ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20221207032927
Preliminary communication

A Novel Task of Loading and Computing Resource Scheduling Strategy in Internet of Vehicles Based on Dynamic Greedy Algorithm

Huiyong LI*, Shuhe HAN, Xiaofeng WU, Furong WANG

Abstract: Focus on the scheduling problem of distributed computing tasks in Internet of Vehicles. Firstly, based on the computing-aware network theory, a distributed computing resource model of the Internet of Vehicles is established, and the seven-dimensional QoS attributes of the computing resources in the Internet of Vehicles (reliability between computing resources, communication costs, computing speed and computing costs of the computing resources themselves , computing energy consumption, computing stability, and computing success rate) are grouped and transformed into two-dimensional comprehensive attribute priorities: computing performance priority and communication performance priority. Secondly, the weighted directed acyclic graph model of distributed computing tasks in the Internet of Vehicles and the seven-dimensional QoS attribute weighted undirected topology graph model of distributed computing resources in the Internet of Vehicles are respectively established. Moreover, a dynamic greedy algorithm-based task of loading and computing resource scheduling algorithm is proposed. Finally, the example analysis shows that the overall performance of this dynamic greedy algorithm-based task of loading and computing resource scheduling algorithm is better than the classic HEFT scheduling algorithm and round robin scheduling algorithm.

Keywords: computing-aware networks; edge computing; fog computing; greedy algorithm; internet of vehicles; scheduling algorithm

1 INTRODUCTION

In recent years, Internet of Vehicles which integrates technologies such as the Internet of Things, Big Data, Mobile Computing, and AI (Artificial Intelligence), has continued to develop [1-5]. Based on vehicle wireless communication technology V2X (Vehicle to Everything), the Internet of Vehicles can connect computing resources such as on-board computing units, roadside computing units and cloud computing centers [6, 7]. In the Internet of Vehicles, service providers can obtain user service needs and road environment information, and provide vehicle users with various information services such as automatic driving, route planning, collision warning, and vehicle entertainment based on these data. These services are generally completed in the Internet of Vehicles through distributed computing technology.

The Internet of Vehicles is a typical mobile Internet of Things. The Internet of Vehicles has the characteristics of limited communication bandwidth, unstable network connection, dynamic network topology changes, and heterogeneous distributed computing resources. The new generation of information network is changing from a network infrastructure centered on information transmission to an intelligent cloud network infrastructure that integrates computing, storage, and transmission resources. The computing-aware networks is a new network architecture proposed to cope with this transformation [8]. Based on ubiquitous network connections, the computing-aware network interconnects dynamically distributed computing and storage resources. Through the unified collaborative interconnection of multi-dimensional resources such as network, storage, and computing resources, massive applications can call ubiquitous distributions on demand and in real time. Computing resources to achieve global optimization of connections and computing in the network, provide a consistent user experience [9].

The computing-aware networks organically integrate technologies such as fog computing [10], edge computing [11], and sea computing [12]. In the Internet of Vehicles system, the on-board computing unit is regarded as a sea computing resource, and the roadside computing unit is regarded as a fog computing resource and an edge computing resource. These computing resources and background cloud computing resources are unified in the computing-aware network framework model. Such a unified computing resource model can significantly improve the management efficiency of various settlement resources in the Internet of Vehicles.

However, how to effectively perform distributed task scheduling in the Internet of Vehicles is still an urgent problem to be solved. Due to the characteristics of heterogeneity and mobility in the distributed computing resources in the Internet of Vehicles, its structure and performance are very different from those of the distributed computing resources on the Internet. Among the attributes of computing resources, attributes such as reliability between computing resources, communication costs, computing costs, computing energy consumption, computing stability, and computing success rate are key factors.

The problem of task allocation and resource scheduling in the Internet of Vehicles is an important distributed mobile computing problem. How to design a reasonable and effective distributed computing resource scheduling algorithm is the key to solving this problem. There have been many research results on resource scheduling algorithms for distributed systems. These algorithms can be divided into single-objective optimal scheduling algorithms and multi-objective optimal scheduling algorithms according to the number of computing resource scheduling targets. The single-objective optimal scheduling algorithm is faster than the multi-objective optimal scheduling algorithm. According to the classification of the relationship between distributed computing tasks, it can be divided into meta-task scheduling problems and dependent task scheduling problems. Since there is data dependence between computing tasks in the distributed computing problem in the Internet of Vehicles, and computing resources in the Internet of Vehicles have multi-dimensional QoS attributes, the distributed resource scheduling algorithm for distributed computing tasks in the Internet of Vehicles is task-dependent multi-objective scheduling algorithm.

In distributed computing research, some only focus on the timing performance of scheduling algorithms. In the related research of multi-objective scheduling problems, many scholars use intelligent optimization algorithms such as GA (Genetic Algorithm) [13, 14] and PSO (Particle Swarm Optimization algorithm) [15, 16] to directly solve the Pareto solution set of multi-objective problems and finally solve them according to the preference relationship. However, such scheduling methods generally assume that distributed tasks are independent meta-tasks.

The existing dependent task scheduling algorithms mainly include: list scheduling algorithm, cluster scheduling algorithm and replication-based scheduling. Among them, the list scheduling algorithm is the one with the highest time efficiency and is widely used. Common list scheduling algorithms include: MCP (Modified Critical Path), ETF (Earliest Time First), and DLS (Dynamic Level Scheduling). However, these algorithms are often single-objective and assume homogeneous computing resources. HEFT (Heterogeneous Earliest Finish Time) is a classic scheduling algorithm for heterogeneous computing resources, which has been widely recognized [17]. The task priority list construction method of the classic HEFT scheduling algorithm is: the priority rank value of each task is the longest path from the task to the exit task. Arrange the tasks in the non-increasing order of the task priority rank values. If the rank values are equal, arrange these nodes randomly. Greedy algorithm is an approximation method to solve the optimization problem. In the greedy algorithm, the optimal decision-making is gradually constructed [18, 19].

Focusing on the computing resource scheduling problem of distributed computing tasks in the Internet of Vehicles, we propose a task allocation and computing resource scheduling algorithm in the Internet of Vehicles based on a dynamic greedy algorithm. Compared with the existing distributed task scheduling algorithm for Internet of Vehicles, the following two improvements are mainly made: 1. Based on the computing-aware network theory, a distributed computing resource model of the Internet of Vehicles is established, and the seven-dimensional QoS attributes of computing resources in the Internet of Vehicles (reliability between computing resources, communication costs, computing speed of the computing resources themselves, computing Cost, calculation energy consumption, calculation stability and calculation success rate) are grouped and converted into two-dimensional comprehensive attribute priorities: computing performance priority and communication performance priority. 2. A task queue sorting algorithm based on dynamic greedy algorithm is proposed. This algorithm modifies the HEFT scheduling algorithm "Arrange tasks in non-increasing order of task priority rank value, if the rank value is equal, then randomly arrange these nodes". Tasks with equal rank values are grouped according to the weight of the successor nodes, so that tasks of the same successor can be allocated to adjacent computing resources as much as possible. At the same time, small tasks are guaranteed to be executed first, so that more tasks can be executed in the same computing resource, and reducing the overhead of transferring data between computing resources. This algorithm ensures that the computing resource selected each time is the local optimum, thus achieving the goal of global optimization.

2 COMPUTING RESOURCE MODEL OF INTERNET OF VEHICLES BASED ON COMPUTING-AWARE NETWORKS

2.1 Distributed Computing Task Scheduling Model in Internet of Vehicles

Distributed computing task scheduling model in Internet of Vehicles: In the $M \times N$ distributed computing resource scheduling model consisting of m tasks and n computing resources. The distributed task scheduling problem can be described as a quadruple, that is, $Dis = (T, R, O, \Theta)$. Among them, T is a task set composed of m tasks, R is a resource collection composed of n computing resources, O represents the scheduling optimization objective function of the scheduling system, Θ represents the scheduling algorithm. Next, establish task model T and resource model R.

2.2 Task Model in Internet of Vehicles

In graph theory, if a directed graph cannot start from a certain vertex and return to that point through several edges, then the graph is a directed acyclic graph (DAG graph). The weighted DAG (directed acyclic graph) model is a common task model. Therefore, we build a weighted DAG model for the distributed processing task of the Internet of Vehicles.

Let the distributed processing task of the Internet of Vehicles be $T = (TASK, p)$. Among them: $TASK = (task_1, task_2, \ldots, task_n)$ is a set of executable tasks, p is a partial order relationship on $TASK$, which is used to illustrate the priority relationship between tasks. That is: if $task_1$, $ptask_2$, it means that $task_1$ must be executed before $task_2$ is executed.

Among them, T can be expressed as a weighted task DAG model $TG = (V, E, W)$, where V is a node set, E is a directed edge set, and W is a node weight set. A node v represents a task in the graph. A directed edge is represented by a node pair $e_ij = (v_i, v_j)$, v_i is called the parent node, and v_j is called the child node. A node without a parent is called an entry, and a node without children is called an exit. The directed edge e_{ij} represents the data dependency relationship between tasks (that is, the partial order relationship). The weight $w(v)$ of a node represents the calculation amount of the task represented by the node. The predecessor task set $pred(v)$ represents the direct predecessor task set of the task represented by node v, and the successor task set $succ(v)$ represents the direct successor task set of the task represented by node v.

Example 1: The distributed processing task T_1 of a certain Internet of Vehicles can be represented by the weighted task DAG model TG_1 in the figure below. In this instance, the data of the task represented by node v_6 depends on nodes v_1, v_2 and v_3. The task represented by node v_7 depends on node v_4 for data. For the task represented by node v_8, the data depends on nodes v_3 and v_5. The data of the task represented by node v_9 depends on nodes v_6, v_7 and v_8.

From the Fig. 1, we can see: $w(v_1) = 500$, $w(v_2) = 1300$, $w(v_3) = 700$, $w(v_4) = 2300$, $w(v_5) = 1500$, $w(v_6) = 2700$, $w(v_7) = 1400$, $w(v_8) = 3100$, $w(v_9) = 1900$; $succ(v_1) = v_6$; $succ(v_2) = v_6$; $succ(v_3) = \{v_6, v_8\}$; $succ(v_4) = v_7$; $succ(v_5) = v_8$; $succ(v_6) = v_9$, $pred(v_6) = \{v_1, v_2, v_3\}$; $succ(v_7) = v_9$, $pred(v_7) = v_4$; $succ(v_8) = v_9$, $pred(v_8) = \{v_3, v_5\}$; $pred(v_9) = \{v_6, v_8, v_9\}$.

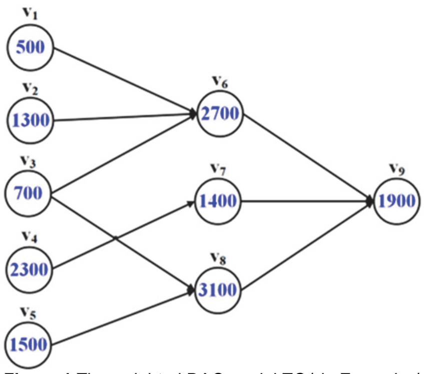

Figure 1 The weighted DAG model TG1 in Example 1

2.3 Reliable Attribute Model of Distributed Computing Resources in Internet of Vehicles

In order to allocate distributed computing resources more effectively and reasonably, it is necessary to quantify and compare each computing resource in the distributed computing resource pool. Internet distributed computing resources have the characteristics of isomorphism and stable communication environment. However, the distributed computing resources of the Internet of Vehicles are heterogeneous and the communication environment is unstable. As can be seen from Fig. 2, in the Internet of Vehicles environment, computing resources can be divided into three categories: background computing resources, roadside computing resources, and on-board computing resources. During the high-speed operation of the vehicle, since the communication mode between the on-board computing resources and other computing resources is mainly wireless communication, its communication quality is affected by the dynamics of the communication environment. Therefore, how to describe and measure the changes in the reliability of data transmission between sub-computing resources in the distributed computing environment of the Internet of Vehicles in the communication environment is a very important issue. We refer to this problem as a reliable property among sub-computing resources of Internet of Vehicles. The model and measurement method of this property are given below.

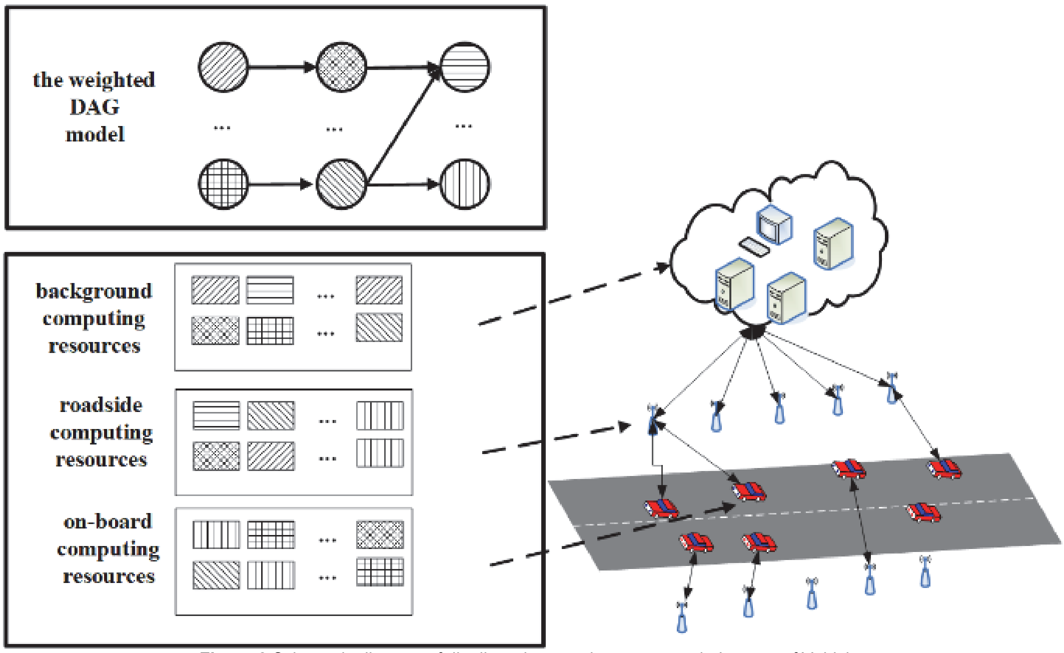

Figure 2 Schematic diagram of distributed computing resources in Internet of Vehicles

Definition 1 In the Internet of Vehicles environment, computing resource A transmits data M to sub-computing resource B, then the probability that B correctly receives M is the dependability from computing resource A to computing resource B, denoted as $De_{A\text{-}B}$.

The calculation method of $De_{A\text{-}B}$ is discussed in two cases:

Case One: When sub-computing resource A directly sends data to sub-computing resource B without passing through other routes.

1. When sub-computing resource A communicates with sub-computing resource B through memory, data bus or wired communication, $De_{A\text{-}B} = 1$.

2. When sub-computing resource A is not connected to sub-computing resource B, $De_{A\text{-}B} = 0$.

3. When the communication mode between sub-computing resource A and sub-computing resource B is wireless communication, let the communication distance be d, the communication radius be R, and the path loss coefficient be a ($2 < a < 6$), then:

$$De_{A-B} = \begin{cases} 1-0.5\left(d/R\right)^{2a}, d < R \\ 0.5\left(2-d/R\right)^{2a}, \mathrm{R} \le d < 2R \\ 0, d \ge 2R \end{cases} \quad (1)$$

Case two: Sub-computing resource A indirectly sends data to sub-computing resource B, which is forwarded by n routing nodes in the middle. If the reliability between the i-th routing node and the i + 1-th routing node is $De_{Ni\text{-}Ni+1}$, then:

$$De_{A-B} = De_{A-Node_1} \times De_{Node_i - Node_{i+1}} \quad (2)$$

Example 2: Assume that in the Internet of Vehicles environment at a certain moment, there are four idle sub-computing resources to choose from among the distributed computing resources connected to sub-computing resource 29: computing resource 285, computing resource 1407, computing resource 19 and computing resource 25. The network topology among these sub-computing resources is shown in Fig. 3.

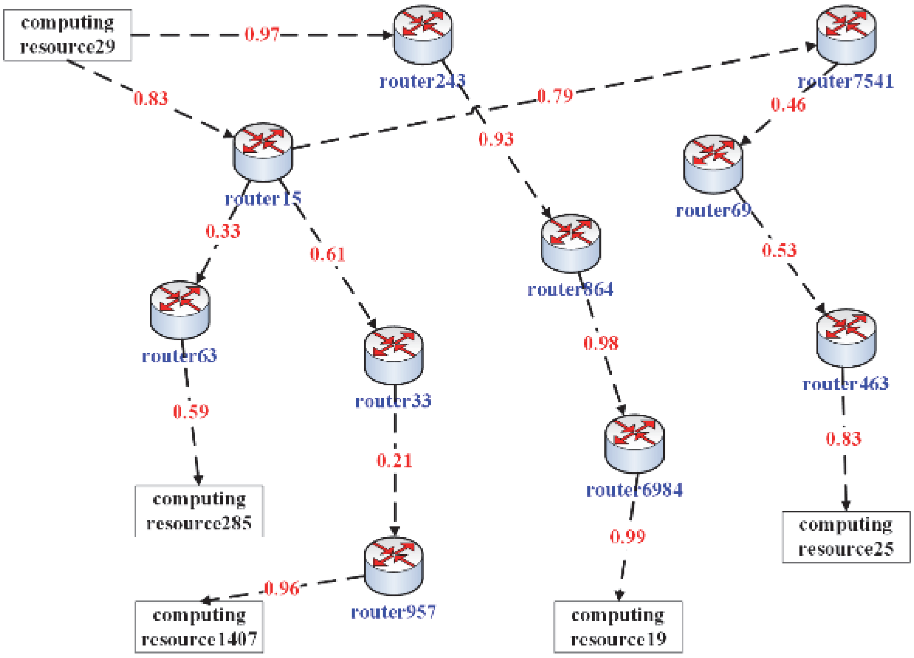

Figure 3 Computing resource network topology in Example 2

In this example, assuming that the identification value of the connection line between network nodes is the reliability of the communication line (obtained according to Eq. (1), the reliability between sub-computing resources can be obtained according to Eq. (2):

$$\begin{aligned} &Pr_{29\to285} = 0.83\times0.33\times0.59 = 0.162, \\ &Pr_{29\to1407} = 0.83\times0.61\times0.21\times0.96 = 0.102, \\ &Pr_{29\to19} = 0.97\times0.93\times0.98\times0.99 = 0.875, \\ &Pr_{29\to25} = 0.83\times0.79\times0.46\times0.53\times0.83 = 0.13. \end{aligned} \quad (3)$$

It can be seen from the above calculation results that, among these candidate computing resources, the sub-computing resource 19 is the most reliable computing resource for the sub-computing resource 29.

2.4 Multi-dimensional QoS Internet of Vehicles Computing Resource Model

The common QoS attributes of distributed computing resources in the Internet of Vehicles include: communication mode, response task time, computing speed, computing cost, computing energy consumption, computing stability, computing success rate, communication time, etc. These QoS attributes of computing resources and the reliability attributes of computing resources measure the attributes of distributed computing resources from different perspectives, thus providing a basis for us to build a multi-dimensional QoS attribute distributed computing resource system for Internet of Vehicles. The topology graph model is a commonly used model to describe distributed computing resources. The weighted and undirected topology graph model is used below to describe the multi-dimensional QoS distributed computing resources in the Internet of Vehicles.

The multi-dimensional QoS distributed computing resources in the Internet of Vehicles can be expressed as a weighted undirected topology graph $RG = (V, \mathrm{E}, W_v, W_e)$, where V is the node set, E is the undirected edge set, and W_v is the node weight set, W_e is the set of edge weights.

In this model, node v represents a computing resource. The undirected edge e_{ij} indicates that the computing resource r_i and the computing resource r_j can communicate. Node weight $w_v(v)$=(speed, cost, energy, stability, success) is a five-dimensional vector, respectively representing the calculation speed, calculation cost, calculation energy consumption, calculation stability and calculation success rate attribute. The edge weight $w_e(e)$=(dependability, price) is a two-dimensional vector, respectively representing the reliability and communication price attributes between computing resources represented by the nodes at both ends of the edge. Among them, the calculation speed of the computing resource is the processor speed of the computing resource, its unit is megahertz, and its value range is speed ∈ [0, 1000]; the computing cost of the computing resource is the computing resource set by the computing resource owner. The price of resources, its value range is cost ∈ [0, 100]; the energy consumption of computing resources is the energy consumption power of this computing resource, its unit is watts, and its value range is energy ∈ [0, 100]; the calculation stability of a computing resource is the ratio of the effective working time of the computing resource to the total time (total time = effective working time + failure time), and its value range is stability ∈ [0, 1]. The success rate is the ratio of the number of tasks successfully calculated by the computing resource to the number of received tasks, and its value range is success ∈ [0, 1]; the definition of reliability between computing resources is as defined in Definition 1, and its value range is dependability ∈ [0, 1]; the communication price between computing resources is the communication price proposed by the communication service provider, its value range is price ∈ [0, 50].

Taking a typical physical scene of the Internet of Vehicles as the background, the weighted undirected topology diagram of the distributed computing resources in the physical scene is constructed as follows:

Example 3: When the car is driving, the time and space attributes of the car change, so the car in the Internet of Vehicles will generate data streams. In the urban Internet of Vehicles environment, the distributed computing mode is used to process the data stream generated by each car during driving. Assuming that a car Car611 in the Internet of Vehicles is driving on a city road, the distributed computing resources available to it are: 1. On-board computing resources: 29, 58, 33, 38, 89, 41; 2. Roadside computing Resources: 263, 613, 368, 417; 3. Computing Center computing resources: 1134 and 1519. These computing resources can communicate wirelessly through WiFi or 5G. In this physical scenario, multi-dimensional QoS distributed computing resources can be represented as a weighted undirected topology diagram as shown in Fig. 4.

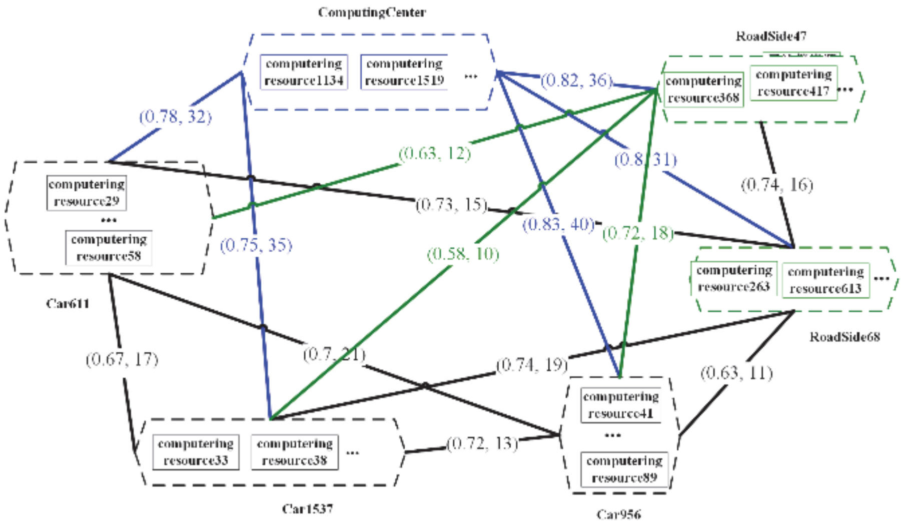

Figure 4 The weighted and undirected topology of computing resources in Example 3

In the figure above, the weights of undirected edges between various computing resources have been marked. Since various computing resources in the Internet of Vehicles are heterogeneous, their node weights are different, and their weights are shown in Tab. 1.

Table 1 Multidimensional QoS attribute values of various computing resources in Example 3

type	location	speed	cost	energy	stability	success
On-board computing resource	Car611	200	32	68	0.53	0.75
	Car1537	260	29	73	0.48	0.8
	Car956	180	36	65	0.57	0.79
roadside computing resource	Roadside68	300	67	35	0.74	0.9
	Roadside47	320	56	43	0.82	0.92
background computing resource	Computing Center	380	73	46	0.92	0.95

3 COMPUTING RESOURCE SCHEEDULING ALGORITHM BASED ON DYNAMIC GREEDY ALGORITHM

Based on the above distributed processing model of the Internet of Vehicles, we propose a computing resource scheduling algorithm based on dynamic greedy algorithm. This algorithm not only considers the heterogeneity of the computing resources of the Internet of Vehicles, but also considers the multi-dimensional communication cost between heterogeneous computing resources. The framework of the algorithm is as follows:

```
Algorithm 1 Computing Resource Scheduling Algorithm Based on Dynamic Greedy
input: Weighted task directed acyclic graph TG, weighted resource undirected topology graph RG
output: task scheduling sequence
Calculate the priority of each task;
Compute property values for each computing resource;
Arrange tasks according to priority and put them into queue TQ;
whileTQ is not empty do
    Take out the task at the head of the TQ queue and delete t in TQ;
    Allocate the optimal computing resources for processing tasks according to the dynamic greedy algorithm;
    Assign the task to the computing resource;
end
```

In the above algorithm, there are two core steps:

constructing a priority list of tasks and selecting appropriate computing resources for each task.

3.1 Algorithm for Constructing Priority List of Top-Level Minimal Successor Tasks

In the construction algorithm of the highest minimum successor task priority list, three parameters are firstly used:

1. The level b-level of the current node v: the longest path length from node v to the end node, denoted as $B(v)$.
2. The minimum weight of the immediate successor nodes of the current node v: the minimum weight of all the successor nodes of the node v, denoted as $S(v)$.
3. The weight of the current node v: $w(v)$.

The first Line of Algorithm 1 can be refined into Algorithm 2.

Algorithm 2 Priority List Construction Algorithm in Dynamic Greedy Scheduling Algorithm

input: Weighted Task Directed Acyclic Graph TG

output: task priority list

```
Calculate B(v), S(v), w(v) of each node in TG;
if B(v_i) ≠ B(v_j) then
    Arrange the nodes in a non-increasing order of B(v) values;
else if S(v_i) ≠ S(v_j) then
        Arrange the nodes in non-decreasing order of S(v) values;
    else if w(v_i) ≠ w(v_j) then
            Arrange the nodes in non-decreasing order of w(v) values;
        else
            Randomly arrange the nodes;
        end
    end
end
```

The idea of the minimum successor task priority algorithm at the highest level is that we group tasks with equal rank values according to the weights of successor nodes, to ensure that tasks of the same successor are allocated to adjacent computing resources as much as possible. At the same time, small tasks are guaranteed to be executed first, so that more tasks can be executed in the same computing resource, reducing the overhead of transferring data between computing resources.

Applying the above algorithm to the weighted task directed acyclic graph TG1 in Fig. 1, the calculation results of relevant parameters are shown in the following table:

Table 2 Calculation results of relevant parameters of TG1

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9
$B(v)$	2	2	2	2	2	1	1	1	0
$S(v)$	2700	2700	2700	1400	3100	1900	1900	1900	0
$w(v)$	500	1300	700	2300	1500	2700	1400	3100	1900

According to Algorithm 2, the task priority list constructed by the task nodes of the directed acyclic graph TG1 is: *Task List* 1 = $\{v_4, v_1, v_3, v_2, v_5, v_7, v_6, v_8, v_9\}$.

3.2 Allocate Multidimensional Attribute Computing Resources of Internet of Vehicles to Task List Based on Dynamic Greedy Algorithm

When assigning computing resources to the task list, it is necessary to compare each sub-computing resource in the system and assign tasks to each sub-computing resource. In the previous analysis, the attributes of the computing resources of the Internet of Vehicles include: reliability between sub-computing resources, communication costs, computing speed, computing costs, computing energy consumption, computing stability, and computing success rate of the sub-computing resources themselves. We want to obtain a scheduling algorithm with the smallest calculation time, the most reliable calculation results, the lowest calculation energy consumption, the lowest calculation cost, and the highest calculation success rate through the scheduling system.

The greedy algorithm is to always make the best choice in the current view when solving the problem, to obtain a local optimal solution. Although the greedy algorithm may not necessarily obtain the optimal solution, it can generally obtain the suboptimal solution. We propose a computing resource allocation algorithm for the multi-dimensional QoS attributes of the Internet of Vehicles. The algorithm can be divided into two steps:

In the first step, in order to compare the multidimensional QoS properties of each computing resource in the Internet of Vehicles, we group, reduce, and normalize these properties. The sub-computing resources in the Internet of Vehicles have seven attributes, among which the five attributes of computing speed, computing cost, computing energy consumption, computing stability and computing success rate describe the attributes of computing resources themselves, and the reliability and Communication costs describe the attributes of the communication environment between computing resources. Therefore, these seven attributes are divided into two groups: the first group of attributes are the reliability and communication cost of the communication environment between computing resources; the second group of attributes are the computing speed, computing cost, computing energy consumption, and computing stability of computing resources and calculation success rate.

In the process of solving multi-dimensional QoS attribute optimization problems, the weight of each attribute is generally set according to the preference relationship and reduced to a single-objective optimization problem. We group the above two groups of attributes to reduce dimensionality, and adopt a balanced preference strategy in the dimensionality reduction process: in the first group, the weight of each attribute is 0.5; in the second group, the weight of each attribute is 0.2. We use the base number method to calculate the priority value, and the process is: (1) The priority value is non-dimensionalized. (2) The direction of the priority value is consistent. (3) Weighted summation of priority values.

In the second step, the tasks are sequentially taken out from the task priority list, and the task with the highest priority in the task list is handed over to the computing resource. According to the principle of hierarchical planning, each task is assigned to the optimal computing resource relative to the computing resource where the predecessor task is located: hierarchical planning means that when selecting computing resources, select according to a set of priority weights firstly. If the group priority weights are the same, then select according to another set of priority weights.

According to the preference relationship of the user, it is possible to choose which set of priority weights of computing resources to compare at first. We will first compare the priority weights (Pri_c) of the calculation-related attributes, and then compare the priority weights (Pri_e) of the communication-related attributes. The calculation resource selection method is called the first dynamic greedy scheduling algorithm. First compare the priority weight (Pri_e), and then compare and calculate the priority weight (Pri_c) of the relevant attributes to select computing resources is called the second dynamic greedy scheduling algorithm. According to the above ideas, line 6 in Algorithm 1 can be refined into the following two algorithms:

Algorithm 3 The first dynamic greedy scheduling resource optimization algorithm

input: task, Weighted Undirected Topological Structure Graph RG
output: The optimal matching resource for the task
Calculate the $Pri_c(v)$ of each node of RG and find $\max(Pri_c(v))$;
if there is only one node $Pri_c(v) = \max(Pri_c(v))$ then
  assign the computing resource represented by the node to the task;
else
  calculate the Pri_e value of each node of RG relative to the previous task node and find $\max(Pri_e(v))$;
  if there is only one node $Pri_e(v) = \max(Pri_e(v))$ then
    assign the computing resource represented by the node to t;
  else
    randomly select the computing resource represented by one of the nodes and assign it to t;
  end
end

Algorithm 4 The second dynamic greedy scheduling resource optimization algorithm

input: task, Weighted Undirected Topological Structure Graph RG
output: The optimal matching resource for the task
Calculate the $Pri_e(v)$ of each node of RG and find $\max(Pri_e(v))$;
if there is only one node $Pri_e(v) = \max(Pri_e(v))$ then
  assign the computing resource represented by the node to the task;
else
  calculate the Pri_c value of each node of RG relative to the previous task node and find $\max(Pri_c(v))$;
  if there is only one node $Pri_c(v)=\max(Pri_c(v))$ then
    assign the computing resource represented by the node to t;
  else
    randomly select the computing resource represented by one of the nodes and assign it to t;
  end
end

Algorithm complexity analysis: Let the number of nodes in the weighted task directed acyclic graph TG be n_t, and the number of nodes in the weighted resource undirected topology graph RG be n_r. In Algorithm 2, the time complexity of calculating the priority value of each task node is $O(n_t)$, and the time complexity of sorting all tasks by priority is $O(n_t lgn_t)$. In Algorithms 3 and 4, the time complexity of calculating the Pri_c value of each resource node is $O(n_r)$, and the time complexity of calculating the Pri_e value of each edge is $O(n_r)$, then a single task allocates optimal computing resources time complexity is $O(n_r)$, so the time complexity of allocating computing resources for each task is $O(n_t n_r)$. Therefore, the time complexity of the dynamic greedy scheduling algorithm is $O(n_t lgn_t + n_t n_r)$.

4 CASE STUDY

In order to verify the effectiveness of the above scheduling algorithm, we apply the above two dynamic greedy computing resource scheduling algorithms to Example 3. And choose the classic HEFT scheduling algorithm and the round-robin scheduling algorithm (Round-Robin) used in the Storm stream processing system as the comparison object.

Firstly, according to Algorithm 2, the task priority list is: $Task\ List_1 = \{t_4, t_1, t_3, t_2, t_5, t_7, t_6, t_8, t_9\}$.

Secondly, calculate its Pri_e and Pri_c, where the Pri_c value is shown in Tab. 3.

Table 3 Multidimensional QoS attribute values and *Pric* values of various computing resources in Example 3

type	location	speed	cost	energy	stability	success	Pri_c
On-board computing resource	Car611	200	32	68	0.53	0.75	0.496
	Car1537	260	29	73	0.48	0.8	0.504
	Car956	180	36	65	0.57	0.79	0.506
roadside computing resource	RoadSide68	300	67	35	0.74	0.9	0.584
	RoadSide47	320	56	43	0.82	0.92	0.614
background computing resource	Computing Center	340	73	46	0.92	0.95	0.612

As shown in Fig. 5, mark the Pri_e value on the connection line between computing resources. Also set the dependability = 1 and $pric_e = 0$ of the communication environment between the sub-computing resources in the same node. For example: the computing resource 29 and the computing resource 58 are both in the node Car611, then the dependability=1, $pric_e = 0$ of the communication environment between them. And the priority of its communication environment is:

$$\begin{aligned} Pr\, i_e &= trust'' \times 0.5 + pric_e'' \times 0.5 = \\ &= 1 \times 0.5 + (1-0) \times 0.5 = 1 \end{aligned} \tag{4}$$

Computing resources are represented by the letter c, and computing tasks are represented by the letter t. The round-robin scheduling algorithm is denoted as RR, the HEFT scheduling algorithm is denoted as HEFT, the first dynamic greedy algorithm is denoted as DGA*, the second dynamic greedy scheduling algorithm is denoted as DGA**, and the unit of scheduling length is seconds (Fig. 6).

From the comparison of the above scheduling results, the round-robin scheduling algorithm has the longest scheduling length, while the classic HEFT scheduling algorithm has the shortest scheduling length. The scheduling length of the first dynamic greedy scheduling algorithm is close to that of the classic HEFT scheduling algorithm, because when the scheduling algorithm selects computing resources for computing tasks, it gives priority to the combined priority of computing resources. However, when the second dynamic greedy scheduling algorithm selects computing resources for computing tasks, because the combination priority of the communication environment is considered firstly, and the combination priority of computing resources is considered secondly. The scheduling length of this algorithm is longer than the classic HEFT scheduling algorithm and the first dynamic greedy scheduling algorithm. But the scheduling length of the second dynamic greedy scheduling algorithm is shorter than that of the round-robin scheduling algorithm.

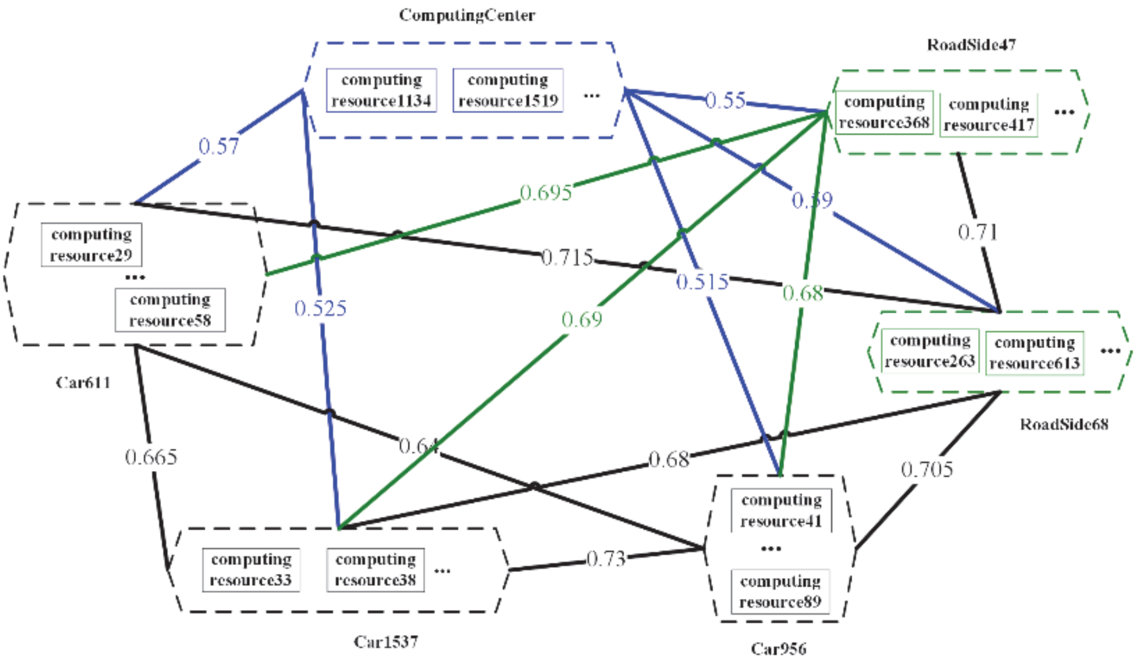

Figure 5 Schematic diagram of the priority value of distributed computing resources

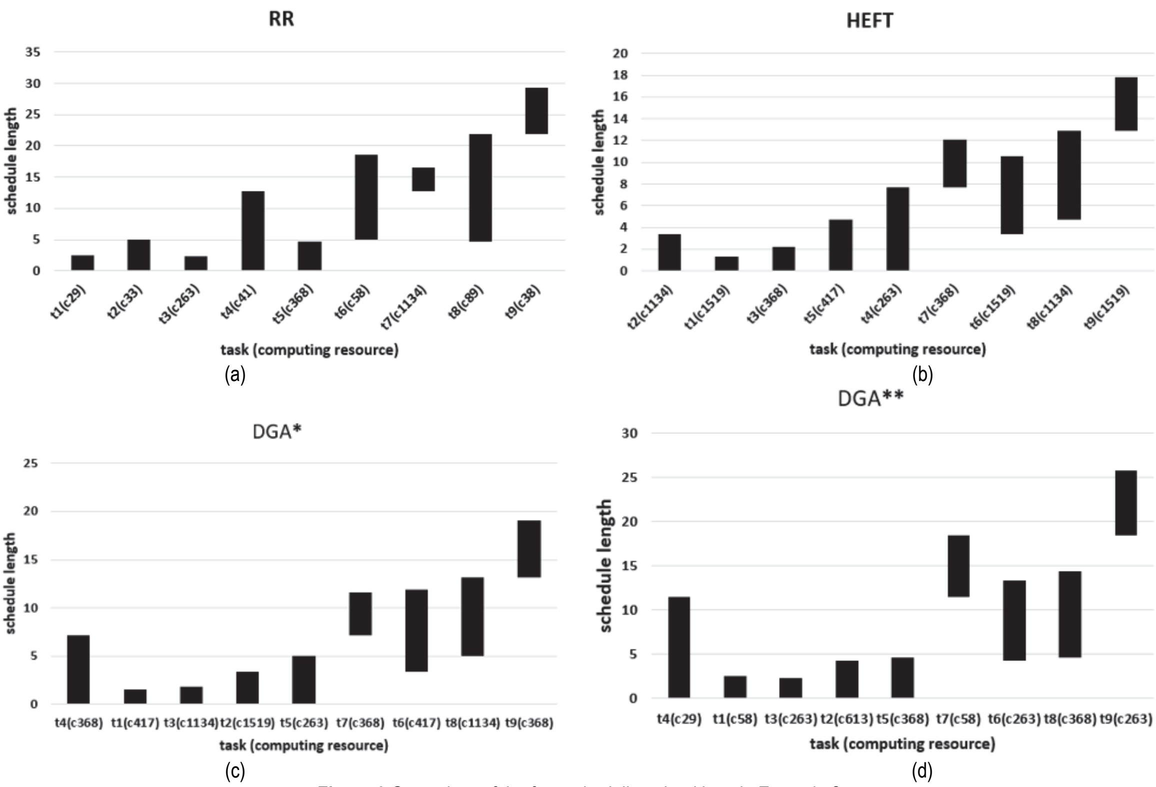

Figure 6 Gantt chart of the four scheduling algorithms in Example 2

In order to compare these four scheduling algorithms, we compare the calculation results of the QoS attribute values of each dimension during the calculation process of the calculation task TG1 in Example 2. Among them, the method of cumulative multiplication is used when calculating the reliability of resources, and the method of summation is used for communication costs. The comparison results are shown in the following figure:

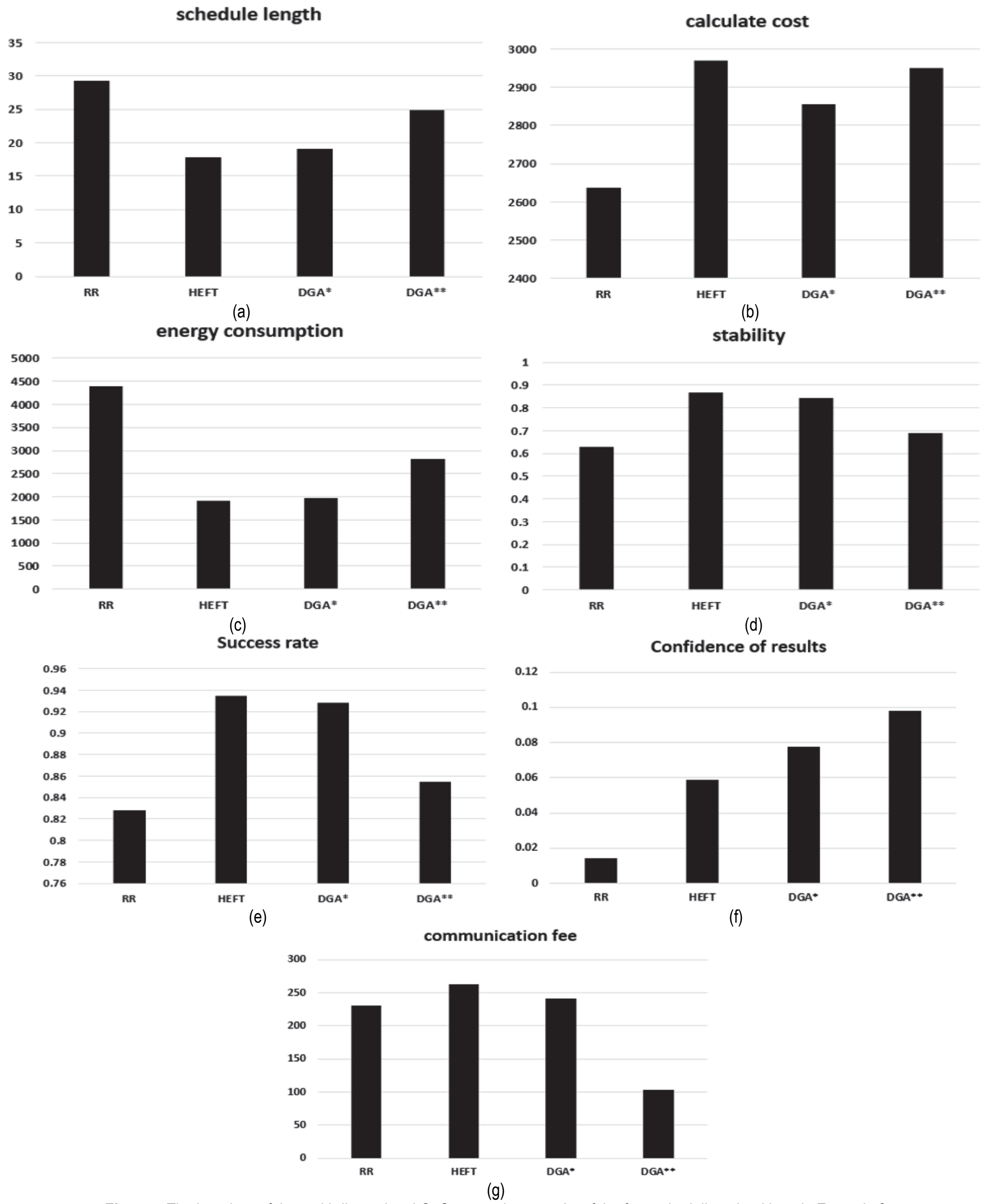

Figure 7 The bar chart of the multi-dimensional QoS comparison results of the four scheduling algorithms in Example 3

From the above comparison, it can be seen that the performance of each attribute of the first dynamic greedy scheduling algorithm is close to that of the classical HEFT scheduling algorithm, while the second dynamic greedy scheduling algorithm is significantly better than the classical HEFT scheduling algorithm and the first dynamic greedy scheduling algorithm in terms of reliability of calculation results and communication costs. Although the round-robin scheduling algorithm is optimal in terms of computing costs, this is in exchange for increasing computing time. Therefore, in the Internet of Vehicles environment with heterogeneous computing resources and complex communication environment, the comprehensive performance of the first dynamic greedy scheduling algorithm and the second dynamic greedy scheduling algorithm is effective.

5 CONCLUSIONS

With focus on the scheduling problem of heterogeneous and mobile computing resources in distributed computing tasks in the Internet of Vehicles, this paper first establishes a distributed computing resource model for the Internet of Vehicles based on the theory of computing-aware networks: Directed acyclic graph task model and seven-dimensional QoS weighted undirected topology graph computing resource model. Secondly, based on the task and resource model, a dynamic greedy algorithm-based task of loading and computing resource scheduling algorithm is proposed. Finally, the effectiveness of the proposed scheduling algorithm is demonstrated through a case study. The first dynamic greedy scheduling algorithm is suitable for mobile computing scenarios with good communication conditions, such as non-main roads with few vehicles. The second dynamic greedy scheduling algorithm is suitable for mobile computing scenarios with poor communication conditions, such as congested urban main roads. Among these two algorithms, users can choose or perform adaptive conversion according to specific physical application scenarios.

Acknowledgements

This work is supported by the 2021 Basic Science Research Program of Nantong, Jiangsu, China (No.: JC2021127), the Science Foundation of Jiangsu Shipping College, China (No.: HYKY/2021B07), the Philosophy and Social Science Research in Colleges and Universities of Jiangsu Province (No.: 2021SJA1659), Nantong City-Level Social and People's Livelihood Science and Technology Project MSZ2022166, the 2022 Basic Science Research Program of Nantong, Jiangsu, China (No.: JC12022054).

6 REFERENCES

[1] Ji, B., Zhang, X., Mumtaz, S., Han, C., Li, C., Wen, H., & Wang, D. (2020). Survey on the internet of vehicles: Network architectures and applications. *IEEE Communications Standards Magazine*, *4*(1), 34-41. https://doi.org/10.1109/MCOMSTD.001.1900053
[2] Priyan, M. K. & Devi, G. U. (2019). A survey on internet of vehicles: applications, technologies, challenges and opportunities. *International Journal of Advanced Intelligence Paradigms*, *12*(1-2), 98-119. https://doi.org/10.1504/IJAIP.2019.096957
[3] Praneeth, V., Kumar, K. R., & Karyemsetty, N. (2021). Security: Intrusion prevention system using deep learning on the internet of vehicles. *International Journal of Safety and Security Engineering*, *11*(3), 231-237. https://doi.org/10.18280/ijsse.110303
[4] Zhao, T. A. & Li, W. (2022). Design Configuration and Technical Application of Rotary-Wing Unmanned Aerial Vehicles. *Mechatronics and Intelligent Transportation Systems*, *1*(1), 69-85. https://doi.org/10.56578/mits010108
[5] Kakulla, S. & Malladi, S. (2022). Phishing and sybil enhanced behavior processing and footprint algorithms in vehicular ad hoc network. *International Journal of Safety and Security Engineering*, *12*(1), 83-96. https://doi.org/10.18280/ijsse.120111
[6] Arooj, A., Farooq, M. S., Akram, A., Iqbal, R., Sharma, A., & Dhiman, G. (2022). Big data processing and analysis in internet of vehicles: architecture, taxonomy, and open research challenges. *Archives of Computational Methods in Engineering*, *29*(2), 793-829. https://doi.org/10.1007/s11831-021-09590-x
[7] 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, 25*(2), 1175-1187. https://doi.org/10.1007/s10586-021-03532-9
[8] Yang, Y. (2019). Multi-tier computing networks for intelligent IoT. *Nature Electronics*, *2*(1), 4-5. https://doi.org/10.1038/s41928-018-0195-9
[9] Ndashimye, E., Ray, S. K., Sarkar, N. I., & Gutiérrez, J. A. (2017). Vehicle-to-infrastructure communication over multi-tier heterogeneous networks: A survey. *Computer networks*, *112*, 144-166. https://doi.org/10.1016/j.comnet.2016.11.008
[10] Yi, S., Li, C., & Li, Q. (2015). A survey of fog computing: concepts, applications and issues. *Proceedings of the 2015 workshop on mobile big data,* 37-42. https://doi.org/10.1145/2757384.2757397
[11] Abbas, N., Zhang, Y., Taherkordi, A., & Skeie, T. (2017). Mobile edge computing: A survey. *IEEE Internet of Things Journal*, *5*(1), 450-465. https://doi.org/10.1109/JIOT.2017.2750180
[12] Lü, Z., Yuan, M., Lü, Y., & Wang, Z. (2021). An overlapping architecture of large-size wide-body aircraft based on cloud sea computing in 5G OGCE. *Advances in Wireless Communications and Applications*, 7-13. https://doi.org/10.1007/978-981-15-5879-5_2
[13] Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications, 80*(5), 8091-8126. https://doi.org/10.1007/s11042-020-10139-6
[14] Zhao, Z. & Yuan, Q. (2022). Integrated Multi-objective Optimization of Predictive Maintenance and Production Scheduling: Perspective from Lead Time Constraints. *Journal of Intelligent Management Decision*, *1*(1), 67-77. https://doi.org/10.56578/jimd010108
[15] Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: an overview. *Soft computing*, *22*(2), 387-408. https://doi.org/10.1007/s00500-016-2474-6
[16] Shao, X. H., Chang, D. F., & Li, M. J. (2022). Optimization of Lateral Transfer Inventory of Auto Spare Parts Based on Neural Network Forecasting. *Journal of Intelligent Systems and Control*, *1*(1), 2-17. https://doi.org/10.56578/jisc010102
[17] Jingmei, L., Xue, W., & Yanxia, W. (2014). An Improved Priority List Task Scheduling Algorithm. *Computer Science, 41*(5), 20-23.
[18] Xiang, D., Lin, H., Ouyang, J., & Huang, D. (2022). Combined improved A* and greedy algorithm for path planning of multi-objective mobile robot. *Scientific Reports, 12*(1), 1-12. https://doi.org/10.1038/s41598-022-17684-0
[19] Kaul, S., Fayaz, S. A., Zaman, M., & Butt, M. A. (2022). Is decision tree obsolete in its original form? A Burning debate. *Revue d'Intelligence Artificielle*, *36*(1), 105-113. https://doi.org/10.18280/ria.360112

Contact information:

Huiyong LI
(Corresponding author)
School of Intelligent Manufacturing and Information,
Jiangsu Shipping College,
Nantong 226010, China
E-mail: lihuiyong@jssc.edu.cn

Shuhe HAN
School of Intelligent Manufacturing and Information,
Jiangsu Shipping College,
Nantong 226010, China
E-mail: hsh@jssc.edu.cn

Xiaofeng WU
School of Intelligent Manufacturing and Information,
Jiangsu Shipping College,
Nantong 226010, China
E-mail: wuxiaofeng@jssc.edu.cn

Furong WANG
School of Intelligent Manufacturing and Information,
Jiangsu Shipping College,
Nantong 226010, China
E-mail: wfr@jssc.edu.cn