.62	0.758313	24
	11246.17	0.712753	22
	11367.18	0.782226	21
	11899.37	0.74847	19
	12133.94	0.719733	19
	116428.2	0.712334	18
	12735.53	0.768231	18
	12941.74	0.798218	18
AM-NSGA-III-c	13054.87	0.748149	16
	13222.71	0.764303	15
	13340.65	0.736728	14
	13551.23	0.730376	13
	13721.36	0.725468	12
	14580.23	0.785957	10
	11042.11	0.749952	19
	11300.03	0.73525	18
	11434.38	0.768727	18
	11575.73	0.77663	16
	11706.61	0.702563	14
	11830.42	0.701207	14
	11944.71	0.754056	14
	12077.8	0.717512	14
AM-NSGA-III-c	12178.16	0.730365	12
	12310.18	0.726708	12
	12441.02	0.72856	11
	12645.51	0.71809	11
	12765.64	0.796187	11
	12959.33	0.78192	11
	13120.02	0.782896	10
	13306.02	0.784019	10
	13475.85	0.759641	9

TABLE A.V  
PARETO FRONT BY COMPARING ALL FEASIBLE SOLUTIONS OBTAINED BY  
NSGA-III, AM-NSGA-III AND AM-NSGA-III-C FOR THE  
LOW-DENSITY SCENARIO.

<b>Algorithms</b>	<b>Objective 1</b>	<b>Objective 2</b>	<b>Objective 3</b>
NSGA-III	10911.69	0.768727	25
	11474.81	0.702563	24
	11704.99	0.701207	22
AM-NSGA-III	10243.95	0.781843	29
	10661.78	0.724627	28
	10880.98	0.776299	27
	10985.55	0.786457	25
	11129.62	0.758313	24
	11246.17	0.712753	22
	11367.18	0.782226	21
AM-NSGA-III-c	11899.37	0.74847	19
	11042.11	0.749952	19
	11300.03	0.73525	18
	11575.73	0.77663	16
	11830.42	0.701207	14
	12178.16	0.730365	12
	12310.18	0.726708	12
	12441.02	0.72856	11
	12645.51	0.71809	11
	13120.02	0.782896	10