

Computing Offloading Decision Based on Adaptive Estimation of Distribution Algorithm in Internet of Vehicles

Fahong Yu, Shanwei Polytechnic, China*

Meijia Chen, Shanwei Polytechnic, China

Bolin Yu, Shenzhen Institute of Information Technology, China

ABSTRACT

Aimed to improve the efficiency of computing offloading in internet of vehicles (IoV), a collaborative multi-task computing offloading decision mechanism with adaptive estimation of distribution algorithm for MEC-IoV was proposed in this paper. The algorithm considered the energy and time consumption as well as priority among different tasks. It presented a local search strategy and an adaptive learning rate according to the characteristics of the problem to improve the estimation of distribution algorithm. Experimental results show that compared with other offloading strategies, the proposed offloading strategy has obvious effects on the total cost optimization; the solutions quality of AEDA is 86.6% of PSO and 67.3% of GA.

KEYWORDS

Adaptive Estimation of Distribution Algorithm, Computing Offloading, Internet of Vehicle

1. INTRODUCTION

With the rapid development of the Internet of Things (IoT), it is inevitable to provide real-time, low-latency services to realize the integration of storage and processing power. Due to massive data computing have gradually been emerged and high system cost of communication and computing, it will be difficult to compute for loading of remote task based on cloud computing (Y. Dai, et al. 2021). Furthermore, it is urgent to solve the contradiction between limited computing, long-term continuous low latency and high quality of service requirements with the rapid development of mobile devices (S. Sharma, et al. 2019). As a technology integrated with the Internet of Things, MEC can provide services and computing center for users nearby and has the possibility to supply real-time, low-latency services. A lot of data centers are relatively small even with some ability of edge computing. Compared with the traditional Cloud Architecture, it seems to be more suitable for the needs of latency, responsiveness and Privacy protection. At present, it has become a trend to combine mobile computing technology with wireless communication technology to promote the development of mobile communication (L. Yao, X. Xu, et al. 2022). MEC can not only increase user experience and utilization rate of bandwidth resources, but also provide some applications for innovation of service by sinking the computing power to the mobile edge.

DOI: 10.4018/IJCINI.312250

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

In order to improve the urban traffic conditions, the Internet of Vehicle (IoV) as a new paradigm is introduced to enhance the information interaction between vehicles and people (T. Ni, et al. 2019). The Internet of Vehicles (IoV) is a kind of new mode, driven by the latest advancements in vehicular communications. In the IoV environment, the vehicle is connected to devices such as smart cameras, sensors and actuators. By the transmitters and receivers in above signal collecting system, vehicles can connect to remote infrastructure and other vehicles (X. Xue, et al. 2020). Vehicles can communicate with remote infrastructures and other vehicles users through these devices (Y. Dai, et al. 2020). The computing tasks of vehicle users can be delivered to the associated MEC servers based on a strategy to improve computation performance greatly (T. Ni, Y. Yao, H. Chang, et al. 2019). However, it is difficult to Rely on lightweight edge servers which set on the roadside for a lot of computing tasks with different granularity and quality of service (QoS) requirements. How to ensure the normal and efficient computation for these complex services will be a challenge (L. Liu, et al. 2019). Some vehicles maybe idle for task operation for computing tasks in Internet of Vehicle (K. Zhang, et al. 2020). How to improve the computing offloading efficiency is an important problem in IoV. Most existing works focus on offloading tasks to MEC servers while the computing capacity of the on-board unit (OBU) was Neglected.

MEC can be used as the core access node for the transmission and processing of edge business tasks of the Internet of Vehicles (Shen, X., et al. 2022). At the center of the cloud model, features related to average task latency and resource costs cannot be ignored due to the distance between the Internet of Vehicle devices and the data center at the edge of the vehicle equipment (Song Yubo, et al. 2021). As the number of mobile terminals increase exponentially, high latency may be a problem for some applications involving communication between ends. When all tasks were offloaded to the edge server due to the channel of communication or network were be congested, the latency may be longer and the execution may be too slowly. In such condition, some potential traffic safety would be caused if the edge server undertakes all tasks

The vehicle users with some tasks take into account the computing ability and maximum service time of each service vehicle to determine the percentage of tasks assigned to each service vehicle to minimize the task execution latency. Combined with the above considerations, this paper proposes a cooperative task scheduling scheme, in which one task vehicle and multiple service vehicles jointly perform on-board tasks. Considering the computing power of on-board unit (OBU) and the computing resources of edge server, a task offloading strategy based on adaptive estimation of distribution algorithm (AEDA) was provided to optimize the latency and energy consumption so as to obtain the lowest total system cost of task offloading by using the computing resources of OBU and MEC reasonably. The key contribution of this paper include:

1. The model of optimization problem was built aiming at the energy consumption and latency in vehicle network by linear weighting algorithm.
2. An offloading strategy of collaborative computing was designed according to the nearest processing principle, the computing power of OBU and the computing resources of edge server. It used the computing resources of OBU and edge server reasonably to complete the task together, which have improved the transmission and response speed.
3. Considering the sequence of task execution and the relationship between tasks, a task offloading priority matrix was designed.
4. An improved estimation of distribution algorithm with a local search strategy and an adaptive learning rate according to the characteristics of the task offloading to improve the estimation of distribution algorithm.

The rest of the paper is organized as follows. The related works are reviewed in Section 2. The system model and the optimization algorithm are presented in Section 3. The simulation results and analyses are presented in Section 4. Finally, summarizes this paper.

2. RELATED WORK

It is becoming a new paradigm for mobile edge computing in distributed systems, which can promote efficiency of computing, reduce task execution latency and make useful for the resources of other terminals. Due to the limited computing power, storage and energy of mobile devices, task scheduling is a classic method to transfer tasks to external platforms (S. Lv, Y. Liu, et al. 2021 and Liu, L, et al. 2021). It can improve computing efficiency, reduce task completion time and effectively make use of the resources of other devices. Since the edge computing network has the characteristics of ultra-dense deployment and simultaneous access of a large number of users, the selection of user computing task mode is extremely important and directly determines the computing time and cost of the system.

Nowadays, there are several researches focus on MEC such as server placement, application placement, task offloading, network architecture, and so on. It is a hot research direction of MEC that how mobile terminals offload tasks to MEC server in order to obtain the minimum processing cost and reasonable offloading decision of all tasks to make full use of the services provided by edge terminals for making useful of the services provided by the edge terminals. It is imperative to combine 5G with artificial intelligence to optimize the management network under the planning of 5G technology. A depth data-driven anomaly detection method based on moving edge computing is proposed (K. Zhang, et al. 2019). While it is troublesome for the high energy consumption. Furthermore, A hybrid computing strategy for the mobile edge computing is provided (Wang L, et al. 2021) to minimize the energy consumption of terminal devices, and a hybrid online method for computing offload adopting deep learning method is presented. To meet the intensive computation demands, an approach that minimizes the total energy consumption of the system by jointly optimizing the task offloading decision through decoupling it into subproblems and leveraging the block coordinate descent (D. Pliatsios, P. Sarigiannidis, T. Lagkas, et al. 2022). Xiong et al. (2020) designed an optimization algorithm for the allocation of network resources and computing resources to minimize the transmission delay and computation time. The vehicle-based fog node is modeled mathematically according to the queuing theory, and an approximate method for solving the unloading optimization problem is introduced.

Moreover, machine learning technology was applied into offloading algorithms to improve the performance of several edge computing. A novel algorithm for allocating computation and caching resources dynamically was proposed (Q. Zhang, et al. 2020). Based on the model, a deep reinforcement learning-based algorithm is applied to maximize system utility (T. Q. Dinh, et al. 2018). To complete task offloading and calculation with lower task delay and lower energy consumption, a dynamic task offloading scheme based on deep reinforcement learning considering the situation of multiple MEC servers. (Zhang, D., Cao, L., Zhu, H. et al. 2022). Prof. Chen et al. (2021), proposed an improved the search tree algorithm by using the branch and bound method to solve the delay minimization problem of computational offloading and resource allocation. By establishing the cloud edge cooperation model, Zhao et al. (2020), designed the optimal decision-making scheme to transfer the request to the edge server or cloud server for serial processing, in which different mobile device requests should pass through the access point in chronological order. In order to maximize resource utilization, Dr. Ning et al. (2019), proposed to combine multiple edge servers for cache allocation and computational offload. To further meet the requirements of low-latency applications, a deep recurrent Q-network considering the order of priority was introduced into the vehicle tasks offloading (Ting Wang, Xiong Luo, et al. 2022).

Different from the traditional research methods of communication, caching and computing technologies separately, an integration strategy was proposed to dynamically coordinate communication, cache and computing (Mu, L.; Ge, B.; Xia, C., et al. 2022). Furthermore, a deep reinforcement learning method that maximizes the reward function is used to solve the joint optimization problems, which is defined as the comprehensive revenue from communication, caching and computing.

In addition, To quickly and efficiently make an optimal computation offloading decision for individual vehicles, A self-learning based distributed computation offloading scheme based on a game-

theoretic model to reach the Nash Equilibrium for IoV through establishing an offloading framework with communication (Q. Luo, C. Li., et al. 2021). To evaluate the impact of uncertain interconnection between the vehicle users and MEC servers on offloading decision-making so as to avoid serious degradation of the offloading efficiency, a task-offloading decision mechanism with particle swarm optimization with adaptive inertia weight factor (Cheng, J., Guan, D. 2021). A distributed computation offloading inspired by game theory and proposed a model-free reinforcement learning algorithm. To minimize both latency and energy consumption, the problem of multi-objective optimization can be transferred into a problem of single-objective optimization by the strategy of weighting coefficients. To minimize the total system costs on both the edge devices and service providers, a dual-side optimization algorithm for MEC was proposed (H. Peng, et al. 2020).

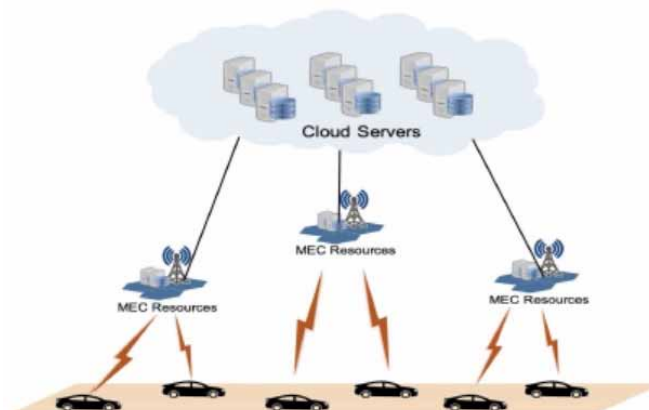
All above works is an excellent solution. The computing offloading decision in Internet of Vehicles often be solved as a single-objective optimization to minimizing latency, cost, or energy consumption and the problem of multi-objective optimization can be transferred to a single objective optimization by a weighted sum algorithm. Furthermore, the differential requirements of computation tasks were considered seldomly for most of existing offloading methods. Based on those existing works, this paper takes the research a step further. Considering the computing power of on-board unit (OBU) and the computing resources of edge server, an offloading strategy of task based on adaptive estimation of distribution algorithm was provided to optimize the latency and energy consumption so as to obtain the lowest total system cost of task offloading by using the computing resources of OBU and MEC reasonably.

3. PROBLEM FORMULATION AND SOLUTIONS

In this section, the problem is formulated mathematically and the proposed solution is described in detail.

Consider a general application of cloud-edge computing (IoV-CEC) in Internet of Vehicles system with N vehicles users and several MEC servers to execute operation of computation offloading. It is assumed that there are two-way roads and the sharing scheme of all computing devices with the same uplink spectrum in Internet of Vehicles system. The edge computing communication framework in Internet of vehicles as figure 1 which contained vehicle computing, edge computing and cloud processing. The on-board computing performs task calculation by the on-board unit of the vehicle, the road side unit (RSU) consisted of edge computing and the storage capacity with certain computing. The cloud processing consists of high-performance server clusters.

Figure 1. Diagram of Internet of Vehicles communication framework



3.1 The Calculation Offloading Model

Consider a IoV cloud-edge computing (IoV-CEC) system with N vehicle users and multiple MEC servers to perform computation offloading operation. According to the principle of proximity, the deployment of real-time tasks was adopted to make full use of the computing power of the close vehicle unit and the computing resources of the edge server, which reduced the transmission latency between vehicle and data center in remote cloud computing. Furthermore, it can collect real-time information of the vehicle, such as location, vehicle condition, road condition, etc. so as to improve the service quality, realize real-time calculation and rapid response. Therefore, the condition of three scheduling decisions were considered in this paper and the scheme of offloading tasks to the remote cloud is ignore:

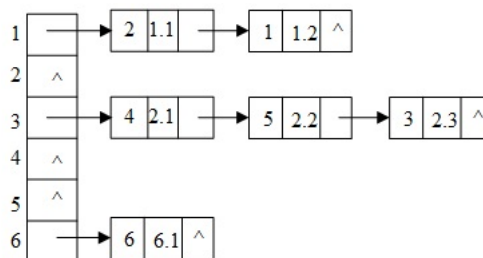
1. Local execution: The calculation task is executed in the local OBU instead of offloading the task.
2. Executing on nearby vehicles by task offloading: The task execution on the surrounding vehicles with computing power by Internet of vehicles.
3. Executing on the edge server by task offloading through the communication from vehicle to roadside unit server.

Let n tasks $T = \{ T_1, T_2, \dots, T_n \}$. Each component of T_i can be expressed in the form of a triple. $T_i = \{ d_p, c_p, r_i \}$. d_p, c_p and r_i is the data amount of task T_i , the data amount of task calculating and the data amount of returned result respectively. Assuming f_v and f_m is the computing power of the on-board unit and the edge server respectively.

The latency matrix can be given as $T_{n \times 3}$, in which the element t_{ij} represents the execution latency during the decision mode j selected by task i . The energy consumption matrix $E_{n \times 3}$, in which the element e_{ij} represents the energy consumption when the task i selection decision mode j .

The priority of tasks offloading needs to be considered in the process of actual task execution since there are data connection between tasks. The tasks' priority model can be established as a linked list with row pointer. The priority value of each task can be represented as a decimal according to a linked list with row pointer which is established based on the priority model $U_{n \times n}$. For the decimal of each node in linked list with row pointer, the integer part represents the number of tasks which used to identify whether they are relevant for two task and the decimal part represents the priority of the task for those tasks with same integer part. If two tasks T_i and T_j with different integer part in their priority values, it is independent of each other for T_i and T_j . On the contrary, the priority is determined by comparing the decimal part of tasks T_i and T_j . By this strategy, the priority value sequence of tasks can be established with n decimal. There is the priority value list $L = [1.2, 1.1, 2.3, 2.1, 2.2, 6.1]$ according to an instance figure 2.

Figure 2. Linked list with row pointer



Local execution. Considering the task carry out on local OBU and let the execution sequence of the tasks is first-come-first-served. The latency of T_i is the summation of executing latency and waiting time. The variables e_{il} and t_{il} represents the energy consumption of the local on-board unit CPU and latency when the task i is executed:

$$t_{il} = \begin{cases} \frac{c_i}{f_v} & i = 1 \\ \sum_i^{N_w} \frac{c_i}{f_v} & i > 1 \end{cases} \quad (1)$$

Here the expression N_w represents the number of tasks to be executed in the waiting pool and e_{il} represents the energy consumption of task i performed locally:

$$e_{il} = t_{il} \times p_{cpu} \quad (2)$$

Here the expression p_{cpu} represents the CPU power of on-board unit.

Executing on nearby vehicles by task offloading. The vehicle is regarded as a sensing node in V2V topology network. The task can be executed on the idle vehicle and calculation result will be returned through communicating strategy among V2V network. The information transmission rate between on-board unit and nearby OBU can be get as formula (3) according to Shannon's second theorem:

$$C(P) = w_v \log_2 \left(1 + \frac{p_1^2 \left(\frac{1}{l_0}\right)^{p_2 p}}{w_v \rho_0} \right) \quad (3)$$

Here the variable w_v represents the communication channel bandwidth of V2V network, p_1 and p_2 is the channel fading factor and path loss factor of the uplink link respectively, l_0 is the linear distance between the vehicle and nearby vehicles, and ρ_0 is the Gaussian noise density inside the communication channel.

The total latency t_{i2} included upload latency, task calculation result latency and calculation result return latency when a task i is offloaded to other vehicles for execution, which can be calculated as formula (4):

$$t_{i2} = \frac{d_i}{C(p_u)} + \frac{c_i}{f_v} + \frac{r_i}{C(p_d)} \quad (4)$$

Here the variable p_u and p_d is the power of uploading data and downloading data of on-board unit respectively.

The total energy consumption e_{i2} can be calculated as formula (5) which containing the energy consumption of tasks uploaded to surrounding vehicle OBU, offloaded from surrounding vehicle OBU and energy consumption of calculation:

$$e_{i2} = p_u \times \frac{d_i}{C(p_u)} + p_o \times \frac{c_i}{f_v} + p_d \times \frac{r_i}{C(p_d)} \quad (5)$$

Here the variable p_o is the calculates the power of the on-board unit.

Executing on the edge server. The task is executed on the edge server by uploading the calculation task to the nearby RSU in V2R network. The calculation result is transmitted back to the on-board equipment from the RSU after calculation task was completed on edge server. The information transmission rate of communicating between the on-board unit and the edge server is indicated as formula (6):

$$C'(P) = w_r \log_2 \left(1 + \frac{p_1^2 \left(\frac{1}{l_1}\right)^{p_2 p}}{w_r \rho_0} \right) \quad (6)$$

Here the variable w_r represents the communication channel bandwidth of V2V network and l_1 is the linear distance between the vehicle and RSU.

The total latency t_{i3} can be calculated as formula (7) which containing upload latency, calculation latency and latency of returning calculation result:

$$t_{i3} = \frac{d_i}{C'(p_u)} + \frac{c_i}{f_m} + \frac{r_i}{C'(p_d)} \quad (7)$$

Here the variable p_u and p_d is the power of uploading data and downloading data of on-board unit respectively and p_o is the calculates the power of the on-board unit.

The total of energy consumption e_{i3} can be calculated as formula (8) which containing energy consumption of the tasks uploaded to the edge server, offloaded from the edge server and calculation:

$$e_{i3} = p_u \times \frac{d_i}{C'(p_u)} + p_m \times \frac{c_i}{f_m} + p_d \times \frac{r_i}{C'(p_d)} \quad (8)$$

Here the variable p_m is the calculated power of the edge server.

In summary, the optimization model can be established based on the optimization objectives of latency and energy consumption. Furthermore, the decision can be selected by a decision matrix $M_{n \times 3}$. The element m_{ij} represents whether to select the j -th decision for the task i , where $m_{ij} = 1$ if the j -th decision is selected by the task i and $m_{ij} = 0$ if the j -th is not decision selected by the task i . The total of the j -th decision selected by the task i is 1, i.e. $\sum_{j=1}^M m_{ij} = 1$ which means only one decision can be selected from M decision models for each task.

It is a multi-objectives optimization involving total latency that upload the tasks and executing, the energy consumption and total latency in tasks offloading to obtain the optimal resource allocation strategy for Internet of vehicles. By a linear weighted method to simplify the optimization, the objective function can be given as formula (9):

$$\min(f) = \min\left(w_t \times \sum_{i=1}^n \sum_{j=1}^M m_{ij} \times t_{ij} + w_e \times \sum_{i=1}^n \sum_{j=1}^M m_{ij} \times e_{ij}\right) \quad (9)$$

Here the variable w_t and w_e is the coefficient of latency and energy consumption, which can be set according to the current status of on-board equipment and task requirements. At the same latency, the sum of the coefficient of latency and energy consumption is 1, i.e. $w_t + w_e = 1$.

The coefficient of latency and energy consumption is dynamic. When the vehicle's energy is sufficient, the weighted coefficient is set as $w_t > w_e$. Alternatively, the weighted coefficient is set as $w_t < w_e$. In the actual scenario, the reasonable allocation of latency and energy consumption can be determined according to the multi-attribute decision-making theory.

3.2 The Description of Proposed Algorithm

To solve the optimization problem in the context of Internet of vehicles, an improved estimation of distribution algorithm was provided. It is necessary to encode individual, the initialization of population, probability model for population update mechanism, local search operation, and adaptive adjustment mechanism of learning rate in the algorithm.

3.2.1 Baseline Algorithms

Based on the principle of statistical learning, estimation of distribution algorithm establishes a probability model according to the problem information to simulate the distribution of individuals in the solution space, then randomly samples the model to generate a new population, and uses the information of dominant individuals to update the model, so as to realize the evolution of the population (Perez-Rodriguez, R. 2022). The baseline framework of a common EDA is described as follows:

Step 1: Initialize the probability model.

Step 2: Generate a new population by sampling randomly based on probability model.

Step 3: Determine the dominant individuals in the population.

Step 4: Update the probability model by learning on the dominant individuals.

Step 5: End evolution according to termination conditions. Otherwise, go back to step 2.

3.2.2 Individual Coding Mode

Considering the priority relationship and the application for offloading decision in internet of vehicles, the PBIL (Population-based Incremental Learning, PBIL) was introduced and a matrix for vehicle task allocation matrix as the solution space were adopted. Assuming that there are n tasks nodes and K decisions for computing offloading decision in Internet of vehicles. Resource allocation matrix of individual can be defined as $R = (r_{ij})_{n \times K}$, in which $i \in [1, n]$, $j \in [1, K]$. The element $r_{ij} \in \{0, 1\}$, $\sum_{j=1}^K r_{ij} = 1$ and $r_{i0} = L[i]$ as the priority value of the task i . It is clear that a task can only acquire one resource to calculate. The value $r_{ij} = 1$ under condition $j \in [1, K]$ and $i \in [1, n]$ represents the task i acquired computing resource j .

3.2.3 Updating Mechanism of Probability Model

For a computing offloading decision problem in Internet of vehicles with n tasks nodes and K decisions, the corresponding probability matrix is defined as $P = (p_{ij})_{n \times K}$, $i \in [1, n]$, $j \in [1, K]$, $\sum_j p_{ij} = 1$ means that the probabilities sum of the task i acquired each resource allocation for calculating is 1 and $p_{ij} \in \{0, 1\}$ means that a probability of the task i acquired a resource allocation j to calculate. The updating process of probability matrix is designed, in which the model is described according to probability matrix updating process of single individual.

Suppose the probability matrix of an individual is $P(t) = (p_{ij}(t))_{n \times K}$, the corresponding computing offloading decision matrix is $R(t) = (r_{ij}(t))_{n \times K}$. For M dominant individuals selected by a strategy, the probability $p_{ij}(t+1)$ for the task i acquired a resource allocation j to calculate in generation $t+1$ is calculated as formula (10):

$$p_{ij}(t+1) = (1 - \alpha) p_{ij}(t) + \alpha \frac{\sum_k p_{ij}^k(t)}{M}, 0 < \alpha < 1 \quad (10)$$

where the variable $p_{ij}^k(t)$ presents a probability that the calculating resource j is allocated to the task i in the k -th individual for M dominant individuals, the express $\sum_k p_{ij}^k(t)$ presents the count of individual that the calculating resource j is allocated to the task i . The variable $\alpha \in \{0, 1\}$ is a learning rate.

The rate of learning is directly proportional to the evolution rate. If learning rate is large, the adjustment range of probability will be large, which results in weak ability in fine search. On the contrary, the adjustment range of probability model will be small, which results in insufficient global search ability. It is an optimal adjustment strategy for learning rate that the rapid global search in condition of large learning rate in early stage of evolution and the increasing detailed search with the population evolution. Therefore, an adaptive adjustment strategy for learning rate was proposed as formula (11):

$$\alpha(t+1) = \frac{T_0}{\lg(1+t)} \alpha(t) \quad (11)$$

3.2.4 Local Search Strategy

The advantage of the distribution estimation algorithm is to search solution space with the probability model established from the distribution of feasible solutions, by which the new individuals can be generated. However, the local optimal solution information is not utilized fully and there is no a mechanism to directly control the local optimal solution information. In order to enhance the local chemotaxis search ability, a specific local search operation for computing offloading decision of vehicles is given. According to the computing offloading decision, the local search operation is carried out by exchanging neighborhood structure and inserting neighborhood structure based on the key offloading decision with the largest consumption contained latency and energy consumption.

1. **Crossing operation:** For all tasks, exchanges the tasks i and j on processing resources in pairs. If the sum of consumption contained latency and energy of tasks i and j is smaller than before exchange and the global consumption is not increase, then the exchange is valid and accepted. The pseudo code of exchange search is as Algorithm 1.

Algorithm 1. Crossing operation

```
1) for (i = 1; i ≤ n; i++)
2) for (j = i; j ≤ n; j++)
3) if ((f(i, ri) + f(j, rj) > f(i, rj) + f(j, ri)) && (f' ≤ f))
4) ri ← j;
5) rj ← i;
6) break;
7) endif
8) endfor
9) endfor
```

Here the express $f(i, r_i)$ represents that the consumption of task i is offloaded and execution on computing resource r_i .

- Mutation operation:** For the key offloading decision with the largest consumption contained latency and energy, an operation was presented to move the tasks from one computing resource to another computing resource in sequence. It can be accepted if the global consumption f is reduced. The pseudo code inserted in the search is as Algorithm 2.

Algorithm 2. Mutation operation

```
1) times= 0;
2) while (times < LStime) // The depth LStime of local search is set to be 40.
3) for (j = arg(rj = Cmax))
4) times++;
5) r = argmin(f(j, r(i)) + f(i, r(i))); // i = 1, 2, ..., r; Select a resource with the least consumes.
6) if (Cr + f(j, i) < Cmax)
7) mj = i*;
8) calculate Cmax;
9) endif
10) endfor
11) end
```

3.2.5 The Framework of Proposed Algorithm

In view of the above design, the framework of adaptive distribution estimation algorithm (AEDA) for solving computing offloading decision in internet of vehicles is given as algorithm 3. Compared with the flow of traditional distribution estimation algorithm, AEDA increases the local search strategy and adaptively adjusts on learning rate of probability model, which helps to enhance the search ability of the algorithm. The framework of the proposed algorithm is given as Algorithm 3.

Algorithm 3. The algorithm framework of AEDA

Step 1: Input: Gen (Maximum number of generations), Pop_size (Population size), $Trunc_size = \xi * Pop_size$, ($0 < \xi < 1$, Truncation selection size), T_0 (initial temperature), $m_pop_size \leftarrow$ the number of m_pop ;
Step 2: initialization: $Pop \leftarrow$ Generate initial population (Pop_size) according to uniform distribution;
Step 3: Fitness evaluation (Pop) according to the formula (9);
Step 4: $opt_pop \leftarrow$ Truncation selection (Pop , $Trunc_size$);
Step 5: $c_sub_pop \leftarrow$ Crossing operation; $l(0 < p_m < 1)$
Step 6: $m_sub_pop \leftarrow$ Mutation operation;
Step 7: $opt_pop \leftarrow c_sub_pop \cup m_sub_pop$
Step 8: $P(t+1) \leftarrow$ Updating probability matrix P according to formula (10) and (11);
Step 9: $Sub_pop \leftarrow$ Sampling operation (P);
Step 10: $elite \leftarrow$ the fittest individual;
Step 11: If condition is not satisfied, then go to Step3;
Step 12: Output: $elite$.

In AEDA, fitness function f described by formula (9) was employed, which should be minimized. During each generation, Truncation selection is used to select the parent population (Pop), and then local sampling strategy is executed to produce crossing individuals (c_sub_pop) and mutation individuals (m_sub_pop). Next, the probability model matrix based on the combination of optima child population ($opt_pop - m_pop$) and local sampling population (c_sub_pop and m_sub_pop) is constructed and the final new individual is generated by the genetic sampling operation.

4. EXPERIMENTAL RESULTS AND ANALYSIS

In order to verify the performance of the proposed algorithm, some experiments were implemented and the task offloading scenario under the Internet of Vehicles were simulated on Matlab r2020a simulation platform. The experimental result based on GA (B. He, et al. 2014) and PSO (P. D. Diamantoulakis, et al. 2015) compare with AEDA on computing offloading decision in Internet of Vehicles were presented. All the algorithms run 30 independent times on the test problems with setting pop to 100 and maximum number of evaluations to 5×10^4 . The parameter settings are listed in Table 1.

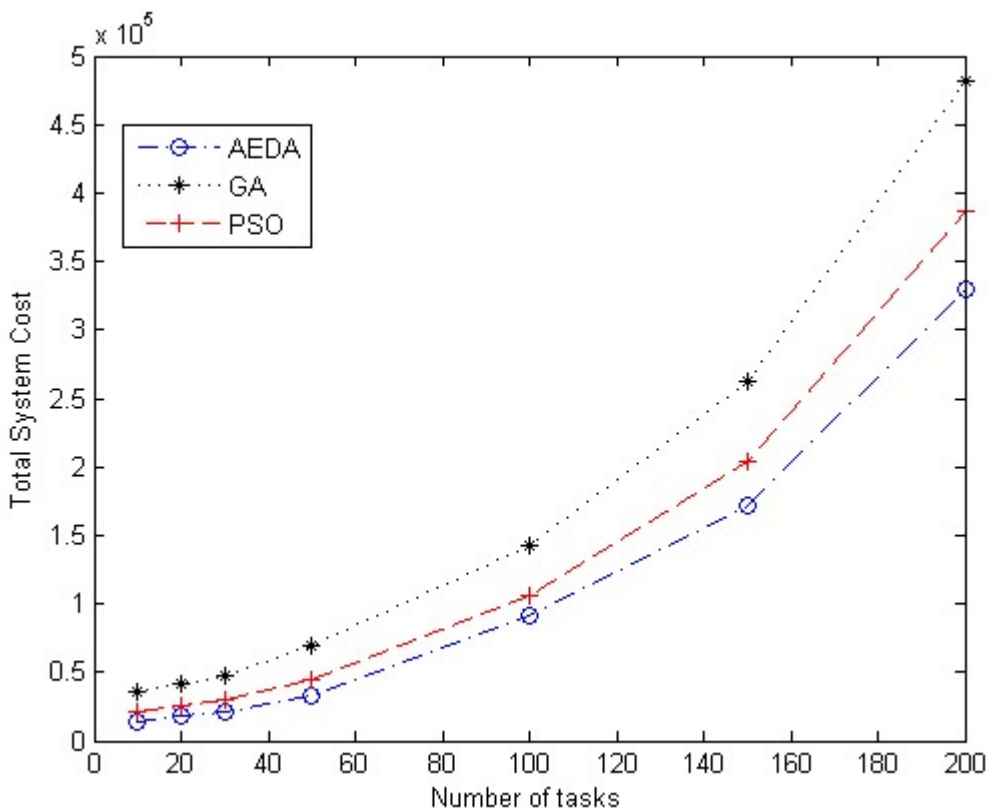
Table 1. Task offloading simulation parameters

Parameters	Value
Computing ability of on-board unit f_v	1 GHz
Computing ability of edge server f_m	4 GHz
Data volume of task c_i	40 kb
energy consumption power of on-board unit CPU p_{cpu}	80mW
Channel bandwidth of V2R w_r	100
Channel bandwidth of V2V w_v	75
Gaussian noise density in communication channel ρ_0	3×10^{-13}
Vehicle transmitting power P_u	0.2 W
Vehicle receiving power P_d	0.15W
Vehicle calculated power p_o	0.5W
Fading factor of path consumption constant in uplink channel p_1	4
Path loss factor p_2	2

4.1 Impact on the Number of Tasks

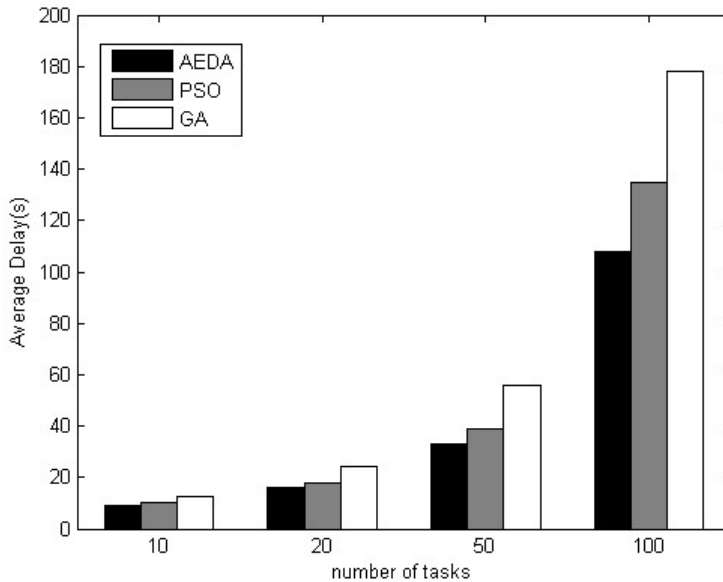
In order to study the influence on the global optimal fitness value for the change of the number of tasks, the number of tasks is taken as 10, 20, 30, 50, 100, 150 and 200 respectively and the maximum number of evaluations is set to 1.5×10^5 , $w_t = 0.5$ and $w_e = 0.5$. The simulation results are shown in Figure 3. It is clear that all trends of fitness change are decreasing for offloading strategies of AEDA, GA and PSO algorithm with the increase of the number of tasks which led to the increase of the energy consumption and the latency during offloading and execution. Compared with GA and PSO algorithms, AEDA algorithm has the lowest total system cost. When the number of tasks is 200, the total cost of AEDA algorithm is 86.6% of PSO and 67.3% of GA.

Figure 3. Impact of changes in the number of tasks



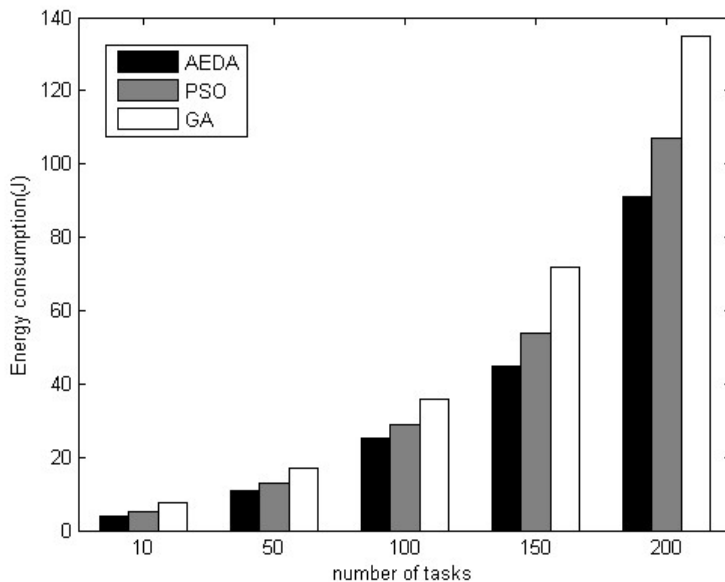
In order to further verify the effectiveness of the proposed task unloading strategy, the comparison of time cost of task offloading by AEDA, PSO and GA were implemented under the condition of different number of tasks and the simulation results was shown in figure 4. When the number of tasks is 10, 20, 50 and 100, the time cost of task offloading using AEDA is the smallest among the three algorithms. The computing offloading based on proposed adaptive estimate of distribution algorithm can get advantages in optimizing latency comparing with PSO and GA.

Figure 4. Comparison of total time cost among the AEDA, PSO and GA



The energy consumption of AEDA, PSO and GA for task offloading under different task quantities is shown in figure 5. With the increase of the number of tasks, the energy consumption of AEDA algorithm is gradually less than that of PSO and GA. The proposed computing offloading based on proposed adaptive estimate of distribution algorithm is more suitable for multi-user and multi-task offloading scenarios.

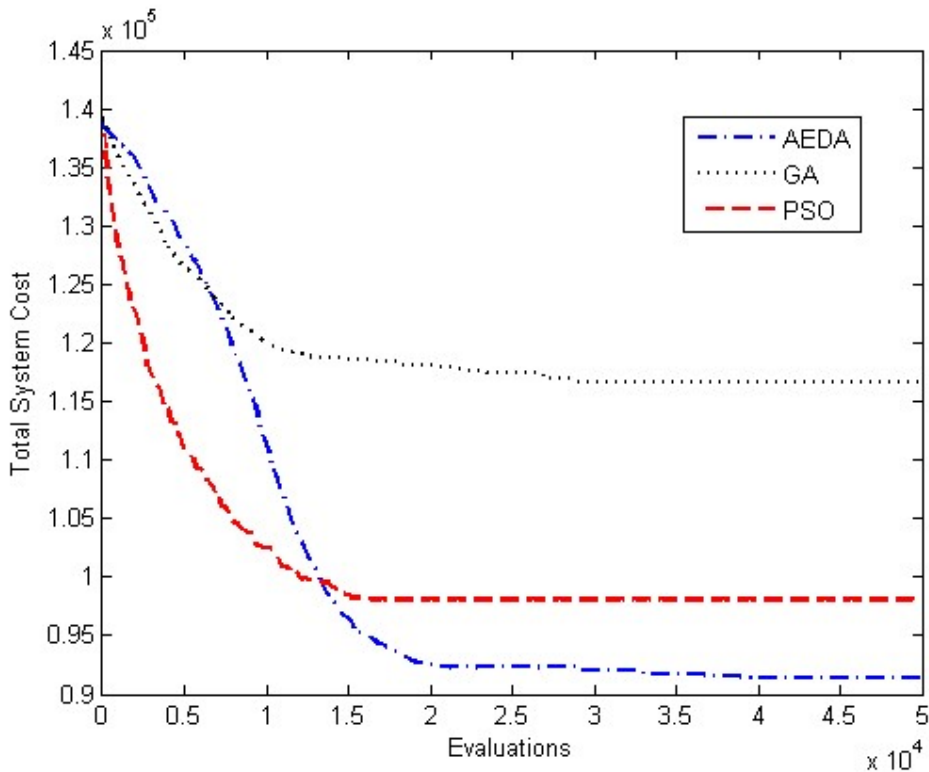
Figure 5. Comparison of total energy consumption among AEDA, PSO and GA



4.2 Influence of Iteration Number

In order to study the influence of the number of evaluations on the total system cost, the experiment on impact of changes in the evaluations were implemented and the simulation results were shown in figure 6, in which the maximum number of evaluations 5×10^4 , the total number of tasks $n = 100$, $w_l = 0.5$ and $w_e = 0.5$ and the experiment results are shown in Figure 6.

Figure 6. Impact of changes in the evaluations

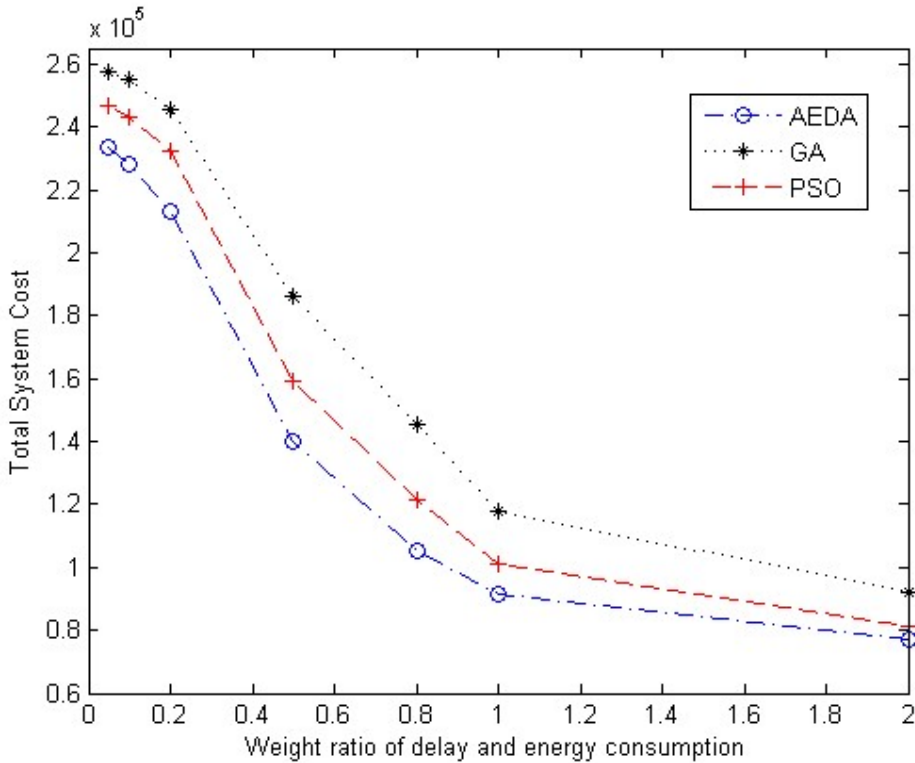


It can be seen that the total system cost obtained by AEDA, GA and PSO algorithms were decreasing with the increase of the evolution, which showed that the algorithms were convergence and get the optimal solution when the number of evaluations is greater than or equal to 2×10^4 . The computing offloading based on adaptive estimate of distribution algorithm can get the minimum total system cost than PSO and GA.

4.3 Influence Analyses for Weight Ratio of Latency and Energy Consumption

To compare the optimization performance of AEDA, GA and PSO algorithms under different weight ratio of latency and energy consumption w_l/w_e , the experiment was carried out and the simulation results were shown in Figure 7, in which the maximum number of evaluations was set to 5×10^4 , the total number of tasks $n = 100$, and the weight ratio of latency and energy consumption w_l/w_e was set to 0.05, 0.1, 0.2, 0.5, 0.8, 1 and 2 respectively.

Figure 7. Influence analyses under different weight ratio w_l/w_e

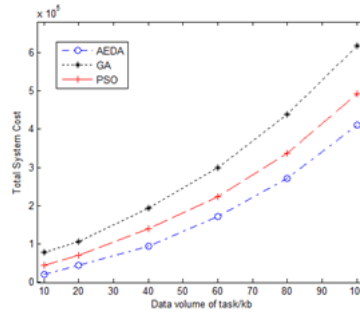


It can be seen that with the increase of weight ratio of latency and energy consumption w_l/w_e , the total system cost obtained by AEDA, GA and PSO algorithms were decrease. For different weight ratio of latency and energy consumption w_l/w_e , the total system cost obtained by AEDA algorithm is always greater than that obtained by GA and PSO. The total system cost obtained by AEDA algorithm is the smallest among the algorithms AEDA, GA and PSO. The adaptive estimate of distribution algorithm and task offloading model can get an optimization solution in different scenarios (such as the reality of low battery power) and can be applied to a variety of real demand scenarios.

4.4 Influence of Data Volume of Task

In order to study the influence on the global optimal fitness value for the change of data volume of task c_i , the experiment was carried out and the simulation results were shown in Figure 8, in which the maximum number of evaluations was set to 5×10^4 , the total number of tasks $n = 100$, $w_l/w_e = 1$ and the data volume of task $c_i = 10, 20, 40, 60, 80, 100$ respectively.

Figure 8. Influence analyses under different data volume of task c_i



It is clear that all trends of total system cost were increasing greatly for offloading strategies of AEDA, GA and PSO algorithm with the increase of the data volume of task c_i . The reason is that the increase of the data volume of computing task will lead to more energy consuming and longer time in the computing process. Compared with GA and PSO algorithms, AEDA algorithm acquired the lowest total system cost. The improved estimation of distribution algorithm is more suitable for multi-user and multi-task offloading scenarios in Internet of vehicles.

5. CONCLUSION

In order to meet the demand of on-board applications with a lower computing latency and solve the problem of insufficient computing resources of on-board units in 5G era, a computing offloading decision based on adaptive estimation of distribution algorithm with a local search strategy and an adaptive learning rate according to the characteristics of the task offloading in Internet of vehicles was proposed, in which an adaptive adjustment mechanism of learning rate was introduced to dynamically adjust the search ability of the algorithm and a crossing operation was given to balance the ability of global search and local search. Considering the sequence of task execution and the relationship between tasks, a task offloading priority matrix was designed. Compared with the experimental results on AEDA, GA and PSO for computing offloading, the proposed AEDA algorithm has obvious advantages and can be suitable for a variety of real scenarios. The impact of the vehicle movement on experiment and optima algorithm based on multi-objective optimization algorithm for offloading problem will be a work in future.

ACKNOWLEDGMENT

This research was supported by the School Scientific Foundation under Grander No. SKQD2021B-035, Innovative industrial design and Research Institute Project under Grander No. cxgy21-004, Higher education collaborative education Foundation of the Education Ministry under Grander No. 202101347006 and 202101355031, the National Natural Science Foundation of China under Grant No. U1801266, Guangdong University Innovation Team Project under Grant No. 2020KCXTD045, Guangdong Education Science Planning Project under Grant No. 2021GXJK515.

REFERENCES

- Chen, Y., Zheng, W., Li, W., & Huang, Y. (2021). Large group activity security risk assessment and risk early warning based on random forest algorithm. *Pattern Recognition Letters*, 144(4), 1–5. doi:10.1016/j.patrec.2021.01.008
- Cheng, J., & Guan, D. (2021). Research on task-offloading decision mechanism in mobile edge computing-based Internet of Vehicle. *J Wireless Com Network*, 11(7), 136–148.
- Dai, Y., Xu, D., Maharjan, S., Qiao, G., & Zhang, Y. (2020). Artificial intelligence empowered edge computing and caching for internet of vehicles. *IEEE Wireless Communications*, 26(3), 12–18. doi:10.1109/MWC.2019.1800411
- Dai, Y., Xu, D., Maharjan, S., & Zhang, Y. (2021). Joint load balancing and offloading in vehicular edge computing and networks. *IEEE Internet of Things Journal*, 6(3), 4377–4387. doi:10.1109/JIOT.2018.2876298
- Diamantoulakis, P. D., Pappi, K. N., Karagiannidis, G. K., Xing, H., & Nallanathan, A. (2015). Joint downlink/uplink design for wireless powered networks with interference. *IEEE Access: Practical Innovations, Open Solutions*, 5(3), 1534–1547.
- Dinh, T. Q., Tang, J., La, Q. D., & Quek, T. Q. (2019). Offloading in mobile edge computing: Task allocation and computational frequency scaling. *IEEE Transactions on Communications*, 65(8), 3571–3584.
- He, B., Liu, A., Yang, N., & Lau, V. K. (2014). On the design of secure non-orthogonal multiple access systems. *IEEE Journal on Selected Areas in Communications*, 35(10), 2196–2206.
- He, Y., Zhao, N., & Yin, H. (2017). Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach. *IEEE Transactions on Vehicular Technology*, 67(1), 44–55. doi:10.1109/TVT.2017.2760281
- Liu, L., Chen, C., Pei, Q., Maharjan, S., & Zhang, Y. (2020). Vehicular edge computing and networking: A survey. *Mobile Networks and Applications*, 9(4), 1318–1327.
- Liu, L., Lang, L., Hu, B., Shi, L., Huang, B., & Zhao, Y. (2021). Emission reduction decision of agricultural supply chain considering carbon tax and investment cooperation. *Journal of Cleaner Production*, 9(4), 126–1305. doi:10.1016/j.jclepro.2021.126305
- Luo, Q., Li, C., Luan, T. H., Shi, W., & Wu, W. (n.d.). Self-Learning Based Computation Offloading for Internet of Vehicles: Model and Algorithm. *IEEE Transactions on Wireless Communications*, 20(9), 5913–5925.
- Lv, S., & Liu, Y. (2021). PLVA: Privacy-preserving and lightweight V2I authentication protocol. *IEEE Transactions on Intelligent Transportation Systems*, 4(9), 427–436.
- Mu, L., Ge, B., Xia, C., & Wu, C. (2022). Multi-Task Offloading Based on Optimal Stopping Theory in Edge Computing Empowered Internet of Vehicles. *Entropy (Basel, Switzerland)*, 24(1), 814–826. doi:10.3390/e24060814 PMID:35741535
- Ni, T., Chang, H., Song, T., Xu, Q., Huang, Z., Liang, H., Yan, A., & Wen, X. (2019). A. Yan, X. Wen, Non-intrusive online distributed pulse shrinking based interconnect testing in 2.5 D IC. *IEEE Transactions on Circuits and Wystems. II, Express Briefs*, 67(11), 2657–2661. doi:10.1109/TCSII.2019.2962824
- Ni, T., Yao, Y., Chang, H., Lu, L., Liang, H., Yan, A., Huang, Z., & Wen, X. (2019). LCHR-TSV: Novel low cost and highly repairable honeycomb-based TSV redundancy architecture for clustered faults. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(10), 2938–2951. doi:10.1109/TCAD.2019.2946243
- Ning, Z., Huang, J., Wang, X., Rodrigues, J. J. P. C., & Guo, L. (2019). Mobile edge computing-enabled internet of vehicles: Toward energy-efficient scheduling. *IEEE Network*, 33(5), 198–205. doi:10.1109/MNET.2019.1800309
- Peng, H., Ye, Q., & Shen, X. (2020). Spectrum management for multiaccess edge computing in autonomous vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(7), 3001–3012.
- Perez-Rodriguez, R. (2022). An estimation of distribution algorithm for combinatorial optimization problems. *International Journal of Industrial Optimization*, 3(1), 47–67. doi:10.12928/ijio.v3i1.5862

Pliatsios, D., Sarigiannidis, P., & Lagkas, T. (2022). Joint Wireless Resource and Computation Offloading Optimization for Energy Efficient Internet of Vehicles. *IEEE Transactions on Green Communications and Networking*, doi:10.1109/TGCN.2022.3189413

Sharma, S., & Kaushik, B. (2019). A survey on internet of vehicles: Applications, security issues & solutions. *Vehicular Communications*, 20(9), 107–112. doi:10.1016/j.vehcom.2019.100182

Shen, X., Chang, Z., & Niu, S. (2022). Mobile Edge Computing Task Offloading Strategy Based on Parking Cooperation in the Internet of Vehicles. *Sensors (Basel)*, 22(2), 49–59. doi:10.3390/s22134959 PMID:35808452

Song, Y., Jing, X., & Feng, Y. (2021). Secure and energy efficient offloading of mobile edge computing in the Internet of vehicles. *Journal of Tsinghua University*, 14(6), 1246–1253.

Wang, L., Wang, S. Y., & Fang, C. (2021). A hybrid distribution estimation algorithm for solving multidimensional knapsack problem. *Control and Decision*, 26(8), 1121–1125.

Wang, T., Luo, X., & Zhao, W. (2022). Improving the performance of tasks offloading for internet of vehicles via deep reinforcement learning methods. *IET Communications*, Vol, 16(10), 1230–1240. doi:10.1049/cmu2.12334

Xiong, Z., Xiao, N., Xu, F., Zhang, X., Xu, Q., Zhang, K., & Ye, C. (2020). An equivalent exchange-based data forwarding incentive scheme for socially aware networks. *J. Signal Process. Syst.*, 93(11), 1–15.

Xue, X., Zhang, K., Tan, K. C., Feng, L., Wang, J., Chen, G., Zhao, X., Zhang, L., & Yao, J. (2020). Affine transformation enhanced multifactorial optimization for heterogeneous problems. *IEEE Transactions on Cybernetics*, 6(4), 723–734. PMID:33320820

Yao, L., Xu, X., Bilal, M., & Wang, H. (2022). Dynamic Edge Computation Offloading for Internet of Vehicles With Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, 9(4), 762–774. doi:10.1109/TITS.2022.3178759

Zhang, D., Cao, L., Zhu, H., Zhang, T., Du, J., & Jiang, K. (2022). Task offloading method of edge computing in internet of vehicles based on deep reinforcement learning. *Cluster Computing*, 4(5), 1175–1187. doi:10.1007/s10586-021-03532-9

Zhang, K., Mao, Y., Leng, S., He, Y., & Zhang, Y. (2020). Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading. *IEEE Vehicular Technology Magazine*, 12(2), 36–44. doi:10.1109/MVT.2017.2668838

Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., Pan, L., Maharjan, S., & Zhang, Y. (2019). Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks. *IEEE Access: Practical Innovations, Open Solutions*, 4(2), 5896–5907.

Zhang, Q., Gui, L., Hou, F., Chen, J., Zhu, S., & Tian, F. (2020). Dynamic task offloading and resource allocation for mobile-edge computing in dense cloud RAN. *IEEE Internet of Things Journal*, 7(4), 3282–3299. doi:10.1109/JIOT.2020.2967502

Zhao, J., Liu, J., Jiang, J., & Gao, F. (2020). Efficient deployment with geometric analysis for mm Wave UAV communications. *IEEE Wireless Communications Letters*, 9(7), 1115–1119.

Fahong Yu received his PhD in Computer Science and Technology from Wuhan University, China, in 2012. In 2012 he was hired as the assistant professor at Jiaxing University and is currently a vice-professor for Shanwei Polytechnic and a guest professor for Yangtze University. He writes and presents widely on issues of intelligent computing.

Meijia Chen received her Master's degrees from Huazhong Normal University in 2007. At the same year, she worked as an assistant Professor for Jiaxing University. She is currently an assistant Professor for Shanwei Polytechnic. Her current research focuses on intelligent computing and privacy protection. She writes and presents widely on issues of intelligent computing.

Bolin Yu received his PhD in Electronic Engineering from Huazhong University of Science and Technology, China, in 2007. He is currently a vice-professor for Shenzhen Institute of Information Technology, China. His current research focuses on intelligent computing and machine learning.