mathematics

Article

Resource- and Time-Efficient Computation Offloading in Vehicular Edge Computing: A Max-Min Fairness Oriented Approach

Shujuan Wang [1], Hao Peng [1] and Dongchao Guo [2,*]

[1] School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China
[2] School of Computer Science, Beijing Information Science and Technology University, Beijing 100192, China
* Correspondence: dongchaoguo@bistu.edu.cn

Abstract: Nowadays, computation offloading has become a research focus since it has the potential to solve the challenges faced when dealing with computation-intensive applications in the Internet of Vehicles (IoVs), especially in the 5G or future network environment. However, major issues still exist and the performance of main metrics can be improved to better adapt to the practical scenarios. This paper focuses on achieving resource- and time-efficient computation offloading in IoVs by boosting the cooperation efficiency of vehicles. Firstly, a fuzzy logic-based pricing strategy is designed to evaluate the cooperation tendency and capability of each vehicle from multiple aspects. Vehicles are encouraged to participate in the offloading process even if they are in a disadvantageous position compared to other vehicles. Secondly, a Max-Min fairness-oriented approach is proposed to find the most suitable offloading decision, and vehicles with poor cooperation capabilities are guaranteed to be treated equally in the offloading. Finally, two heuristic algorithms are presented to solve the problem with applicable complexity and to suit the practical IoV environment. Extensive simulation results prove that the proposed approach achieves remarkable performance improvements in terms of delay, service cost and the resource utilization ratios of vehicles.

Keywords: Internet of Vehicles; Vehicular Edge Computing; computation offloading; fuzzy logic; Max-Min fairness

MSC: 94D05

check for updates

Citation: Wang, S.; Peng, H.; Guo, D. Resource- and Time-Efficient Computation Offloading in Vehicular Edge Computing: A Max-Min Fairness Oriented Approach. *Mathematics* **2022**, *10*, 3735. https://doi.org/10.3390/math10203735

Academic Editor: Ioannis G. Tsoulos

Received: 8 September 2022
Accepted: 3 October 2022
Published: 11 October 2022

1. Introduction

With the rapid development of wireless communications technology and the Internet of Vehicles (IoVs), a number of vehicles are equipped with On-Board Units (OBUs) to support a variety of intelligent applications, such as autonomous driving, video-aided real-time navigation, and so on [1–4]. However, many of these applications are delay-sensitive and require a huge amount of computation resources, which brings great challenges to single User Vehicles (UVs) with limited computation resources [5]. According to the report released by Cisco [6], 300 million vehicles participating in wireless communications can generate more than 400 million gigabytes of data. It is not difficult to imagine that a huge amount of computation tasks may be generated and thus require a great deal of computation resources. Obviously, it is impossible for a UV to solely meet the requirements of such explosive tasks.

To cope with this issue, Vehicular Edge Computing (VEC) is proposed to improve the computation performance by deploying computation resources at the edge of the network [7,8]. As shown in the left part of Figure 1, through Vehicle-to-Infrastructure (V2I) [9] communications, the computation task can be unloaded to nearby RoadSide Units (RSUs) which have sufficient computation resources and thus achieve satisfying Quality of

Service (QoS) [10,11]. However, the idea of offloading computation tasks to RSUs through V2I communications has two inherent shortcomings. For one thing, due to the fast-moving characteristic of vehicles, the duration of the communication between a UV and an RSU is usually very short. It is hard to guarantee that a UV will stay in the coverage of an RSU long enough to receive the computed result. For another thing, the infrastructure laying and maintenance require high costs [12], and it is unrealistic to realize full road coverage of RSUs. The scope of V2I communications and the applicability of V2I-based computation offloading is thus limited.

In scenarios where RSUs are deployed sparsely, such as highways, Vehicle-to-Vehicle (V2V) based computation offloading [13] can better cope with the explosive growth of computation resources demand. Vehicles running on the road with idle computation resources can be regarded as "mobile servers". As illustrated in the right part of Figure 1, The UV can unload computation tasks to other Service Vehicles (SVs) within its communications range through V2V communications, forming a more flexible computation task offloading system. In doing so, the problem of short communications durations between vehicles and RSUs can be alleviated, and the resource utilization ratios of vehicles are improved. However, in existing work which is aimed in the direction of using nearby vehicles as mobile servers to unload computation tasks, vehicles are often assumed to be selfless and are always willing to donate their computation resources to strange vehicles [14]. Nonetheless, vehicles are rational and will not provide free help for other vehicles in practical scenarios. As a result, an efficient incentive mechanism is needed to promote the willingness of vehicles to participate in the task offloading process.

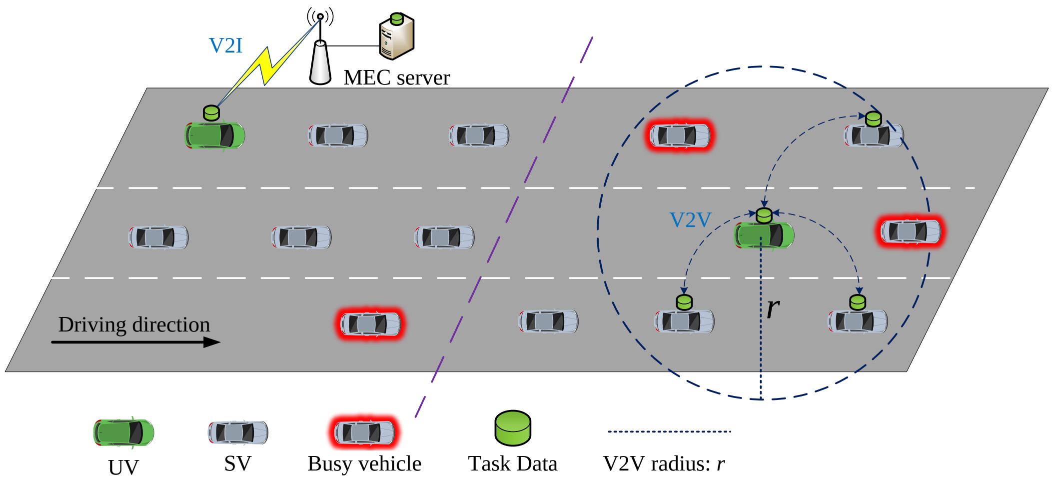


Figure 1. VEC task offloading model.

Moreover, since UV is also rational, it hopes to complete the computation task as soon as possible and with minimum cost. As a result, how to motivate vehicles to realize quick and efficient computation offloading so that the computation task can be completed timely with minimum cost, remains a big problem.

In this work, we consider a typical highway task offloading scenario where a UV unloads its computation task to SVs via V2V communications. In order to encourage SVs to participate in the cooperation process, we introduce a dynamic and fair pricing strategy for SVs, under which a UV is obligated to pay SVs for the computation resources, as well as the cooperation provided by them. Meanwhile, a Max-Min fairness-oriented approach is designed for the computation offloading decision process, which achieves good offloading performance with acceptable cost. The main contribution of this work can be summarized as follows:

(i) We propose a fuzzy logic-based [15] pricing strategy to quantify the willingness and the capability of SVs for undertaking the computation tasks unloaded from UVs. The proposed pricing strategy takes multiple parameters such as the financial situation and the volume of rentable resources of an SV, as well as the distance, speed difference and link reliability between an SV and the UV, into consideration for determining the unit price of an SV for executing the computation task. Other things being equal, the proposed pricing strategy promotes vehicles that have a small volume of rentable resources and difficult financial situations to the final SVs, thus encouraging most vehicles to join in the task offloading process and improving the overall resource utilization ratio.

(ii) We propose a Max-Min fairness-oriented [16] approach to solve the problem of finding the most suitable offloading decision. Essential to the approach is guaranteeing that vehicles that have few rentable resources and poor financial situations are treated equally to vehicles that have large resources and good financial conditions, in the process of determining the final service vehicles. This in turn is beneficial for the requesting UV to minimize the total payment, as well as for the system to maximize the resource utilization ratio and minimize the overall delay. Moreover, we present two heuristic algorithms to find the optimal solution to the formulated problem. The proposed algorithms achieve satisfying performance with applicable complexity and suit the practical IoV environment.

(iii) Extensive simulation results prove that the proposed approach achieves remarkable performance improvements in terms of the task completion delay, service cost and the utilization ratios of SVs.

The rest of this paper is organized as follows: Section 2 introduces the related work. Section 3 describes the preliminaries. The problem formulation is given in Section 4. Section 5 introduces the proposed approach in detail. Section 6 shows the simulation results and evaluates the performance. Finally, Section 7 concludes this paper.

2. Related Work

2.1. V2V-Based Computation Offloading

The authors in [17] analyzed the efficiency of a computation offloading algorithm in the vehicular cloud and improved the performance by using the computing resources of idle vehicles. To reduce the completion delay of the task and to improve the utilization ratios of vehicle resources, a two-stage cooperative offloading scheme was proposed in [18]. In the first stage, a cluster is formed among multiple vehicles and the selected cluster head is used for resource allocation. In the second stage, the fuzzy logic principle is exploited to screen cluster heads and SVs, to optimize the response time and completion delay. The authors in [19] proposed a cluster-based cooperative offloading algorithm, which applies the fuzzy logic principle to determine the appropriate cluster heads. Moreover, a stable matching method is designed to optimize energy consumption and task latency. The authors in [20] proposed fully exploring the idle resources of vehicles by utilizing the gathering period of vehicles in an urban environment. The task execution time is optimized with a Max-Min fairness scheme which is solved by the particle swarm optimization algorithm. The authors in [21] proposed a multi-hop V2V collaborative task offloading scheme, which makes full use of the multi-hop vehicular ad hoc networks to support computation offloading. The reliability model of multi-hop routing is built and the optimal offloading strategy with the lowest relaying and computing cost is executed by the binary search algorithm.

Although these works have provided some effective methods to solve the problem of optimizing the completion delay and resource utilization rate in V2V-based computation offloading, they all work under the assumption that vehicles are selfless and volunteer to provide cooperation, which is not practical and thus limiting the applications of these works.

2.2. Computation Offloading with Incentives

The authors in [22] proposed to decompose the system profit optimization problem into three sub-problems, task transmission delay optimization, the offloading ratio optimization, and finally, the offloading payment optimization. The system profit is optimized based on incentive compatibility and individual rationality of vehicles. The authors in [23] noticed that there are many parked vehicles on the road and their resources are underutilized. Therefore, the authors introduce a pricing model to offload tasks in a dynamic environment, which helps vehicles to find economical offload strategies and to improve the resource utilization of parked vehicles. In [24], the authors proposed a novel architecture that integrates satellite networks and 5G cloud-supported vehicular networks. Delay and cost are jointly optimized through a simulated annealing algorithm. Similarly, the work of [25] considered the computation offloading problem in the satellite-terrestrial Internet of Vehicles. An auction-based incentive mechanism is brought forward to motivate vehicle owners to join in the offloading process. The authors in [26] proposed a differentiated pricing scheme that sets incentive factors according to the needs of users and the volume of onboard resources. In this way, the incentive for vehicles can be adjusted in real-time. Moreover, a reverse auction-based task scheduling algorithm was proposed to select the edge nodes for task offloading.

Although the above works provide compensation for SVs, the existing incentive mechanisms still lack specific considerations for vehicle cooperation intentions. Simple economic compensation will lead to fierce competition among vehicles, and in turn, results in uneven distribution of wealth in the system, which will cause a reduction in the pertinence of the incentive mechanisms. In other words, vehicles that have strong intentions to participate in the offloading may be deprived of the opportunities if they have a small volume of computing resources or are in poor financial conditions, with the existing incentive mechanisms. In view of this, this paper firstly proposes a fuzzy logic-based pricing strategy that jointly considers the moving characteristics of vehicles in the system and the service intentions of vehicles. The unit price of a vehicle to help UV with the computation task is calculated and adjusted according to its real-time situation. After this, a Max-Min fairness-oriented offloading algorithm is designed, which works on the basis of the unit price of each vehicle determined by the pricing strategy. With careful consideration, vehicles that have a small volume of computing resources or are in poor financial conditions are favored in the determination of final SVs. In this way, the performance of the computation offloading, including the completion delay, the overall cost, and the resource utilization ratios are improved significantly.

3. Preliminaries

This paper focuses on the problem of V2V-based computation task offloading in Vehicular Edge Computing networks, as shown in the right part of Figure 1. When a UV has a computation task that needs to be offloaded to other vehicles, the UV will firstly multicast this offloading requirement message to all neighboring vehicles, asking for their cooperation. Upon receiving the request, a neighboring vehicle determines the proper price it wants to charge for providing help with the computation task, according to the fuzzy logic-based pricing strategy. It then responds to the UV with the price. After receiving all responses from the neighboring vehicles, the UV makes the optimal offloading decision based on the Max-Min fairness-oriented approach and acknowledges the appropriate proportions of the computation task to be unloaded to each SV. The negotiation process is similar to the famous "Three-Way Handshake" and the exchanged messages can be contained in beacons so that the overhead is negligible. Denote the UV as u_0, and the potential SVs set as $\boldsymbol{SV} = \{sv_1, sv_2, \dots, sv_K\}$. The computation task of u_0 is defined as $T_0 = \{D_0, G_0, T_0^{max}\}$, where D_0 represents the size of the task, G_0 is the number of CPU cycles required to process 1-bit data of the task, and T_0^{max} is the maximum acceptable completion delay of the task. The task can be divided into several sub-tasks of any size to enable synchronous processing on multiple SVs.

To obtain a microscopic vehicle movement model closer to the real high-speed scene, this paper adopts the car-following model in [27] as the mobility model. Time is slotted. At the beginning of each time slot Δt, a vehicle v_i updates its speed and then maintains this speed throughout the time slot. The update of v_i's speed at time $(t+\Delta t)$ can be expressed as follows:

$$v_i(t+\Delta t) = v_i(t) + \delta_i(t) \cdot a \cdot \Delta t, \tag{1}$$

where $v_i(t)$ denotes the speed of v_i at time t, $\delta_i(t) \in [-1,1]$ is a random adjusting parameter for acceleration or deceleration at time t, and a is the acceleration constant. The minimum and maximum allowable speeds of a vehicle are denoted as v_{min} and v_{max}, and the safety distance and the following distance between any two vehicles are denoted as L_{safe} and L_{follow}, respectively. The microscopic mobility of a vehicle is constrained by its speed and distance according to Formula (2) as follows:

$$v_i(t+\Delta t) = \begin{cases} v_{min}, & \text{if} \quad v_i(t+\Delta t) < v_{min} \\ v_{max}, & \text{if} \quad v_i(t+\Delta t) > v_{max} \\ v_i(t) + a \cdot \Delta t, & \text{if} \quad L_{i,i+1}(t) > L_{follow} \\ v_i(t) - a \cdot \Delta t, & \text{if} \quad L_{i,i+1}(t) < L_{safe} \end{cases}, \tag{2}$$

where $L_{i,i+1}(t)$ is the distance between two consecutive vehicles v_i and v_{i+1} at time t. It is noted that Formula (2) is defined according to the actual driving conditions of vehicles. The safety distance is the minimum driving distance between vehicles, which depends on the real-time speeds of the vehicles. It can ensure that the rear vehicle has enough distance to brake in case of emergency, and reduces the incidence of chain traffic accidents. The following distance is the maximum driving distance between vehicles, which can save more space for other vehicles behind.

The V2V communications among vehicles are assumed to comply with IEEE 802.11p standards [28], the data transmission rate between any two vehicles can be calculated as follows:

$$R = B \cdot log_2\left(1 + \frac{P_{up} \cdot h}{\sigma^2}\right), \tag{3}$$

where B is the channel bandwidth, P_{up} is the transmission power of a vehicle, h is the channel gain and σ^2 is the noise power. The symbols used in this paper are listed in Table 1.

Table 1. Summary of important symbols.

Symbols	Meaning
u_0	The user vehicle.
sv_j	The j^{th} service vehicle.
T_0	The computation task of u_0.
T_0^{max}	The maximum acceptable completion delay of T_0.
T_{uj}^{serv}	The achievable communication time between u_0 and sv_j.
O_j	The proportion of tasks offloaded from u_0 to sv_j.
O_j^{max}	The maximum achievable task proportion of sv_j to u_0.
$v_i(t)$	The speed of vehicle v_i at time t.
R	The V2V data transmission rate.
f_j	The amount of computing resources that sv_j can lend.
t_j	The completion delay on sv_j.
t_{total}	The total completion time of the task.
P	The total price that u_0 should pay for asking SVs.
C	The base price of unit size data of task.
α_j	The sv_j's dynamic adjustment parameters of price.
w_j	The financial condition of sv_j.
d_{uj}	The distance between u_0 and sv_j.
v_{uj}	The relative speed between u_0 and sv_j.
r_{uj}	The link reliability of the communication channel between u_0 and sv_j.

4. Problem Formulation

The task offloading process is mainly divided into three parts: task transmission, task calculation and result return. Since the result is usually small and has little impact on the delay of the whole offloading process, this paper ignores the delay of the result return. Denote the offloading decision as $\boldsymbol{d} = \{O_1, O_2, \ldots, O_j, \ldots\}$, where O_j represents the proportion of task that is offloaded to the j^{th} SV. The task transmission delay for u_0 to unload O_j proportion of the task onto vehicle sv_j can be calculated as follows:

$$t_j^{up} = \frac{O_j \cdot D_0}{R}. \tag{4}$$

Similarly, the task calculation delay by sv_j to complete the assigned task is calculated as follows:

$$t_j^{cal} = \frac{O_j \cdot D_0 \cdot G_0}{f_j}, \tag{5}$$

where f_j is the amount of computing resources that sv_j can lend. Thus, the sub-task completion delay on sv_j can be calculated as:

$$t_j = t_j^{up} + t_j^{cal}. \tag{6}$$

The computation task can only be considered as finished until all sub-tasks are completed and the calculation results are returned to the UV. Thus, the total completion time of the task can be written as:

$$t_{total} = max\Big\{t_1, \ldots, t_j, \ldots, t_{|\boldsymbol{d}|}\Big\}. \tag{7}$$

Moreover, the pricing decision for the final selected SVs is denoted as $\boldsymbol{p} = \{p_1, p_2, \ldots, p_j, \ldots\}$, where p_j is the price that sv_j asks for calculating a unit size of the task. As a result, the total price that u_0 should pay for asking SVs to complete the task can be calculated as follows:

$$P = \sum_{j=1}^{|\boldsymbol{d}|} p_j \cdot O_j \cdot D_0. \tag{8}$$

The cooperative offloading of multiple vehicles is considered to reduce the task completion delay of the UV. However, vehicles complete tasks with varied lengths of time. Since the final completion delay is the maximum of all completion delays for sub-tasks, it is preferred to minimize the final completion delay by aggregating the completion delays of all sub-tasks. In this way, the completion time of each sub-task is minimized and the overall completion delay of the task is optimized. The resource of each vehicle can be released quickly once the sub-task is finished, thus improving the resource utilization ratio of the system. The resources utilization ratio of the overall offloading system is calculated as the same in [29]:

$$Utility = \frac{\sum_{j=1}^{|\boldsymbol{d}|} t_j}{|\boldsymbol{d}| \cdot t_{total}}. \tag{9}$$

Obviously, the resource utilization ratio is a vital metric to describe the efficiency of the offloading system. A high resource utilization ratio indicates that the idle resources on the finally selected SVs are fully exploited, which is welcomed both by the UV and SVs. Meanwhile, the UV also wants to accomplish the computation task with minimum delay and cost, which requires sufficiently exploring the resources of nearby vehicles that ask for a low unit price yet have strong capabilities for providing fast and reliable cooperation. Taking all these aspects into consideration, this paper aims to design a resource- and time-efficient approach for task offloading in VEC networks. Given the definition of resource utilization ratio in Formula (9), the maximization of the resource utilization is strongly correlated to the minimization of the completion delay of the task. The optimization problem is formulated as follows:

$$maximize \quad Utility, \tag{10}$$

$$\text{s.t.} \quad C1 : 0 \leq O_j \leq 1, \tag{10a}$$

$$C2 : \sum_{j=1}^{|\boldsymbol{d}|} O_j = 1, \tag{10b}$$

$$C3 : p_j > 0, \tag{10c}$$

$$C4 : t_{total} \leq T_0^{max}, \tag{10d}$$

$C1$ and $C2$ state that the portion of the task that is offloaded to a single SV is varied between $[0, 1]$, and the sum of the size of offloaded sub-tasks equal to that of the original task. $C3$ indicates that an SV will only ask for a positive reward. $C4$ states that the total delay of the offloaded sub-tasks must not violate the maximum tolerant delay of the original task, otherwise the task computation is considered as failed.

5. Methodology

To solve the above problem, we first propose a fuzzy logic-based pricing strategy to determine the price that an SV asks for joining in the offloading cooperation. Given the pricing strategy, a Max-Min fairness-oriented decision approach is proposed to find the optimal solution, and a heuristic algorithm is further proposed to improve the complexity of the solution to better adapt to practical applications.

5.1. Fuzzy Logic Based Pricing Strategy

We propose a fuzzy logic-based pricing strategy to help vehicles to determine the price they can ask for lending their resources to complete the computation task together. The determination of the price that an SV can ask is tricky. A good price should be smartly decided for each vehicle, which not only depicts the multiple important attributes of a vehicle to serve as an SV, but also helps the vehicle to win in the competition for becoming the final SVs.

The base price of a vehicle for calculating the unit size data of a task is defined as C, to reflect the instant capability of a vehicle sv_j to help with the task computation, as well as the utility of selecting sv_j as an SV, the price that sv_j will charge u_0 for helping it to calculate a unit size data can be expressed as:

$$p_j = C \cdot (1 + \alpha_j), \tag{11}$$

where α_j is the dynamic adjusting parameter of sv_j to determine its price. It should be noted that in order to make profits, the final price of sv_j is higher than the basic price C, and $\alpha_j \in [0.2, 1]$. We employ a fuzzy information system to determine the appropriate value of α for each vehicle, according to multiple metrics that describe the capabilities and the intentions of vehicles to provide cooperation. In view of the highly dynamic characteristic of IoVs and the randomness of fast-moving vehicles, multiple attributes that represent the mobility and communications impact on the cooperation performance are analyzed and incorporated into the pricing model. Meanwhile, social attributes that reflect the willingness of vehicles to participate in the cooperation are also valued.

5.1.1. Multiple Metrics

The proposed pricing model takes five instant factors as the inputs of the fuzzy information system. For any vehicle sv_j, the price it can charge u_0 is determined by the following parameters: the financial condition of sv_j, the distance and relative speed between itself and u_0, the computing resources it has and the link reliability of the communication channel between itself and u_0. The above-mentioned parameters are denoted as $w_j(t_0)$, $d_{uj}(t_0)$, $v_{uj}(t_0)$, $f_j(t_0)$ and $r_{uj}(t_0)$, respectively, where t_0 represents the moment when sv_j receives the offloading request from u_0. The distance between sv_j and u_0 at time t_0 can be calculated as (The distance is calculated on the 2D coordinate system and the result is applicable for the 3D coordinate system):

$$d_{uj}(t_0) = \sqrt{[x_j(t_0) - x_u(t_0)]^2 + [y_j(t_0) - y_u(t_0)]^2}, \tag{12}$$

where $(x_j(t_0), y_j(t_0))$ and $(x_u(t_0), y_u(t_0))$ are the coordinates of vehicle sv_j and u_0 at time t_0. Similarly, the relative speed between sv_j and u_0 at time t_0 can be calculated as:

$$v_{uj}(t_0) = |v_u(t_0) - v_j(t_0)|, \tag{13}$$

where $v_j(t_0)$ and $v_u(t_0)$ are the speeds of vehicle sv_j and u_0 at time t_0. Considering the existence of Line of Sight (LOS) in the VEC networks, the Rician fading channel model is adopted in this paper. The Rician K-factor directly reflects the quality of the channel, and it is related to the distance between the transmitting node and the receiving node. To better describe the vehicle channel condition, we use the Rician K-factor module in [30], and there is:

$$K_{uj}(t_0) = 39.832 - 0.288 d_{uj}(t_0), \tag{14}$$

where $K_{uj}(t_0)$ is the value of Rician K-factor between sv_j and u_0 at time t_0. A large K_{uj} means weak channel fading and good channel quality between u_0 and sv_j, which makes the transmission more reliable. Thus, the link reliability of u_0 and sv_j at time t_0 is represented by $K_{uj}(t_0)$.

5.1.2. Fuzzification and Fuzzy Rules

The membership function defines the way that each value in the input space can be mapped to a membership value within the range of $[0, 1]$, which is the membership degree. Membership functions are designed by dividing each linguistic variable into an overlapping fuzzy set, which is obtained through experiments. A set of membership functions for each fuzzy variable can be expressed as:

$$Z_F = \{(y, \mu_Z(y)) : y \in Y, \mu_Z(y) \in [0, 1]\}, \tag{15}$$

where $\mu_Z(y)$ is the membership function of Z, it also indicates the degree at which y belongs to Z. The Y refers to a universe of discourse.

In this work, two membership function sets are designed in a triangular format due to their low complexity and flexibility, based on the input and output variables. The triangular membership function can be expressed as follows:

$$\mu_Z^{tri}(y) = \begin{cases} 0, & \text{if} \quad y < a \\ \dfrac{y-a}{b-a}, & \text{if} \quad a \le y \le b \\ \dfrac{c-y}{c-b}, & \text{if} \quad b \le y \le c \\ 0, & \text{if} \quad y > c \end{cases}, \tag{16}$$

where a denotes the lower bound of the triangular curve, b denotes the modal value of the triangular curve and c denotes the upper bound of the triangular curve.

Figure 2a exhibits the survey of the input membership functions of $w_j(t_0)$, $d_{uj}(t_0)$, $v_{uj}(t_0)$, and $f_j(t_0)$, and Figure 2b defines the membership function of $r_{uj}(t_0)$. Meanwhile, "Low", "Medium" and "High" are used to represent the pricing fuzzy sets of each metric. The fuzzy rules are defined based on the if-then logical operation, and each rule represents a fuzzy implication between a condition and a conclusion. An output can only be obtained after evaluation of the fuzzy rules, thus the design of the fuzzy rules is critical for the final performance. In this work, the fuzzy rules are designed based on the following considerations:

- $w_j(t_0)$: A vehicle will be proactive to participate in the offloading cooperation if it is in poor financial condition, thus it will asks a low price to lure the UV to choose it as the final SV. On the contrary, a vehicle will not be interested in the cooperation if it has good financial situation, and it will asks a high price.
- $d_{uj}(t_0)$: Since the communications quality is inversely proportional to the distance between two vehicles, a large distance represents a poor transmission quality, thus a high price for cooperation, and vice versa.
- $v_{uj}(t_0)$: The communications quality is inversely proportional to the relative speed between two vehicles, thus a large relative speed represents a poor transmission quality, thus a high price for cooperation, and vice versa.
- $f_j(t_0)$: To fully utilize the idle resources on nearby vehicles, the designed pricing model promotes vehicles with a small volume of resources. As a result, when a vehicle has a small amount of computing resources, it will lower the price to competing for serving as a SV, and vice versa.
- $r_{uj}(t_0)$: The communications quality is proportional to the link reliability between two vehicles, thus a high link reliability represents a good transmission quality, thus a low price for cooperation, and vice versa.

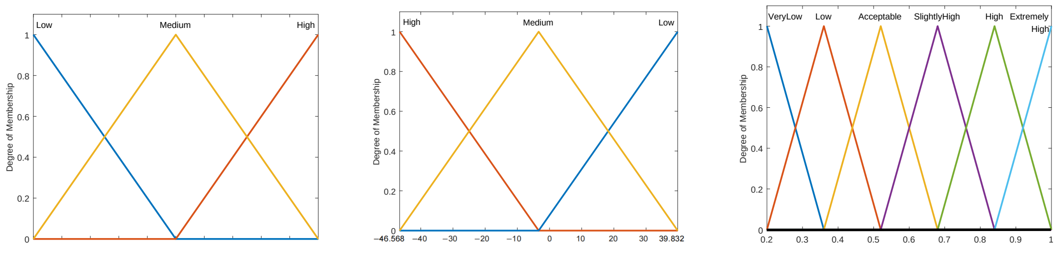


Figure 2. FIS membership function: (**a**) Input membership function of $w_j(t_0)$, $d_{uj}(t_0)$, $v_{uj}(t_0)$ and $f_j(t_0)$. (**b**) Input membership function of $r_{uj}(t_0)$. (**c**) Output membership function of α.

Each of the five input fuzzy variables has three fuzzy sets, so there are $3^5 = 243$ fuzzy rules in total. The more fuzzy sets, the more fuzzy rules, and the more details can be described. However, a large number of fuzzy sets advances the difficulty of making fuzzy rules and thus increases the calculation complexity. Therefore, it is important to find an appropriate number of fuzzy sets to strike a balance between the precision demand and the complexity. Based on the above considerations, we can obtain the fuzzy rules in all cases. Some examples of the designed fuzzy rules are illustrated in Table 2.

Table 2. Fuzzy rules examples.

Rules	$w_j(t_0)$	$d_{uj}(t_0)$	$v_{uj}(t_0)$	$f_j(t_0)$	$r_{uj}(t_0)$	α_j
1	Low	Low	Low	Low	Low	VeryLow
2	Low	Medium	Medium	Low	Low	Low
3	High	High	Low	Low	Low	Acceptable
4	High	High	Low	High	Low	SlightlyHigh
5	Medium	Medium	Medium	Medium	Medium	SlightlyHigh
6	Medium	High	High	Medium	Medium	High
7	High	High	High	High	High	ExtremelyHigh

5.1.3. Defuzzification

Our goal is to determine the value of "α" for each vehicle. The fuzzy sets of "α" are defined as: "VeryLow", "Low", "Acceptable", "SlightlyHigh", "High" and "Extremely-High". The output membership function is shown in Figure 2c, which is used to convert a fuzzy value to a numerical value. The center of gravity (COG) method is adopted for defuzzification and obtains the value of α accordingly.

5.2. Max-Min Fairness Oriented Offloading Approach

The core of the offloading decision is to determine how much proportion of the task can be offloaded to a vehicle given the vehicle's physical conditions. In other words, it is important to find out the maximum volume of computation tasks that a vehicle can handle, so as to guarantee that the final offloading decision is feasible. After that, the decision can be adjusted based on the above-proposed pricing model, under the constraint of the maximum acceptable offloading proportion of each vehicle. As mentioned above, the task offloading between the UV and an SV is completed only when the UV has received the calculation result from the SV. As a result, the UV and the SV must keep the connection between them alive the whole time. Thus, the problem of solving the maximum acceptable offloading proportion of a vehicle is transformed into the problem of determining the achievable communication time between these two vehicles.

Denote the achievable communication time between u_0 and sv_j as T_{uj}^{serv}, given the deadline constraint on the task as T_0^{max}, $T_{uj}^{serv} = min\left\{T_{uj}^{link}, T_0^{max}\right\}$, where T_{uj}^{link} is the time duration of the communication link established between u_0 and sv_j. Assume that u_0 sends out the task offloading requests to its neighboring vehicles at time t_0, the communication radius of a vehicle is r, and the time slot when the connection between u_0 and sv_j is disconnected is denoted as t_{uj}^{break}, thus, $T_{uj}^{link} = t_{uj}^{break} - t_0$.

Assume that vehicles drive in straight lines during the offloading, and overtaking is not allowed. Given the mobility model, the horizontal location update of a vehicle sv_j in each time slot can be calculated as follows:

$$\begin{aligned} x_j(t+1\cdot\Delta t) &= x_j(t) + v_j(t)\cdot\Delta t, \\ x_j(t+2\cdot\Delta t) &= x_j(t+1\cdot\Delta t) + v_j(t+1\cdot\Delta t)\cdot\Delta t, \\ &\cdots \\ x_j(t+n\cdot\Delta t) &= x_j[t+(n-1)\cdot\Delta t] \\ &\quad + v_j[t+(n-1)\cdot\Delta t]\cdot\Delta t. \end{aligned} \tag{17}$$

Therefore, the driving distance of sv_j in n time slots can be expressed as follows:

$$x_j(t+n\cdot\Delta t) = x_j(t) + \sum_{q=0}^{n-1} v_j(t+q\cdot\Delta t)\cdot\Delta t. \tag{18}$$

Thus, starting from time slot t_0, the distance between u_0 and sv_j after n time slots can be expressed as in Equation (19).

$$\begin{aligned} d_{uj}(t) &= \sqrt{\left[x_j(t)-x_u(t)\right]^2 + \left[y_j(t)-y_u(t)\right]^2} \\ &= \sqrt{\left\{\left[x_j(t_0) + \sum_{q=0}^{n-1} v_j(t_0+q\cdot\Delta t)\cdot\Delta t\right] - \left[x_u(t_0) + \sum_{q=0}^{n-1} v_u(t_0+q\cdot\Delta t)\cdot\Delta t\right]\right\}^2 + \left[y_j(t_0)-y_u(t_0)\right]^2}; \end{aligned} \tag{19}$$

$$\sqrt{\left\{\left[x_j(t_0) + \sum_{q=0}^{N-1} v_j(t_0+q\cdot\Delta t)\cdot\Delta t\right] - \left[x_u(t_0) + \sum_{q=0}^{N-1} v_u(t_0+q\cdot\Delta t)\cdot\Delta t\right]\right\}^2 + \left[y_j(t_0)-y_u(t_0)\right]^2} > r. \tag{20}$$

Assume that the connection between u_0 and sv_j is disconnected at the $(N-1)^{th}$ time slot, then we have the inequality in (20). According to the car-following mobility model, the speed change of a vehicle in each time slot is uncertain, and it is impossible to accurately calculate N in Formula (20).

To make the solution feasible and applicable, we calculate the lower bound of N in the worst-case scenario and define $t_{uj}^{break} = \lfloor N \cdot \Delta t \rfloor$. Applying Equations (4)–(6) together, the maximum acceptable offloading proportion of vehicle sv_j can be calculated as follows:

$$O_j^{max} = \frac{T_{uj}^{serv}}{D_0 \cdot \left(\frac{1}{R} + \frac{G_0}{f_j} \right)}. \tag{21}$$

Now that we have the maximum acceptable offloading proportion of each neighboring vehicle of u_0, we can find the optimized offloading decision as follows. Given the potential SVs set of u_0 as $\boldsymbol{SV} = \{sv_1, sv_2, \ldots, sv_K\}$, we firstly divide $\boldsymbol{SV}$ into $(2^K - 1)$ subsets and check if the sum of the maximum acceptable offloading proportions of vehicles in each subset meets the requirement of the task T_0. After then, we apply the Max-Min fairness allocation one by one on subsets that satisfy the required computation amount of T_0, to determine the specific proportions that should be offloaded to vehicles in the subsets. Finally, we calculate the total price u_0 should pay on each selected subset, and choose the one with the minimum cost as the final vehicle set to execute the computation offloading.

More specifically, assume there is a vehicle set $\boldsymbol{sb_i}$, which satisfies the requirement of task T_0, that is $\sum_{sv_j \in \boldsymbol{sb_i}} O_j^{max} \geq 1$, the specific Max-Min fairness oriented offloading decision process is as follows. It is observed from Equation (7) that the total delay of the computation offloading system achieves the lowest when each SV in the system has the same delay, that is $t_1 = t_2 = \ldots = t_{|\boldsymbol{d}|} = t_{op}$, thus, we propose to use a Max-Min fairness allocation method for the multi-round task allocation. Denote t_j^1 and t_{op}^1 as the delay of vehicle sv_j and the optimal delay in the first round allocation, respectively. Set $t_1 = t_2 = \ldots = t_{op}$, considering Equations (4)–(6) jointly, the optimal delay in the first round allocation can be written as:

$$\begin{aligned} t_{op}^1 &= \frac{O_1^1 \cdot D_0}{R} + \frac{O_1^1 \cdot D_0 \cdot G_0}{f_1}, \\ &= \frac{O_2^1 \cdot D_0}{R} + \frac{O_2^1 \cdot D_0 \cdot G_0}{f_2}, \\ &= \ldots \\ &= \frac{O_{|\boldsymbol{sb_i}|}^1 \cdot D_0}{R} + \frac{O_{|\boldsymbol{sb_i}|}^1 \cdot D_0 \cdot G_0}{f_{|\boldsymbol{sb_i}|}}, \end{aligned} \tag{22}$$

where O_j^1 represents the proportion of the requested task that should be offloaded to vehicle sv_j in the first round allocation. From Equation (22), it can be deduced that the general form of O_j^1 can be expressed as:

$$O_j^1 = O_1^1 \cdot \frac{f_j \cdot (f_1 + G_0 \cdot R)}{f_1 \cdot (f_j + G_0 \cdot R)}. \tag{23}$$

Moreover, according to constraint $C2$, $\sum_{j=1}^{|\boldsymbol{sb_i}|} O_j^1 = 1$, thus:

$$O_1^1 = \frac{1}{\sum_{j=1}^{|\boldsymbol{sb_i}|} \frac{f_j \cdot (f_1 + G_0 \cdot R)}{f_1 \cdot (f_j + G_0 \cdot R)}}. \tag{24}$$

Once O_1^1 is determined, the values of $O_2^1, O_3^1, \ldots, O_{|sb_i|}^1$ can be calculated according to Equation (23). However, the calculated optimal offloading proportions are obtained without considering the impacts of vehicles' moving characteristics, and it needs to be adjusted accordingly. From the above analysis, we have obtained the maximum acceptable offloading proportion of vehicle sv_j under the practical conditions as O_j^{max}. The adjustment is performed by comparing the optimal allocation proportion with the actual maximum acceptable offloading proportion of each vehicle. For any vehicle sv_j, if the calculated optimal allocation proportion is not larger than the maximum acceptable offloading proportion, which is $O_j^1 \leq O_j^{max}$, then the final proportion allocated on sv_j is $O_j^{final} = O_j^1$. Otherwise, $O_j^{final} = O_j^{max}$, and the extra proportion of task that sv_j cannot process is collected for a new round allocation, which is performed on the vehicles in the same set with extra capabilities to handle these proportion of task.

The second round allocation process is similar to the first round allocation process, a new optimal delay t_{op}^2 is calculated and proportions should be allocated to vehicles are calculated the same way in the first round. After that, if there are still some extra proportions to be allocated, a third-round allocation is initiated. The process repeats until there are no extra proportions of tasks. Finally, the proportion allocated to vehicle sv_j can be calculated as $O_j^{final} = \sum O_j^k$, where k stands for the number of allocation rounds. The pseudo-code of this method is shown in Algorithm 1. Lines 3–9 of Algorithm 1 show the subset filtering process, where vol represents the maximum task proportion volume that each subset can complete, and only subsets that can complete the whole task can be kept for further consideration. Lines 10–15 show the specific process for determining the offloading proportions and costs on the subsets filtered by the above process. Finally, lines 16–23 sort the executable subsets according to the cost they charge for the UV and the subset with the lowest cost is returned as the final offloading decision.

Algorithm 1 Max-Min fairness for task allocation

Require: T_0, $\boldsymbol{SV}$, O_j^{max}, p_j
Ensure: the offloading decision
1: Divide $\boldsymbol{SV}$ into $(2^K - 1)$ subsets, denoted as $\boldsymbol{SV} = \{\boldsymbol{sb_1}, \boldsymbol{sb_2}, \ldots, \boldsymbol{sb_{2^K-1}}\}$;
2: $i = 1$;
3: **while** $(i \leq 2^K - 1)$ **do**
4: $\quad vol = \sum_{sv_j \in \boldsymbol{sb_i}} O_j^{max}$;
5: $\quad$ **if** $vol < 1$ **then**
6: $\quad\quad \boldsymbol{SV} = \boldsymbol{SV} \backslash \{\boldsymbol{sb_i}\}$;
7: $\quad$ **end if**
8: $\quad i = i + 1$;
9: **end while**
10: $m = 1$;
11: **while** $(m \leq |\boldsymbol{SV}|)$ **do**
12: $\quad$ determine the final proportions allocated on vehicles in subset sb_m according to Equation (22)–(24), and denote the proportion allocation vector as $\boldsymbol{pv} = (O_1^{final}, O_2^{final}, \ldots, O_{|\boldsymbol{sb_m}|}^{final})$;
13: $\quad P_m = \sum_{j=1}^{|\boldsymbol{sb_m}|} p_j \cdot O_j^{final} \cdot D_0$;
14: $\quad m = m + 1$;
15: **end while**
16: $P^* = P_1, ns = 1, i = 2$;
17: **while** $(i \leq |\boldsymbol{SV}|)$ **do**
18: $\quad$ **if** $P^* > P_i$ **then**
19: $\quad\quad P^* = P_i, ns = i$;
20: $\quad$ **end if**
21: $\quad i = i + 1$;
22: **end while**
23: The ns^{th} subset with the corresponding proportion allocation result is the final offloading decision.

The Max-Min fairness task allocation requires to be executed $[|\boldsymbol{sb_m}| \cdot (|\boldsymbol{sb_m}| + 1)]/2$ times for subset $\boldsymbol{sb_m}$, and $2^K \cdot K^2$ times for the whole set $\boldsymbol{SV}$, thus, the complexity of Algorithm 1 is of the order $\mathcal{O}(2^K)$. To further reduce the complexity, we propose a heuristic algorithm as shown in Algorithm 2, whose complexity is reduced to the order of $\mathcal{O}(K^2)$, where K is the number of vehicles that respond to u_0's request.

Algorithm 2 A heuristic algorithm for task allocation

Require: T_0, $\boldsymbol{SV}$, O_j^{max}, p_j
Ensure: the offloading decision
1: Sort vehicles in $\boldsymbol{SV}$ ascendingly according to their asking prices;
2: $vol = 0, j = 1$;
3: **while** $(j \leq |\boldsymbol{SV}|$ && $vol < 1)$ **do**
4: $\quad vol = vol + O_j^{max}$;
5: $\quad j = j + 1$;
6: **end while**
7: $J = j - 1$;
8: determine the final proportions allocated on vehicles in subset $\boldsymbol{ns} = \{sv_1, sv_2, \ldots, sv_J\}$ according to Equations (22)–(24), and denote the proportion allocation vector as $\boldsymbol{pv} = (O_1^{final}, O_2^{final}, \ldots, O_J^{final})$;
9: The subset ns with the corresponding proportion allocation result is the final offloading decision.

The vehicles in $\boldsymbol{SV}$ are sorted ascendingly according to the prices asked by them (line 1). After that, vehicles are sequentially checked and the sum of the proportions that are allocated to them is compared with the total size of the computation task. The checking stops when the checked vehicles can complete the computation task together (lines 2–6). Finally, the Max-Min fairness allocation is performed on these vehicles and the final offloading decision is made (lines 7–9).

6. Performance Evaluation

In this section, we evaluate the performance of the proposed approach and compare it with referenced algorithms as follows:

(1) The maximum performance value first (maxPFV) scheme in [31]: in this scheme, the distance, relative velocity, link reliability and available computing resources are considered as fuzzy logic inputs to obtain the performance value (PFV) of each service vehicle, and UV selects the SVs sequentially according to their PFV values in a descending manner.
(2) The maximum computing resources first scheme (maxCom): UV selects SVs sequentially according to their volumes of computing resources in a descending manner.
(3) The minimum price first scheme (minPrice): UV selects SVs sequentially according to their required unit prices in an ascending manner.
(4) The random offloading scheme (Random): UV randomly selects SVs for task offloading.

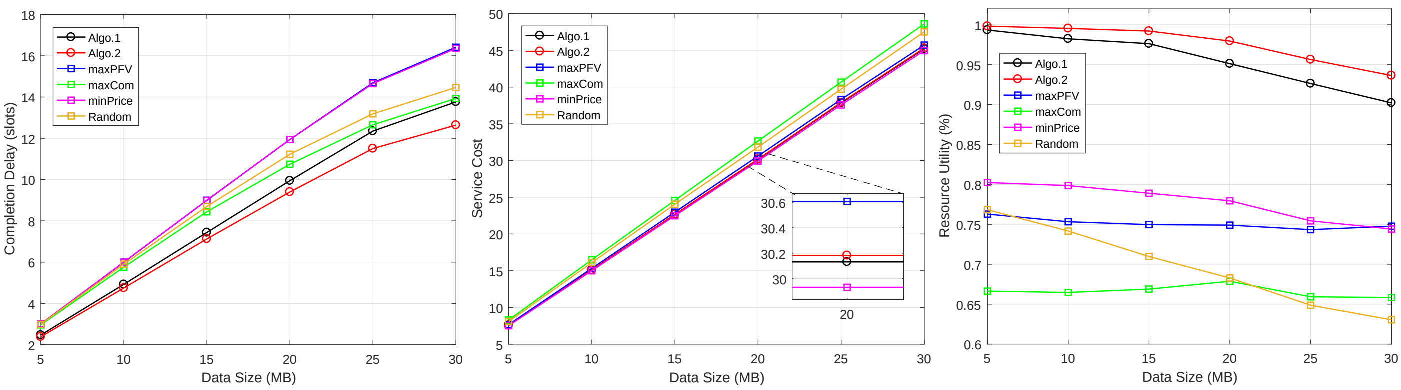
The simulation area is set as a 2-km straight road with three lanes, and vehicles run in the same direction. The specific parameter settings are shown in Table 3.

6.1. Performance Comparisons under Different Sizes of Tasks

The number of vehicles on the road is set as 50, a UV generates a computation task to be offloaded to SVs. The size of the task is randomly generated between [5, 30] MB and a maximum tolerable delay is determined between [3, 18] time slots. The change of task completion delay with the size of task data increases is shown in Figure 3a. We can see that as the size of the task increases, the task completion delay also increases. Moreover, the proposed Algorithm 2 achieves the shortest completion delay among all comparison algorithms, and the performance advantage keeps increasing with the increment of the data size of the task, indicating good time efficiency of Algorithm 2 when dealing with large size computation tasks.

Table 3. Simulation parameter settings.

Parameters	Value
R	27 Mpbs
r	300 m
v_{min}	16 m/s
v_{max}	33 m/s
L_{follow}	300 m
Δt	0.5 s
D_0	[5, 30] MB
G_0	100 cycles/bit
T_0^{max}	[3, 18] s
f_j	[0.5, 2] GHz
$w_j(t_0)$	[0, 100]
C	1.2×10^{-7}
Length of road section	2000 m
Width of single lane	3.75 m



(**a**) (**b**) (**c**)

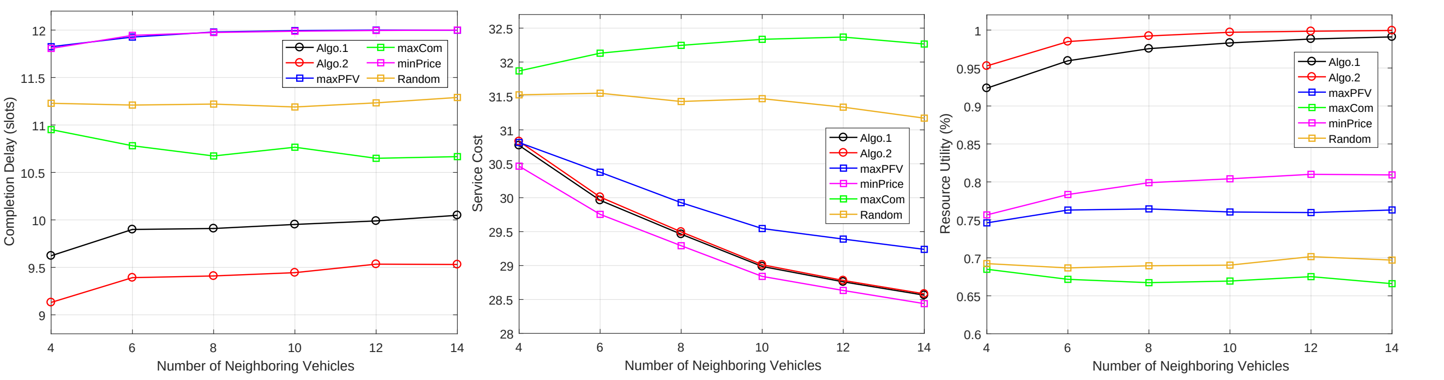
Figure 3. Performance comparisons under different sizes of tasks: (**a**) Completion delay vs Data size; (**b**) Service cost vs Data size; (**c**) Resource utility vs Data size.

Figure 3b shows the experimental results of the service cost under different sizes of tasks. It can be seen that as the amount of task data increases, the service cost also increases. As expected, the cost of minPrice is the lowest among all comparison algorithms. Since minPrice is designed to minimize the service cost without consideration of other factors. However, the good performance of minPrice on the service cost is accomplished at the sacrifice of other important performance metrics. It can be seen that minPrice performs poorly in the completion delay (Figure 3a) and the resource utility (Figure 3c). Overall, the performance of minPrice is not satisfying. In the meantime, it is noticed that the proposed Algorithm 1 has the second-best performance on the service cost, and Algorithm 2 has the third-best service cost. Thus, the proposed Algorithms 1 and 2 achieve near-optimal performance on the service cost.

From the simulation results in Figure 3c, it can be seen that the resource utility decreases with the increment of the data size of the task. The reason is that due to the maximum acceptable delay constraint of the task, when the data size of the task is large, it is difficult to complete the task timely with a few SVs. Instead, UV has to employ more SVs to execute sub-tasks simultaneously to meet the delay constraint. In doing so, each SV is assigned a relatively smaller size of sub-task, compared with that when the task size is small. Overall, UV employs more SVs to do the computation and each SV is assigned a smaller size of sub-task, thus degrading the resource utilization ratio of each SV. Moreover, there is a clear tendency that with the increment of data size, the resource utility of Random degrades drastically and becomes the lowest among all comparison algorithms after the data size expands to 20 MB and above. It is understandable since a UV always randomly selects vehicles as SVs to unload sub-tasks, and the probability for the UV to find the optimal set for computation offloading is low, especially with a large-size task. Nonetheless, it is clearly seen that the proposed algorithms maintain the highest resource utility among all algorithms, with different data sizes of the task, which justifies the resource efficiency of the proposed Algorithms 1 and 2.

6.2. Performance Comparisons under Different Number of Neighboring Vehicles

In this part, the data size of the task is set to 20 MB, and the maximum tolerable delay constraint is set to 12 s. The number of neighboring vehicles is randomly generated between [4, 14]. Figure 4a illustrates the completion delay of compared algorithms under varied numbers of neighboring vehicles. It is seen that the proposed Algorithm 2 achieves the shortest delay of all algorithms, and Algorithm 1 has the second shortest delay with different numbers of neighboring vehicles. The completion delay of both algorithms slowly increases with the number of neighboring vehicles, and the trend becomes faint when the number of neighboring vehicles becomes large enough.



(**a**) (**b**) (**c**)

Figure 4. Performance comparisons under different numbers of neighboring vehicles: (**a**) Completion delay vs Number of neighboring vehicles. (**b**) Service cost vs Number of neighboring vehicles. (**c**) Resource utility vs Number of neighboring vehicles.

Figure 4b gives the service cost of all algorithms with different numbers of neighboring vehicles. Except for maxCom and Random, the rest of the four algorithms have similar performance trends in that the service cost decreases with the number of neighboring vehicles. The reason is that given more neighboring vehicles, UV is more likely to find SVs with low prices, thus reducing the cost. In the meantime, since UV inclines to choose SVs with low prices (whose computing capabilities may not be as strong as those with high prices), it may require extra time to complete the task computation, thus prolonging the completion delay.

As explained above, the proposed Algorithms 1 and 2 favor choosing SVs with low prices which have a small volume of computing resources and treat SVs equally through Max-Min fairness allocation. In doing so, the idle resource on each SV can be utilized sufficiently and in turn, the overall resource utilization ratio is optimized. As Figure 4c shows, the proposed Algorithm 2 achieves the highest resource utility among all compared algorithms, and Algorithm 1 achieves the second best. Even when the number of vehicles around UV is small, the proposed algorithms can still manage to reach high resource utilities over 90%, which are far ahead of the rest of the algorithms. Moreover, it is seen that the resource utility slightly increases with the number of neighboring vehicles. Since UV has more potential SVs to choose from, SVs that have a small amount of resources and ask for low prices are selected as the final SVs. Their small amount of resources are equally treated and fully utilized with the proposed algorithms, thus improving the resource utility.

7. Conclusions

Computation offloading in vehicular edge computing is a new paradigm for intelligent transportation in IoVs. This paper aims at the problem of promoting vehicles on the road to provide efficient cooperation for completing computation tasks. To accurately quantify the capabilities of vehicles to execute the task, a fuzzy logic-based pricing strategy is designed to help each vehicle determine the most suitable price for lending its computing resource and providing cooperation. Based on the prices, a Max-Min fairness-oriented approach is proposed and two algorithms are presented to solve the problem. The algorithms favor choosing vehicles that have a small amount of resources and ask for low prices to be the final service vehicles, thus utilizing the idle resources on vehicles sufficiently. Simulation results show that the proposed approach achieves good performance in the completion delay and resource utility with an acceptable service cost.

The proposed approach considers single-hop cooperation among vehicles for completing the computation task. When the task is extremely large or vehicles running within the communication range of a UV are sparse, the task may not be accomplished successfully. An interesting future research direction for this work is to employ a multi-hop collaborative computation offloading mechanism for enhancing the system adaptability for a wide range of task sizes and time-varying densities of vehicles in the network.

Author Contributions: Conceptualization, S.W.; Data curation, H.P.; Funding acquisition, S.W.; Investigation, D.G.; Methodology, S.W.; Software, H.P.; Writing—original draft, S.W. and H.P.; Writing—review & editing, D.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (Grant No. 61962032, 61561029), Yunnan Ten Thousand Talents Plan Young & Elite Talents Project, and Yunnan Province Fund for Excellent Young Scholars (Grant No. 202001AW070003).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhu, X.; Luo, Y.; Liu, A.; Bhuiyan, M.Z.A.; Zhang, S. Multiagent deep reinforcement learning for vehicular computation offloading in IoT. *IEEE Internet Things J.* **2020**, *8*, 9763–9773. [CrossRef]
2. Guo, J.; Huang, G.; Li, Q.; Xiong, N.N.; Zhang, S.; Wang, T. STMTO: A smart and trust multi-UAV task offloading system. *Inf. Sci.* **2021**, *573*, 519–540. [CrossRef]
3. Wang, S.; Zhang, Q.; Chen, G. V2V-CoVAD: A vehicle-to-vehicle cooperative video alert dissemination mechanism for Internet of Vehicles in a highway environment. *Veh. Commun.* **2022**, *33*, 100418. [CrossRef]
4. Du, J.; Yu, F.R.; Chu, X.; Feng, J.; Lu, G. Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization. *IEEE Trans. Veh. Technol.* **2018**, *68*, 1079–1092. [CrossRef]
5. Wang, J.; Feng, D.; Zhang, S.; Tang, J.; Quek, T.Q. Computation offloading for mobile edge computing enabled vehicular networks. *IEEE Access* **2019**, *7*, 62624–62632. [CrossRef]
6. Yang, Q.; Zhu, B.; Wu, S. An architecture of cloud-assisted information dissemination in vehicular networks. *IEEE Access* **2016**, *4*, 2764–2770. [CrossRef]
7. Wu, Y.; Wu, J.; Chen, L.; Yan, J.; Luo, Y. Efficient task scheduling for servers with dynamic states in vehicular edge computing. *Comput. Commun.* **2020**, *150*, 245–253. [CrossRef]
8. Li, X.; Dang, Y.; Aazam, M.; Peng, X.; Chen, T.; Chen, C. Energy-efficient computation offloading in vehicular edge cloud computing. *IEEE Access* **2020**, *8*, 37632–37644. [CrossRef]
9. Ahangar, M.N.; Ahmed, Q.Z.; Khan, F.A.; Hafeez, M. A survey of autonomous vehicles: Enabling communication technologies and challenges. *Sensors* **2021**, *21*, 706. [CrossRef] [PubMed]
10. Wang, R.; Zeng, F.; Deng, X.; Wu, J. Joint computation offloading and resource allocation in vehicular edge computing based on an economic theory: Walrasian equilibrium. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 3971–3983. [CrossRef]
11. Dai, P.; Hu, K.; Wu, X.; Xing, H.; Teng, F.; Yu, Z. A probabilistic approach for cooperative computation offloading in MEC-assisted vehicular networks. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 899–911. [CrossRef]

12. Zhou, Z.; Liu, P.; Feng, J.; Zhang, Y.; Mumtaz, S.; Rodriguez, J. Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3113–3125. [CrossRef]
13. Waheed, A.; Shah, M.A.; Mohsin, S.M.; Khan, A.; Maple, C.; Aslam, S.; Shamshirband, S. A Comprehensive Review of Computing Paradigms, Enabling Computation Offloading and Task Execution in Vehicular Networks. *IEEE Access* **2022**, *10*, 3580–3600. [CrossRef]
14. Tang, C.; Wu, H. Optimal computational resource pricing in vehicular edge computing: A Stackelberg game approach. *J. Syst. Archit.* **2021**, *121*, 102331. [CrossRef]
15. Muhammad, K.; Obaidat, M.S.; Hussain, T.; Ser, J.D.; Kumar, N.; Tanveer, M.; Doctor, F. Fuzzy logic in surveillance big video data analysis: Comprehensive review, challenges, and research directions. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–33. [CrossRef]
16. Gupta, A.; Amudala, D.N.; Sharma, E.; Budhiraja, R. Max-Min Fairness for Wireless-Powered Spatially Correlated Massive MIMO Multi-way Relaying. In Proceedings of the ICC 2021-IEEE International Conference on Communications, Montreal, QC, Canada, 14–23 June 2021; pp. 1–6.
17. de Souza, A.B.; Rego, P.A.L.; de Souza, J.N. Exploring computation offloading in vehicular clouds. In Proceedings of the 2019 IEEE 8th International Conference on Cloud Networking (CloudNet), Coimbra, Portugal, 4–6 November 2019; pp. 1–4.
18. Buda, S.; Guleng, S.; Wu, C.; Zhang, J.; Yau, K.L.A.; Ji, Y. Collaborative vehicular edge computing towards greener ITS. *IEEE Access* **2020**, *8*, 63935–63944. [CrossRef]
19. Bute, M.S.; Fan, P.; Liu, G.; Abbas, F.; Ding, Z. A cluster-based cooperative computation offloading scheme for C-V2X networks. *Ad Hoc Netw.* **2022**, *132*, 102862. [CrossRef]
20. Chen, C.; Chen, L.; Liu, L.; He, S.; Yuan, X.; Lan, D.; Chen, Z. Delay-optimized v2v-based computation offloading in urban vehicular edge computing and networks. *IEEE Access* **2020**, *8*, 18863–18873. [CrossRef]
21. Deng, Z.; Cai, Z.; Liang, M. A multi-hop VANETs-assisted offloading strategy in vehicular mobile edge computing. *IEEE Access* **2020**, *8*, 53062–53071. [CrossRef]
22. Ning, Z.; Dong, P.; Wang, X.; Hu, X.; Liu, J.; Guo, L.; Hu, B.; Kwok, R.; Leung, V.C. Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks. *IEEE Trans. Mob. Comput.* **2020**, 21, 1319–1333. [CrossRef]
23. Huang, X.; Yu, R.; Ye, D.; Shu, L.; Xie, S. Efficient workload allocation and user-centric utility maximization for task scheduling in collaborative vehicular edge computing. *IEEE Trans. Veh. Technol.* **2021**, *70*, 3773–3787. [CrossRef]
24. LiWang, M.; Dai, S.; Gao, Z.; Du, X.; Guizani, M.; Dai, H. A computation offloading incentive mechanism with delay and cost constraints under 5G satellite-ground IoV architecture. *IEEE Wirel. Commun.* **2019**, *26*, 124–132. [CrossRef]
25. Zhang, X.; Zhang, H.; Dai, S.; Liu, Y. An Incentive Mechanism for Computation Offloading in Satellite-Terrestrial Internet of Vehicles. In Proceedings of the 2021 7th International Conference on Computer and Communications (ICCC), Chengdu, China, 10–13 December 2021.
26. Xue, J.; Hu, Q.; An, Y.; Wang, L. Joint task offloading and resource allocation in vehicle-assisted multi-access edge computing. *Comput. Commun.* **2021**, *177*, 77–85. [CrossRef]
27. Zhou, H.; Liu, B.; Luan, T.H.; Hou, F.; Gui, L.; Li, Y.; Yu, Q.; Shen, X. Chaincluster: Engineering a cooperative content distribution framework for highway vehicular communications. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2644–2657. [CrossRef]
28. Dai, Y.; Xu, D.; Maharjan, S.; Zhang, Y. Joint load balancing and offloading in vehicular edge computing and networks. *IEEE Internet Things J.* **2018**, *6*, 4377–4387. [CrossRef]
29. Chen, C.; Zhang, Y.; Wang, Z.; Wan, S.; Pei, Q. Distributed computation offloading method based on deep reinforcement learning in ICV. *Appl. Soft Comput.* **2021**, *103*, 107108. [CrossRef]
30. Fan, T.; Chen, W.; Li, C.; Shu, X.; Chang, F. Small scale characteristics analysis on highway V2V channel for intelligent transportation systems. In Proceedings of the 2019 5th International Conference on Transportation Information and Safety (ICTIS), Liverpool, UK, 14–17 July 2019; pp. 789–793.
31. Bute, M.S.; Fan, P.; Zhang, L.; Abbas, F. An Efficient Distributed Task Offloading Scheme for Vehicular Edge Computing Networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 13149–13161. [CrossRef]