electronics

MDPI

Article

Learning-Based Task Offloading for Marine Fog-Cloud Computing Networks of USV Cluster

Kuntao Cui [1], Bin Lin [2,*], Wenli Sun [1] and Wenqiang Sun [1]

[1] Navigation College, Dalian Maritime University, Dalian 116026, China; ckt001@163.com (K.C.); itslab@dlmu.edu.cn (W.S.); sunwenqiang@dlmu.edu.cn (W.S.)
[2] Information Science and Technology College, Dalian Maritime University, Dalian 116026, China
* Correspondence: binlin@dlmu.edu.cn

Received: 30 September 2019; Accepted: 2 November 2019; Published: 5 November 2019

check for updates

Abstract: In recent years, unmanned surface vehicles (USVs) have made important advances in civil, maritime, and military applications. With the continuous improvement of autonomy, the increasing complexity of tasks, and the emergence of various types of advanced sensors, higher requirements are imposed on the computing performance of USV clusters, especially for latency sensitive tasks. However, during the execution of marine operations, due to the relative movement of the USV cluster nodes and the network topology of the cluster, the wireless channel states are changing rapidly, and the computing resources of cluster nodes may be available or unavailable at any time. It is difficult to accurately predict in advance. Therefore, we propose an optimized offloading mechanism based on the classic multi-armed bandit (MAB) theory. This mechanism enables USV cluster nodes to dynamically make offloading decisions by learning the potential computing performance of their neighboring team nodes to minimize average computation task offloading delay. It is an optimized algorithm named Adaptive Upper Confidence Boundary (AUCB) algorithm, and corresponding simulations are designed to evaluate the performance. The algorithm enables the USV cluster to effectively adapt to the marine vehicular fog computing networks, balancing the trade-off between exploration and exploitation (EE). The simulation results show that the proposed algorithm can quickly converge to the optimal computation task offloading combination strategy under heavy and light input data loads.

Keywords: task offloading; marine fog-cloud computing networks; unmanned surface vehicles

1. Introduction

In recent years, USVs have made important advances in civil, maritime, and military applications. In the foreseeable future, unmanned surface vehicles will perform more missions with important application prospects.

They can be used for maritime search and rescue, maritime inspections, environmental monitoring, etc. In military applications, they can be used for different tasks, including maritime security, special operations support, and maritime interception operations support, etc.

During tasks execution, USVs need to complete necessary marine situational awareness, autonomic strategy formulation, dynamic team formation, and joint mission evaluation by itself or other cooperative nodes within the same mission framework, and use encrypted radio frequency communication link for information distribution and collaboration with other platforms (unmanned or not).

With the continuous improvement of autonomy, the increasing complexity of tasks, and the emergence of various types of advanced sensors, higher requirements are imposed on the computing performance of USV cluster, especially for latency sensitive tasks [1]. The main reasons include the following aspects:

- In complex mission operations, USV team nodes must be assigned different roles, depending on limited platform size, energy, and payloads. For example, some nodes need to be installed with more payloads to complete the detection and sensing functions for the cluster, and no more computing resources can be configured. At the same time, some nodes can be configured with stronger computing resources, but fewer other payloads. By assigning different roles to cluster nodes, it is possible to make unmanned node functions more specific and fully utilized.
- At the level 1 of unmanned system's autonomy, cluster nodes do not need to have strong computing resources. All sensor data will be transmitted to the remote control center, and the operator judges the situation and sets the corresponding commands. As the task complexity increases, the cluster node autonomy must be improved, and remote control mode is far from meeting the requirements of complex tasks, and cannot take the advantages of unmanned systems.
- At the same time, cluster nodes need to be equipped with more advanced sensors to achieve higher requirements of detection and sensing through more sensor data to meet the needs of complex tasks. This large amount of data cannot be handled by the remote control center because it will generate a large delay and cannot meet the requirements of cluster nodes to respond to situational changes in a short time.

In the past few years, we focused on research of collaborative autonomy and control of USVs, and we had achieved level 2 of unmanned system's autonomy. In recent years, we have begun to research new communication technologies because we find that the existing maritime communication technologies cannot be able to support autonomy of USVs to cope with more complex tasks, especially latency-sensitive tasks.

The picture below shows a certain type of USV that we have developed which has significant applications in some areas. USVs are undergoing marine testing, as shown in Figure 1.

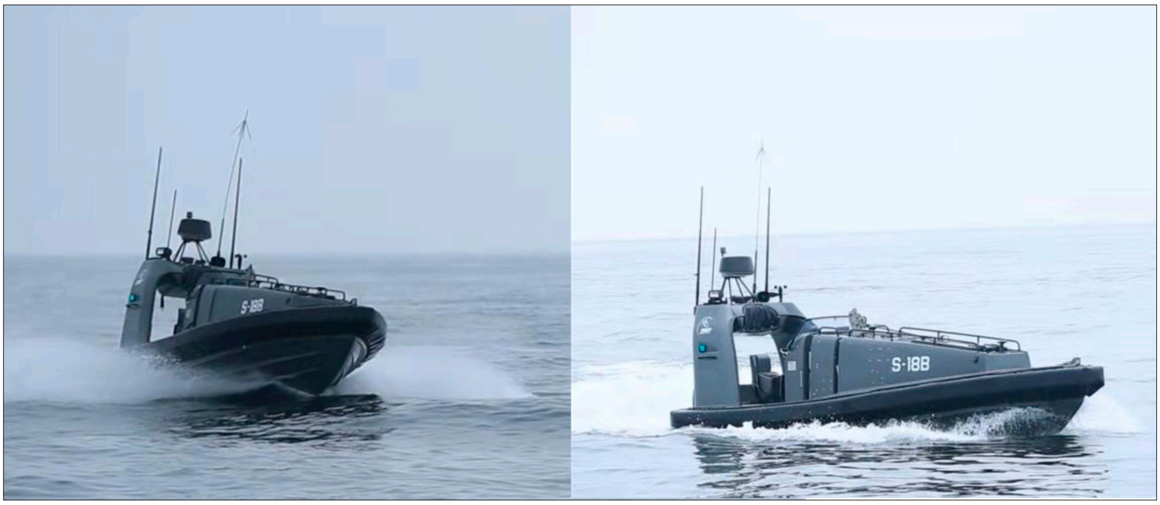


Figure 1. Marine test of unmanned surface vehicles (provided by our research team).

For the above reasons, it is important to make full use of the computing resources of USV cluster. At the same time, combined with the computing resource of the remote cloud, the overall computing performance can be significantly improved.

However, due to the limited marine communication conditions (such as bandwidth limitation, channel quality, etc.) between USV cluster and remote cloud, if a large number of complex computing task are offloaded to a centralized cloud with rich computing resources, the transmission costs are also relatively high.

Although the application of marine unmanned systems is just at the beginning, the development capability is very strong. The capacity of marine unmanned platforms is increasing. Advanced sensor technologies continue to emerge and are used in a wider range of applications and deployments. In order to cope with these factors, there is an urgent need for a reliable, flexible, and marine communication mechanism with effective carrying capacity to achieve multi-level information process, exchange, and transmission.

Therefore, this paper proposes a marine fog-cloud computing architecture for USV clusters, which can dynamically allocate USV cluster and cloud resources to achieve overall optimized computing performance.

The contribution of this work can be summarized as follows:

- This paper proposes a marine fog-cloud computing architecture for USV clusters, and analyzes the network dataflow.
- For typical application scenarios, we consider a marine vehicular fog-cloud computing model of moving USV cluster nodes.
- An optimized learning-based computation offloading mechanism based on classic MAB theory is proposed. This mechanism enables USV cluster nodes to dynamically make decisions by learning the potential offloading performance of their neighboring team nodes to minimize average computation task offloading delay.
- Furthermore, we propose an optimized algorithm named AUCB algorithm and design corresponding simulations to evaluate the performance under typical conditions.

2. Related Work

The research of unmanned systems involves a deep integration of many disciplines such as systems engineering, control engineering, information, and communication engineering. It is also the frontier research field and future development direction of marine science and technology.

With the rapid development of information technology, technological advances have paved the way for the emergence of complex services. Computation task offloading is attractive for Internet of Things (IoT) and edge computing. Typically, task offloading can occur between sensors, edge devices, fog nodes, or IoT nodes [2].

In order to improve the task scheduling efficiency of the marine collaborative edge systems and reduce communication cost, an optimized algorithm is proposed in [3].

SDN and fog computing have been integrated into maritime broadband communication systems to minimize the total weight delay of the stand-alone scheduling scenario and to achieve the minimum delay for weighted upload packets in [4].

In the marine mobile computing environment, the network status changes at any time during task execution, and the migration decision problem in the mobile fog environment needs to be transformed into a runtime offloading decision-making problem.

If the migration decision is made without considering the dynamics of the marine mobile fog environment, the strategy will result in incorrect migration, and ultimately affect the comprehensive dynamic computing performance of the USV fog cluster.

As far as the scope of this paper is concerned, at present, there are many researches on static computation offloading decision-making. In the system development stage, the offloading strategy is formulated through program analysis. After development completion, the algorithm strategy will no longer change. For example, the problems studied in [5–8] are static offloading decisions.

Some existing related research based on mobile device edge computing mostly utilizes the idle mobile node device around user terminals to complete the offloading of computing task through D2D communications. The corresponding research is carried out in [9,10].

Reference [11] designs the computation offloading strategy under renewable energy supply and the opportunistic-based mobile self-organizing cloudlet offloading strategy.

USV clusters work in a challenging communications environment, and the reliable communication system is a vital factor for them to operate safely and accomplish mission tasks. It needs to be able to ensure the reliability and effectiveness of the communication mechanism under the premise of coping with the ever-changing adverse factors, even in the harsh electromagnetic and network environment.

Therefore, in a dynamic environment, the migration strategy design should be formulated according to the current situation, and the migration strategy should change constantly.

Driven by the latest developments in artificial intelligence, the fog radio access networks are seen as a potential architecture to support IoT services. A joint mode selection and resource management algorithm based on deep reinforcement learning is proposed in [12].

F-RAN is an emerging architecture that takes advantage of edge computing and distributed storage in edge devices. Reference [13] proposes a NOMA-based F-RAN architecture with powerful edge computing capabilities to meet the heterogeneous requirements of mobile vehicular systems.

In a fog computing network, a mobile device can offload its data or computation intensive tasks to a fog node in its vicinity. Based on theoretical analysis, a multi-objective optimization problem is proposed in [14] with the objective of minimizing energy consumption, processing delays and communication costs for each mobile terminal.

The vehicular edge computing network integrates the computing resources of nearby vehicles and provides computing services. A learning-based offloading mechanism is proposed in [15].

However, obtaining an optimal strategy in such a dynamic system is challenging. In addition to immediate rewards, reinforced learning (RL) also considers long-term goals, which are important for time-varying dynamic systems. Reference [16] proposes a RL-based optimized algorithm to solve the task allocation problem in wireless mobile edge computing network.

Reference [17] proposes a task offloading algorithm based on deep Q-network. The algorithm can learn to develop an optimal offloading strategy without relying on the prior knowledge of dynamic statistics.

Related research on computation migration systems, cloud service operators and cloud resource operators in [18] from three aspects: migration decision-making, task access control and energy-efficient resource management. The research fully considers the problems and challenges brought by the mobile cloud environment to the computation offloading system.

Fog computing is expected to provide low latency computing services at the edge of the network for the IoT systems. Reference [19] proposes a computation task offloading algorithm to simulate the competition between IoT terminals and to distribute the limited computing resources of the fog nodes.

Fog computing can provide latency-sensitive service for terminals and reduce power consumption and traffic congestion. It achieves efficient resources utilization and better performance [20].

Reference [21] performs a strict comparative analysis of the fog computing and the conventional cloud computing in the IoT environment. The results show that the performance of fog computing is superior to traditional cloud computing with the increase in the number of applications requiring real-time services.

Reference [22] proposes a new fog computing model that can alleviate the potential problems of dedicated computing infrastructure and the slow response in cloud computing. The results show that fog computing can greatly improve the performance of the analysis service compared to using only the cloud model.

Reference [23] investigates the joint allocation of radio and computing resources to optimize system performance and improve user satisfaction, and proposes a matching game mechanism to provide a distributed solution for the joint resource management.

We adopt a lot of ideas from related research on resource management, communication, and other aspects in [1,24].

Reference [25] proposes a fog-cloud computing architecture for unmanned aerial vehicles and carried out the corresponding computing performance analysis. A utility-aware data transmission mechanism for delay tolerant networks is proposed in [26], which considers the internal properties of nodes and external contacts.

3. System Model

3.1. Computing Architecture

In this section, we will explain several existing problems of marine communication, and discuss the proposed model and expected goals.

During the execution of a marine mission, the USV cluster must be able to perform several important functions such as information distribution and task collaboration, which include information distribution, task coordination, team roles allocation, dynamic team formation, interaction with team members or external devices, etc.

The cluster nodes can use wireless communication link for information distribution. They can distribute dynamic mission information to other cooperative nodes (unmanned or not).

However, the marine communication environment has corresponding special characteristics—such as large scale, limited bandwidth, communication delay, etc.—and requires the following considerations:

- For the most computing tasks, a large amount of data generated by the USV cluster nodes which are responsible for situational awareness, needs to be processed, and the processed data is much smaller than the raw data, which is below several orders of magnitude.
- The computing performance of remote cloud is much higher than that of each USV cluster node, but limited by the marine communication conditions between marine cluster and remote cloud, the transmission delay is also very large and unrealistic, if all raw data is offloaded to cloud for calculation.
- Although the computing performance of the cluster node is lower than remote cloud server, considerable computing performance can also be obtained by making full use of the parallel computing performance of each cluster node and the communication bandwidth advantage within cluster coverage.
- In general, for most computing tasks, USVs are usually energy insensitive.

Therefore, we propose a marine fog-cloud computing architecture, and develop an optimized learning-based offloading mechanism to solve several mentioned problems.

The marine fog-cloud computing architecture can provide a promising solution to these problems. In the proposed computing architecture, task offloading to the distributed USV cluster nodes enables the full use of underutilized computing resources to mitigate the communication load, and reduce processing delays.

Generally, the moving USV cluster nodes can be classified into task nodes (TaNs) and computation nodes (CoNs). TaNs generate computation tasks that need to be offloaded to available CoNs of USV fog cluster, or remote cloud, as shown in Figure 2.

Each node of USV fog cluster has networked structure features such as decentralization and roles equality. By optimizing the utilization of fog node resources, the overall performance of the network is optimized. The fog coordinator mainly performs roles allocation, task coordination, dynamic team forming, or radio frequency communication between fog cluster and remote cloud.

Although the conditions of marine communication are relatively limited currently, but when the fog cluster nodes are few, or the computing resources are not particularly rich, the fog-cloud computing architecture can also obtain considerable overall computing performance by fully utilizing the computing resources of remote cloud.

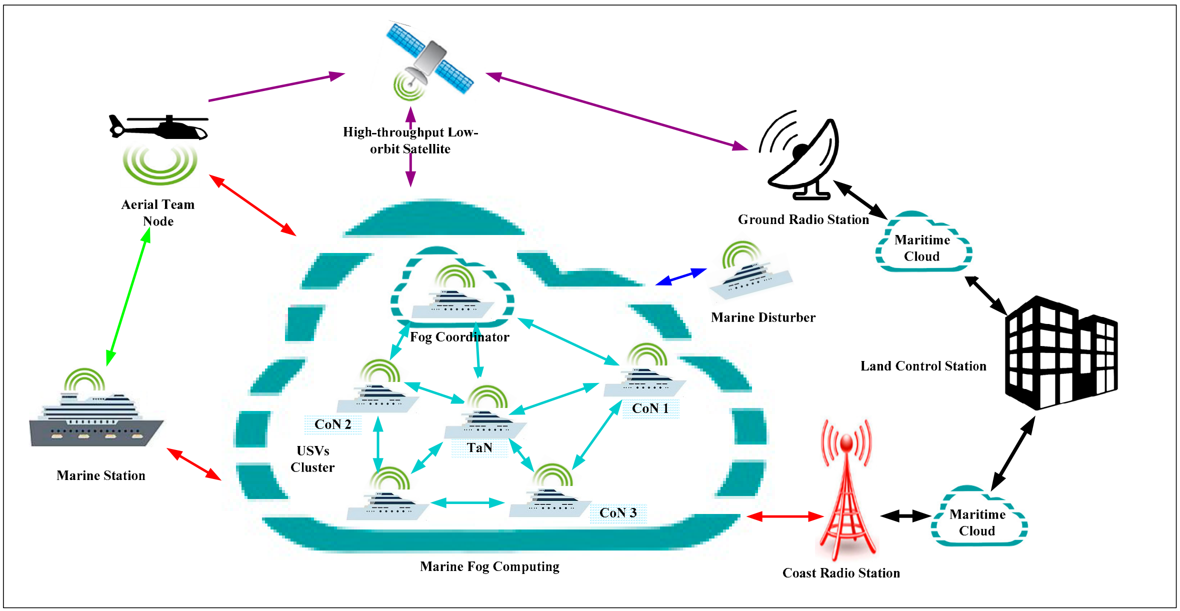


Figure 2. Marine fog-cloud computing architecture for USV cluster.

When the computing resources available to the fog cluster nodes are abundant, the remote cloud will be less important to the overall computing performance. Therefore, the fog-cloud computing architecture fully considers the comprehensive consideration of the above situations to achieve acceptable computing benefits.

Vehicular fog computing integrates fog computing and vehicular networks and is expected to provide real-time service for latency-sensitive task [27].

In this paper, the proposed task allocation and resource management algorithms of fog-cloud computing architecture are optimized to reduce processing delay and communication load, and take full advantage of the computing performance of the fog-cloud computing architecture.

3.2. *Marine Fog-Cloud Computing Network Dataflow*

With the continuous improvement of unmanned system's autonomy and the increasing complexity of tasks, higher requirements are imposed on the computing performance of USV cluster, especially for latency sensitive tasks.

Many tasks must be completed at the marine cluster level to meet latency requirements—such as teamwork keeping, resource allocation, cluster coordination, etc.—as shown in Figure 3. It also shows the computation offloading delays considered in the proposed computing model, including transmission delay and computation delay.

In addition, because industrial big data is often unstructured, it is pruned and refined by local fog before being sent to the remote cloud [28].

In general, the transmission delay includes the delay in which the raw data is migrated from TaNs to the CoNs, and the delay in returning the processed data. The transmission delay is affected by the amount of data, transmission power, channel state, etc.

The computing delay refers to the delay when the CoNs complete the data computation, which is usually determined by the raw data size and the computing performance or available resources of the CoNs.

However, during task execution, the channel state and the computing performance or available resources of the computation node cannot be known to the task node in advance, when making a computation migration strategy.

On the other hand, the signaling overhead will be much high, if each task node requests the parameters of all candidate computation nodes in each time period.

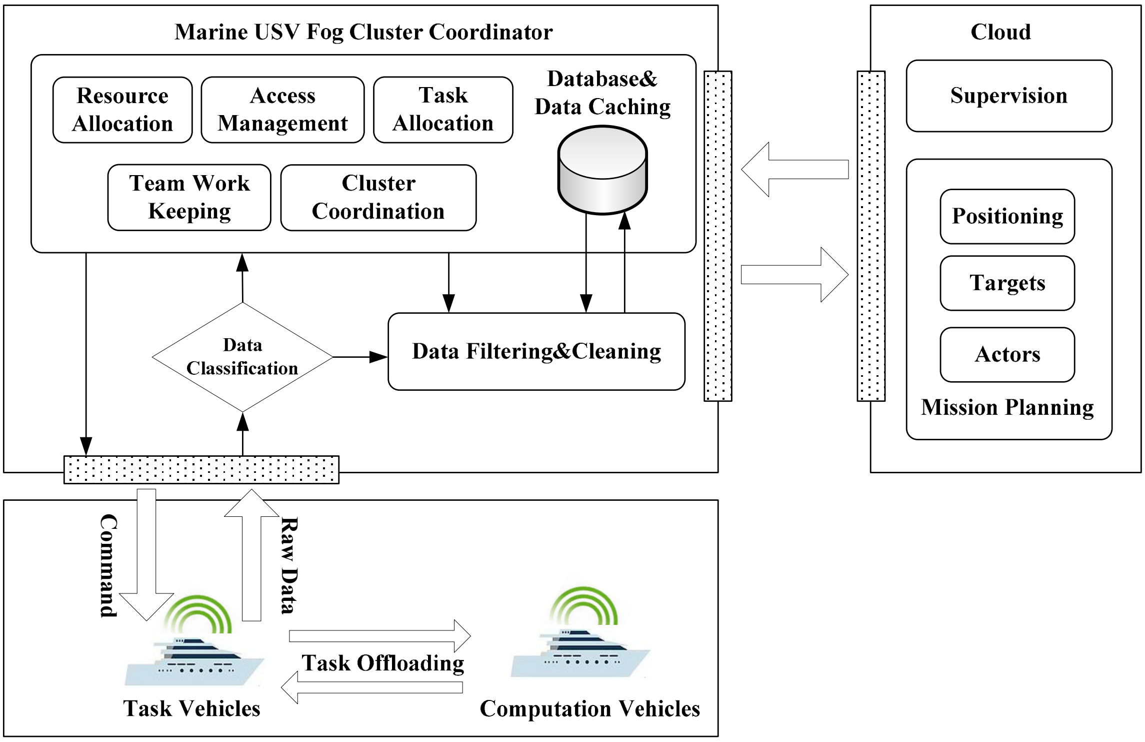


Figure 3. Dataflow of marine fog-cloud computing network for USV cluster.

Therefore, we need to develop an effective scheduling mechanism or algorithm that can be used to take full advantage of the computing performance of the fog-cloud computing architecture, and reduce processing delay and communication load, when making the offloading decision without additional signaling overhead.

3.3. Typical Application Scenarios of Marine Fog-Cloud Computing

This section describes typical application scenarios of the marine fog-cloud computing architecture for USV clusters. These application scenarios require low latency of computing services.

The fog-cloud computing architecture and efficient algorithm design can be used for providing optimization for computing services.

3.3.1. Marine Situational Awareness (MSA)

The USV's MSA systems can obtain sensing information from onboard sensors, or receive cluster information through the communication module, including task collaboration data from teammates, and convert the information into a generic situation picture. This is important for the unmanned system's collaborative autonomy.

Four different levels of requirements should be achieved, including:

Level 1, object evaluation, it completes the fusion and elimination of data onboard sensors, or external collaborative data sources of heterogeneous nodes, and generates an overall situational map within the task framework.

Level 2, situational assessment, it is used to evaluate the fused general contextual picture to identify object attributes within the coverage of the teammates' sensors (eg, hostile, friendly, neutral, etc.) and to prioritize emerging or potential threat-related targets.

Level 3, predictive consciousness, it is used to identify the motion parameters and future trajectories of possible threats and assesses the intentions of possible threats.

Level 4, process optimization, it determines that when information from an MSA system does not meet the operation requirements, other means can be taken to obtain important information, such as requesting a mission plan from another platform or from a communication network [29].

The MSA system is also capable of collecting and maintaining other information such as sea states, weather conditions, etc. All the information will also be provided to the mission plan to make autonomic strategy for guiding USV cluster nodes' behaviors.

With the emergence of advanced sensors, a large amount of data is generated for situational analysis. USV cluster nodes need to intelligently analyze, process, and convert related data into a general situation picture in the mission framework for collaborative nodes. The processed information can also be distributed to remote monitoring and control centers.

3.3.2. Autonomic Strategy Formulation

Autonomic strategy formulation determines and establishes the goals, mission and objectives of USV cluster. It enables the USV cluster to identify and select the appropriate plan among all available strategies that can achieve its goals.

With the continuous adoption of intelligent algorithms including reinforcement learning, the autonomy level of USV clusters is also constantly improving. At the same time, more and more complex and real-time tasks require USV cluster to make autonomous decisions immediately to respond to the changing environmental requirements internal or external.

This is extremely important for unmanned systems. In addition to being able to accomplish tasks better, it also enables USV clusters to respond effectively to complex and changing environments, embodying the intelligence of unmanned systems.

Although the computing performance of USV cluster nodes is still relatively low compared with cloud computing, but with the continuous development of information technology, the computing performance of USV cluster nodes is also continuously improved.

Especially for latency-sensitive tasks, it is also an important method to obtain considerable computing performance, by making full use of the USV cluster computing resources, especially when marine communication conditions (bandwidth limitation, channel quality, etc.) are limited between the cluster and remote cloud.

3.3.3. Dynamic Team Formation

It accommodates the composition and rebuild of the USV cluster as required, including teammates identification, team structures, and roles allocation within mission framework.

It is very important for USV clusters to cope with changes to refactor the team organization, including team member loss or failure, new team members joining, current task changes, and other conditions.

As conditions change, the structure of the team will be dynamically adjusted and the roles will be re-allocated to form new interactions.

In the process of dynamic team formation of multiple heterogeneous platforms, team members are regrouped according to the requirements of tasks, forming a new organization to complete a separate task, or merging the organizations to form a new joint task team.

The heterogeneous fog cluster can dynamically perform computation migration/offloading, and utilize the dynamic resources of mobile platforms in the network to provide computing services. The marine fog cluster should also allow mobile nodes to join and leave at any time.

The computing task is completed through the cooperation of the mobile platforms, and the dynamic performance of the edge computing service is maintained.

Through the dynamic mobile edge computation migration and offloading strategy, the computing performance of USV fog cluster nodes are optimized under the conditions of limited maritime communication resources. The marine fog-cloud computing architecture reduces unnecessary offloading of computing tasks and communication tasks and maintains good dynamic computing performance.

In summary, a satisfactory computation offloading strategy can fully utilize the heterogeneous, diversified storage and computing resources of USV cluster nodes, and should also consider the mobility of the cluster nodes.

3.3.4. Joint Mission Evaluation

Joint mission evaluation can be used to provide a higher-level operational assessment of large-scale joint operations and diversification of mission implementation.

It should have the ability to integrate multiple mission nodes, multi-dimensional situational information space, operation patterns, etc., to build modular joint mission evaluation space.

After operations completion, it is not difficult to evaluate the results of simple task for USV cluster. However, for large-scale, multi-factor, multi-node participated operations, USV cluster needs to conduct comprehensive assessment of task completion degree based on a large number of multi-dimensional observation data, and generate task evaluation results.

Usually, mission evaluation should be completed in a short period of time and clear evaluation results should be drawn, so it is also time sensitive.

The task evaluation results will be sent to the remote monitoring center to reduce communication load, data processing, and communication delay.

3.4. Massive Data of Heterogeneous Sensors and Computing-Intensive Tasks

With the continuous improvement of unmanned system's autonomy, and the increasing complexity of tasks, various types of advanced sensors have emerged. At the same time, higher requirements are imposed on the computing performance of USV cluster, especially for latency sensitive tasks.

The following functions should be achieved: multi-sensor data fusion, re-configurability of sensor weighting, adaptability of fault sensors and erroneous data, intelligent heterogeneous data association, etc.

3.4.1. Multi-Sensor Information Fusion

In complex and uncertain marine surface environments, USVs must have abilities of integrate multi-sensor data to maintain accurate and continuous sensing of the surrounding conditions and to transform sensory data into meaningful information within the task framework.

At the same time, this feature provides the USVs with the adaptability to perform tasks in dynamic, complex situations.

3.4.2. Reconfigurability of Sensor Weighting

This feature refers to the re-configurability of the sensor weight of USVs during task execution. When USV's heterogeneous sensor networks are used for multi-sensor data fusion processing, each sensor onboard may have different weights for different applications and different tasks.

The sensor management system must have the ability to dynamically reconfigure sensor weights to achieve good performance of the USV platforms for different mission operations.

3.4.3. Adaptability of Faulty Sensors and Erroneous Information

In many cases, USVs need to work in a heterogeneous sensor network due to the requirements of different tasks. This is also one of the important parameters to be considered when designing a multi-sensor data fusion system.

Based on this situation, the sensor management system must be dynamically self-adaptive of sensor failure and erroneous data, thereby enhancing the adaptability of USV sensor network to cope with changes.

3.4.4. Intelligent Heterogeneous Data Association

In general, multi-sensor data fusion systems based on heterogeneous sensor networks must have abilities to process different sensor data simultaneously.

Since the combination of heterogeneous sensors may change during task execution, the data combination also changes. Intelligent heterogeneous data associations must be performed before multi-sensor data fusion and input to the decision-making module of the USVs.

3.5. Computation Tasks Offloading

For the above typical application scenarios, we consider a marine vehicular fog-cloud computing system in which USV cluster nodes can be classified into two categories: TaNs and CoNs as shown in Figure 2. TaNs generate computation tasks that need to be offloaded to CoNs of USV fog cluster, or remote cloud.

At the same time, they can also complete partial computing tasks, if conditions permit. Roles of TaN or CoN are not fixed during task execution, which depends on whether the computation resources on board are shareable and sufficient or not. CoNs are employed as fog computing nodes to provide computation services, while TaNs generate computation tasks that need to be offloaded.

For each TaN of fog cluster, the surrounding CoNs in same marine operations within its communication range *Cr* can be considered as candidate available computation task allocation nodes.

TaN can obtain the dynamic information of each available CoN, including USVs ID, position, and speed, provided by automatic identification system. The computation task will be migrated to several of available CoNs of fog cluster or remote cloud according to task offloading algorithms.

In the proposed framework, task migration decisions will be completed in a distributed manner. Each TaN can make its computation task migration decisions independently, in order to avoid additional, large signaling overhead.

In the above typical application scenarios, we focus on a representative task offloading in the marine operations for total T time periods. The TaN generates tasks, makes computation offloading decisions in discrete-time t, selects several available CoNs or remote cloud, and performs computation task offloading and receives the processed results.

Denote the available CoNs set by $\mathrm{N}(t)$, and we should note that $\mathrm{N}(t)$ may change with time since USV cluster nodes are moving during execution of operations.

3.5.1. Computation Delay

In the proposed mechanism, the input raw data size in time period t can be denoted by x_t (in bits) which should be offloaded from TaN to CoN n. Denote the output processed data size by y_t (in bits) which will be fed back to TaN. The computing performance parameter is denoted by ω_t, which indicates the number of CPU cycles required to process each bit of data

The total workload can be given by $x_t\omega_t$ in time period t [30].

For each CoN n, the maximum computing capability can be denoted by $F(n)$ (in CPU cycles per second). In general, multiple computing tasks may be processed simultaneously, and the available computing capability to TaN can be denoted by $F(t,n)$ in time period t. Therefore, the computation delay $D_c(t,n)$ of CoN n is

$$D_c(t,n) = \frac{x_t\omega_t}{F(t,n)} \tag{1}$$

3.5.2. Transmission Delay

However, in the real systems of USV fog cluster, $F(t,n)$ cannot be known to the TaN in advance. The uplink transmission rate between TaN and CoN $n \in \mathrm{N}(t)$ can be denoted by $R_{t,n}^{(u)}$ in time period t. The uplink channel state between TaN and CoN n can be denoted by $H_{t,n}^{(d)}$. The interference power at

CoN n is $I_{t,n}^{(u)}$. Denote transmission power by P, channel bandwidth by W of TaN and the noise power by δ^2. Therefore, the uplink transmission rate between TaN and CoN n is

$$R_{t,n}^{(u)} = W\log_2(1 + \frac{PH_{t,n}^{(u)}}{\delta^2 + I_{t,n}^{(u)}}) \tag{2}$$

The downlink transmission rate can be given by

$$R_{t,n}^{(d)} = W\log_2(1 + \frac{PH_{t,n}^{(d)}}{\delta^2 + I_{t,n}^{(d)}}) \tag{3}$$

The downlink channel state between CoN n and TaN is denoted by $H_{t,n}^{(d)}$. The interference power at the TaN is denoted by $I_{t,n}^{(d)}$. The total transmission delay $D_t(t,n)$ can be written as

$$D_t(t,n) = \frac{x_t}{R_{t,n}^{(u)}} + \frac{y_t}{R_{t,n}^{(d)}} \tag{4}$$

In the real systems of USV fog cluster, both $R_{t,n}^{(u)}$ and $R_{t,n}^{(d)}$ cannot be known to the TaN in advance.

3.5.3. Offloading Delay

In time period t, the total offloading delay $D(t,n)$ is the computation delay $D_c(t,n)$ plus the transmission delay $D_t(t,n)$. If conditions permit, TaNs can also perform partial computation tasks. So for TaNs, only the computation delay $D_c(t,n)$ should be considered, and A represents a set of TaNs.

$$D(t,n) = \begin{cases} D_c(t,n), n \in A \\ D_c(t,n) + D_t(t,n),\ n \notin A \end{cases} \tag{5}$$

4. Problem Formulation

The TaN can formulate a task migration strategy A_t that allocates a computation amount to each CoN to minimize offloading delay. Therefore, the problem can be formulated as

$$D(t,A_t) = \max\{D(t,1), D(t,2), \ldots, D(t,n)\} \tag{6}$$

$$\begin{aligned} \text{P1 : min} \quad & \tfrac{1}{T}\sum_{t=1}^{T} D(t,A_t) \\ & \text{s.t. } D(t,A_t) \le t_0, \forall t \end{aligned} \tag{7}$$

For the task migration strategy A_t, the offloading delay $D(t,A_t)$ is the maximum of the delays of available computing nodes.

If the exact values of relevant parameters, such as computing capability $F(t,n)$, uplink transmission rates $R_{t,n}^{(u)}$ and downlink transmission rates $R_{t,n}^{(d)}$ of all available CoNs, can be known to TaN in advance, it is not difficult to calculate the offloading delay $D(t,n)$.

However, in real systems and marine operations, due to the relative movement of the USV cluster nodes, the state and interference of the wireless communication channel will change rapidly with time, and the resources of the CoN n may be shared by several computing tasks simultaneously. Therefore, the transmission rate and computing capability will also change rapidly with time. The above parameters are difficult to predict in advance. Furthermore, if each TaN requests the relevant parameters of all CoNs in each time period t, the signaling overhead between TaN and CoN will be much high.

When making the current offloading decisions, TaNs do not know the performance of the CoN in advance. We design an optimized learning-based computation task offloading mechanism that helps TaNs learn the offloading performance of available CoNs through historical data.

5. Proposed Algorithms

5.1. Problem Analysis

We develop an optimized learning-based offloading algorithm which enables TaN to learn the offloading performance of available CoNs to minimize the average delays. In the same computing task, it can be assumed that there are various different input data sizes, and the ratio of the output data size to the input data size is kept constant throughout the time. It is a reasonable assumption when the computing task types are the same. Therefore, let $y_t/x_t = \alpha_0$ and $\omega_t = \omega_0$ for $\forall t$.

Then the offloading delay can be defined as

$$u(t,n) = \begin{cases} \omega_0 F(t,n), n \in A \\ \frac{1}{R_{t,n}^{(u)}} + \frac{\alpha_0}{R_{t,n}^{(d)}} + \omega_0 F(t,n), n \notin A \end{cases} \tag{8}$$

$u(t,n)$ is the total delay of offloading one bit data to CoN n in time period t, which reflects the average offloading performance of CoN. The offloading delay is

$$D(t,n) = x_t u(t,n) \tag{9}$$

In a real system, the size of the input data x_t can be known to TaN, when making offloading decisions at time t. However, for $\forall n \in \mathrm{N}(t)$, it is impossible to know the exact value of $u(t,n)$ and its distribution in advance. This requires the TaN to learn to get a relatively accurate estimate.

5.2. Optimized Algorithms

The problem is similar to MAB problem. In our proposed framework, TaN can be considered the player and each available CoN corresponds to an action with unknown loss distribution. The player then decides which combination of bets should be taken to minimize the average loss and achieve considerable gains.

The main challenge of classic MAB problem is whether it can effectively balance the trade-off between exploration and exploitation (EE) by exploring different operations to understand the relatively accurate estimates of each distribution.

Several excellent algorithms have been proposed to solve MAB problems, such as the UCB1 and UCB2 algorithms based on the upper confidence bound (UCB). The MAB mechanism has been adopted in wireless communications to learn unknown environments, including channel access and mobility management [31].

The proposed problem is similar to the classic MAB problem. However, there are still several problems that need to be solved. First, we need a combination offloading strategy of USV fog cluster nodes to achieve higher performance rather than only one best node.

Second, the available CoNs set $\mathrm{N}(t)$ changes with time, but the number of actions is fixed in classic MAB problems. The computing resources of CoNs may available or unavailable within the communication coverage of TaN during task execution, and the computing resources of candidate CoNs may be unavailable, causing a dynamic selection space. Therefore, existing solutions need to be optimized to fully and effectively utilize the empirical information of the remaining CoNs.

Finally, in the classic MAB problem, the performance loss is equal in each time period. In the proposed algorithms, we introduce a weighting factor for the input data size x_t. It enables the algorithm exploit more when x_t is high and explore more when x_t is low, to reduce exploration costs and achieve balanced computing performance.

For each computing node, we set the upper and lower computing thresholds of the input data size to x_p^+ and x_p^-. When x_t is higher than x_p^+, the computing capability is saturated. When it is lower than x_p^-, the computing capability is excessive.

$$\hat{x}_t = \max\left\{\min\left(\frac{x_t - x_p^-}{x_p^+ - x_p^-}, 1\right), 0\right\} \tag{10}$$

Therefore, we propose an AUCB algorithm, as follows (see Algorithm 1).

Algorithm 1 Adaptive UCB algorithm for computation task offloading

```
1: Input: $\alpha_0$,$\omega_0$,$\beta$,$x_p^+$,$x_p^-$
2: for $t$ = 1, ... ,$T$ do
3:     if Any CoN $n$ is new computation fog node or One node's performance does not meet the requirements then
4:         Update available CoN list once
5:         Update new computation node $u(t,n) = \max\{u(t,1), u(t,2), \ldots, u(t,n-1)\}$,$k_{t,n} = 1$,$t_n = t$
6:     else
7:         Observe $x_t$
8:         Calculate the computing performance coefficient of each candidate CoN $n \in \mathrm{N}(t)$:
9:         $\hat{u}_{t,n} = \overline{u}_{t-1,n} + \sqrt{\frac{\beta(1-\hat{x}_t)\ln(t-t_n)}{k_{t-1,n}}}$
10:        Offload the task to CoNs:
11:        $A_t = \arg\min D(t, A_t)$
12:        Observe delay $D(t, A_t)$
13:        Update $\overline{u}_{t,a_t} \leftarrow \frac{\overline{u}_{t-1,a_t} k_{t-1,a_t} + u_{t,a_t}}{k_{t-1,a_t}+1}$
14:        Update $k_{t,a_t} \leftarrow k_{t-1,a_t} + 1$
15:    end if
16: end for
```

The proposed algorithms consider the appearance time of new CoN n and the input data size x_t, and can dynamically adjust the exploration weight and introduce the load-awareness and occurrence-awareness during task offloading. The computation offloading strategy is made on line 11. In the algorithm design, when the system communication characteristics change significantly, the computation offloading count will be reset.

6. Simulations and Performance Analysis

6.1. Performance Analysis

In real systems, reliable allocation algorithms can make performance loss $\overline{\rho}_T$ stable and acceptable without known relevant current parameters in advance. The average delay of optimal solution is denoted by $\overline{D}^*$ within time period T.

$$\overline{\rho}_T = \frac{1}{T}\sum_{t=1}^{T} D(t, A_t) - \overline{D}^*_T \tag{11}$$

$$\delta_T = \frac{1}{\overline{D}^*_T}\frac{1}{T}\sum_{t=1}^{T} D(t, A_t) - 1 \tag{12}$$

Similarly, percentage of performance loss is denoted by δ_T. In the next section, we design several simulations to evaluate the performance of the proposed AUCB algorithm in both heavy and light input data load conditions, which varies across time.

The idea of the bandit algorithms is to see how much regret the choice will bring, and the less regrets the better. In the MAB problem, the indicator used to measure good or bad is cumulative regret or performance loss. By simulating the same number of times with different selected algorithms, we can compare the cumulative regret growth rate of selected algorithms. If an algorithm is better, its cumulative regret growth is also slower than others.

6.2. Simulations

We design corresponding simulations to evaluate the offloading delay and performance loss of the proposed algorithm. We rationalize the corresponding parameters to a certain extent to simplify the calculation process.

In our simulations, x_t and $u(t,n)$ varies across time in a natural way with an initial value. In the series of $u(t_0,n)$, the first one is cloud node, others are fog cluster nodes. Next, we will analyze the performance of the proposed algorithm from multiple perspectives and assess the impact of changes in the main parameters. Main parameters' values in simulations are shown in Table 1. Table 2 shows the performance settings of CoNs, and the available or unavailable time of computing resources.

Table 1. Parameter values in simulations.

Parameter	Value	Unit
β	2	
x_p^-, x_p^+	1.8, 2.2 1.6, 2.4 1.2, 2.8	Mbit
x_0	2	Mbit

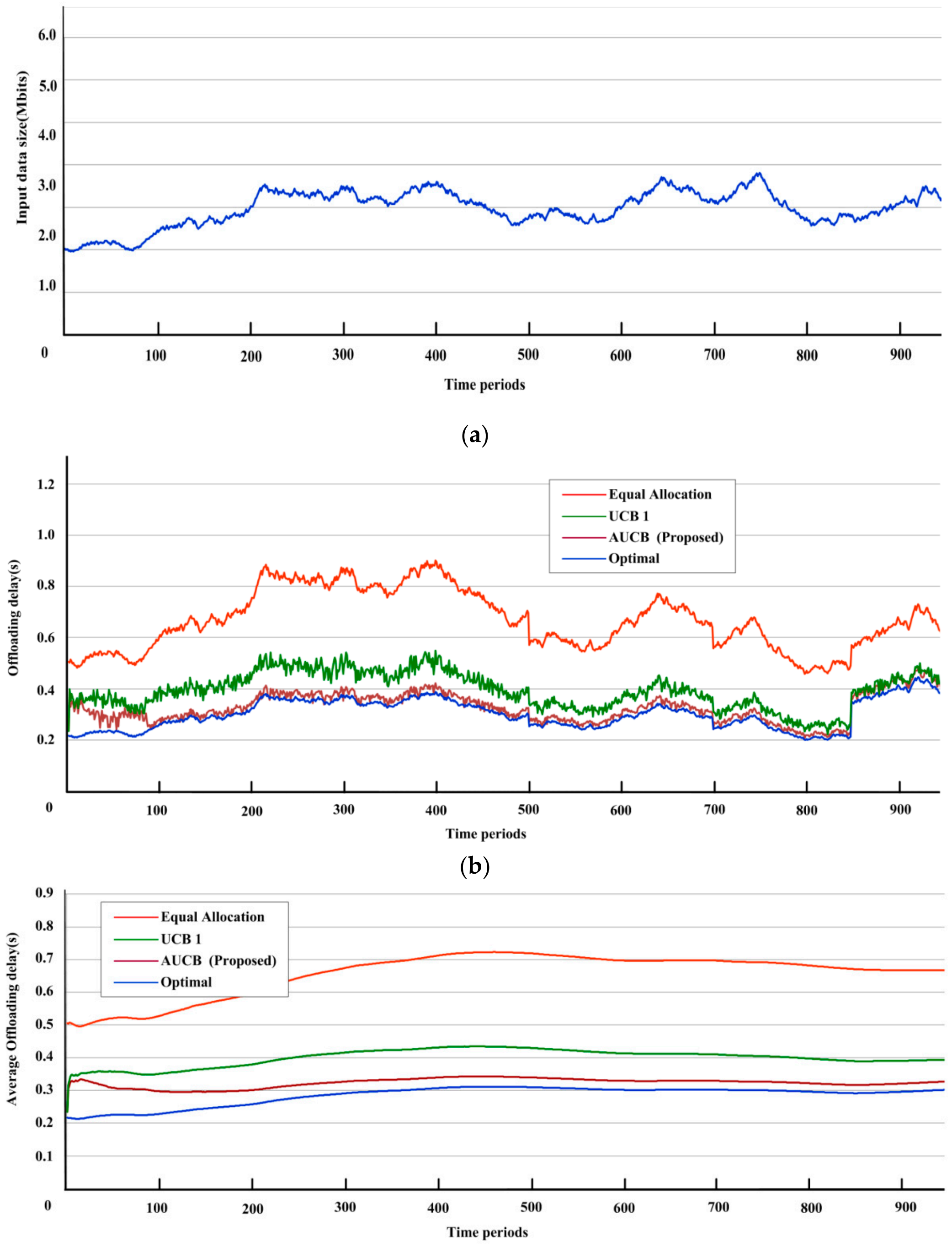
Table 2. Performance settings of CoNs.

Index of CoNs	Recourse Available Time	Recourse Unavailable Time	$u(t_0,n)$ (s/Mbit)
1	1	850	0.2
2	1	—	0.75
3	1	—	1.25
4	500	—	0.65
5	1	—	0.85
6	1	—	1.0
7	700	—	0.5

6.2.1. Performance of Selected Algorithms under Heavy Data Load Conditions

Figure 4a shows the input data size varies across time in a natural way with initial value of 2 Mbits. For most of the time, the data input load is large. Figure 4b shows the performance of the selected algorithms under diverse resource available time of CoNs. In our simulations, we set up different scenes to evaluate the effect of available time and unavailable time.

In the first epoch, two CoNs indexed by 4 and 7 are available, while in the next epoch, remote cloud node is unavailable. The results show that the proposed algorithm can learn the task offloading performance of the newly appeared CoNs more quickly, and effectively utilize the performance of the remaining CoN when one node is unavailable. It can reduce performance loss by about 70% compared to the UCB1 algorithm. The average task offloading delay is shown in Figure 4c, where the average performance of AUCB converges faster than UCB1 and other selected algorithms to achieve the optimal offloading performance.



(**a**)

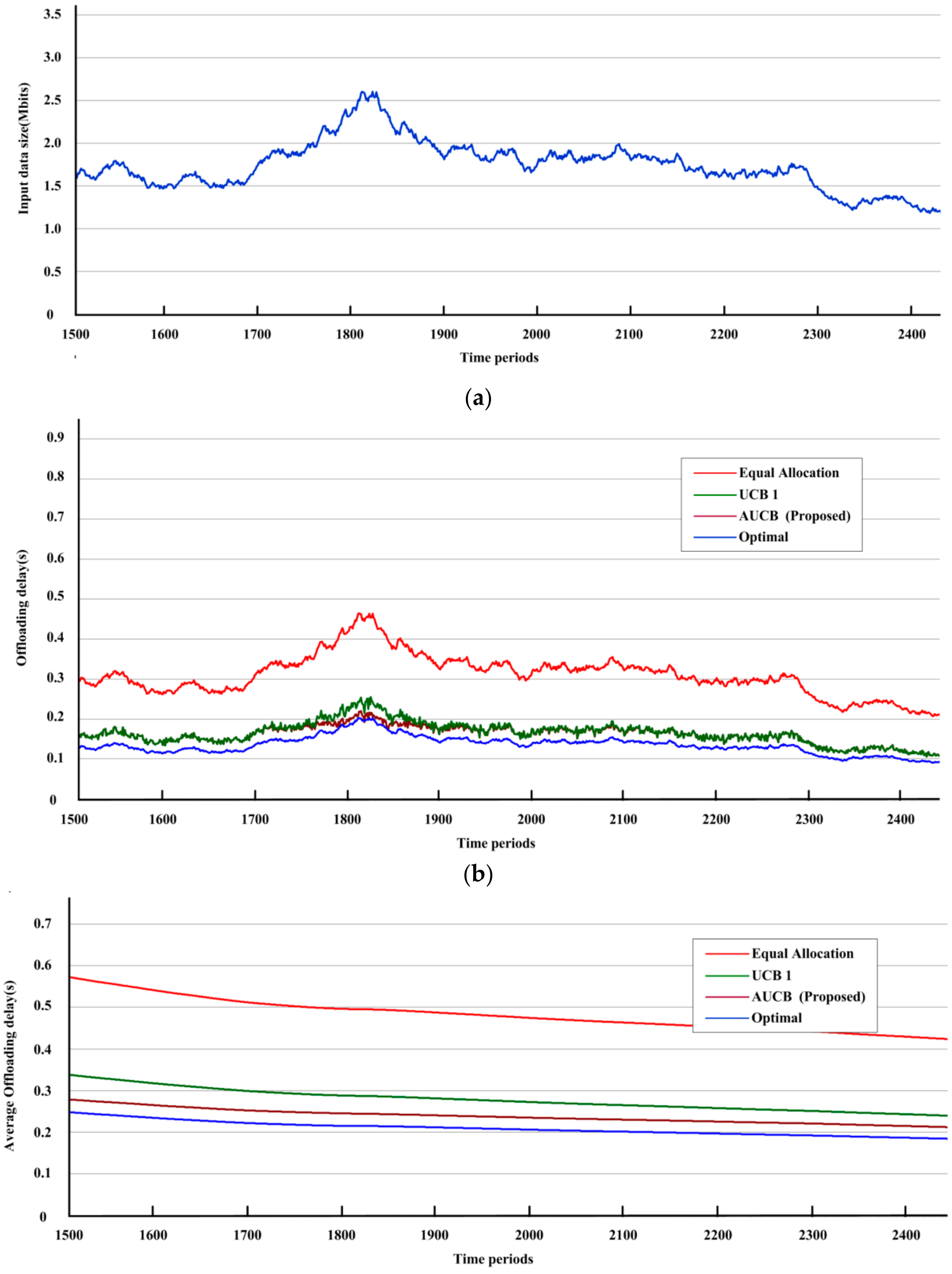
(**b**)

(**c**)

Figure 4. Computing performance of selected algorithms under heavy data load conditions. (**a**) Input data load. (**b**) Offloading delay. (**c**) Average offloading delay.

6.2.2. Performance of Selected Algorithms under Comprehensive Conditions

Figure 5a also shows the input data size varies across time, but for most of the time, the data input load is light. Figure 5b shows the performance of the selected algorithms under comprehensive conditions without any changes of cluster nodes. Under this condition, the proposed algorithm and UCB1 can achieve similar results, but when data input load raises, the proposed algorithm will get better performance than UCB1 immediately. The simulations show that the proposed algorithm achieves better performance than other algorithms, and can obtain more optimized values faster, especially under heavy data load conditions.



(**a**)

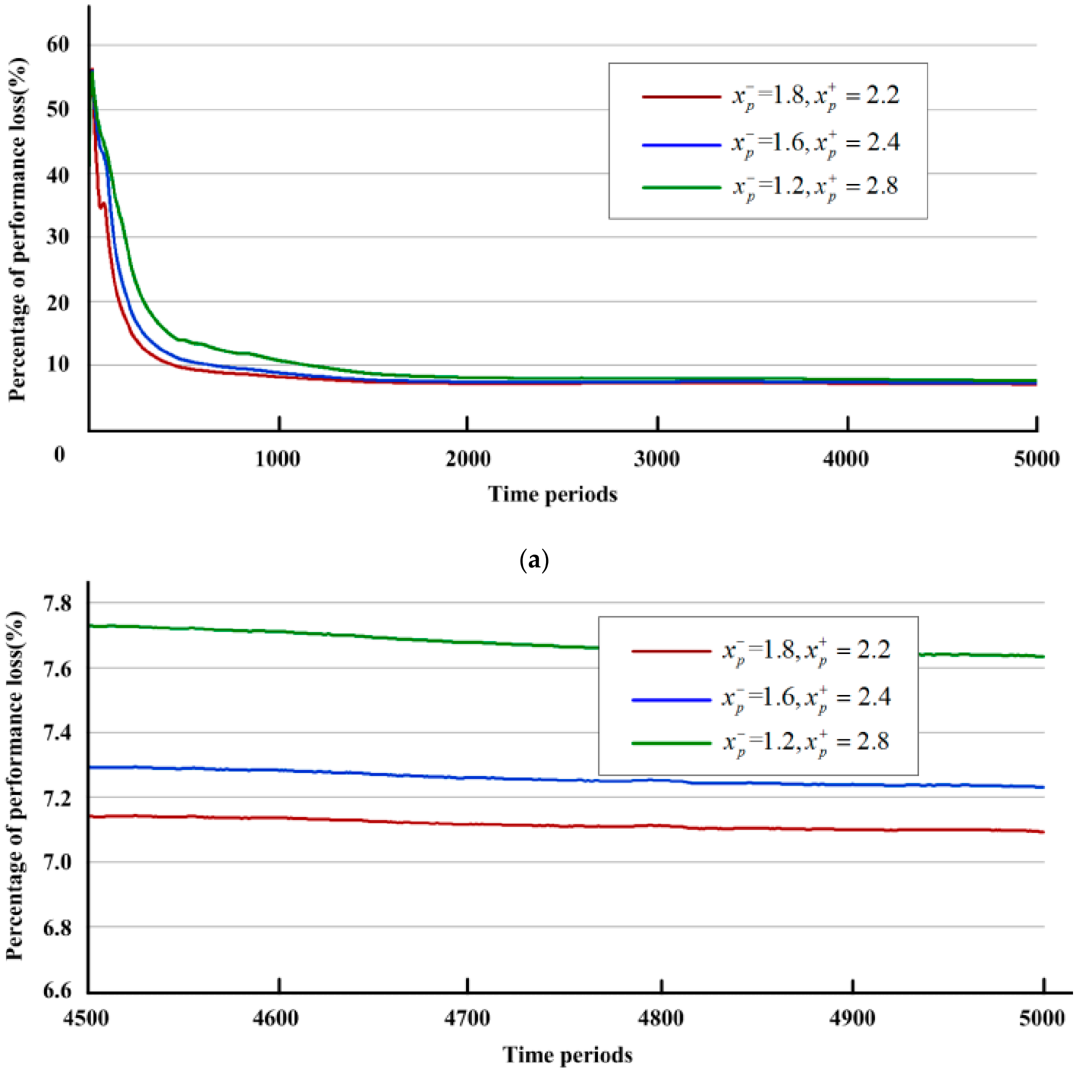
(**b**)

(**c**)

Figure 5. Computing performance of selected algorithms under comprehensive conditions. (**a**) Input data load. (**b**) Offloading delay. (**c**) Average offloading delay.

6.2.3. Performance Loss Comparison of Computing Thresholds Settings

Figure 6a shows that different computing check thresholds settings may affect the computing performance of the proposed algorithm at the beginning of simulations, but as time progresses, the difference of performance becomes very small and is ignorable.



(**a**)

(**b**)

Figure 6. Percentage of performance loss compared with optimal values. (**a**) Percentage of performance loss (time periods: 0–5000); (**b**) Percentage of performance loss (time periods: 4500–5000).

Figure 6b shows that the proposed algorithm achieves near-optimal computation offloading performance. At the end of the simulation period *T*, its performance loss is less than 8% of the optimal delay, which is superior to other selected algorithms.

The simulation results show that the proposed AUCB algorithm can still effectively balance the trade-off between exploration and exploitation under both heavy and light input data load conditions. The algorithm performance is relatively good compared to the optimal solution and can provide a bounded deviation.

By introducing the recourse available or unavailable time of CoN and the normalized input data size, AUCB algorithm is both load-aware and occurrence-aware. The computing performance is also relatively good.

7. Conclusions

In this work, we propose a marine fog-cloud computing architecture for USV clusters, and study the computation task offloading problem in the marine vehicular computing system architecture. We develop an optimized learning-based computation task offloading mechanism based on classic MAB theory. It enables vehicles to learn the potential offloading performance of their neighboring team nodes to minimize average computation task offloading delay. The simulation results show that computing performance of proposed AUCB algorithm is relatively good under both heavy and light input data load conditions.

Next, we will build a more detailed model or framework, and the corresponding simulation environment, to support the optimization of the proposed algorithm in order to obtain better performance. At that stage, the simulation platform will satisfy more complex tasks' requirements, while retaining the further expansion and exploration capabilities.

Author Contributions: Validation, visualization and writing—original draft, K.C.; conceptualization, data curation, formal analysis, writing—review and editing, B.L.; methodology, W.S. (Wenli Sun) and W.S. (Wenqiang Sun).

Funding: This work is fully sponsored by First-Class Discipline Open Research Foundation (2019) of Navigation College, Dalian Maritime University. This work was also supported by National Natural Science Foundation of China under 61971083 and 51939001 and Dalian Science and Technology Innovation Fund under 2019J11CY015.

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

References

1. Cui, K.T.; Sun, W.L.; Sun, W.Q. Joint computation offloading and resource management for USV cluster of fog-cloud computing architecture. In Proceedings of the IEEE International Conference on Smart Internet of Things, Tianjin, China, 9–11 August 2019.
2. Aazam, M.; Zeadally, S.; Harras, K.A. Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities. *Future Gener. Comp. Syst.* **2018**, *87*, 278–289. [CrossRef]
3. Yang, T.T.; Zhang, Y.; Dong, J. Collaborative optimization scheduling of maritime communication for mobile edge architecture. In Proceedings of the IEEE International Conference on RFID Technology & Applications, Macau, China, 26–28 September 2018.
4. Yang, T.T.; Cui, Z.Q.; Wang, R.; Zhao, J.; Su, Z.; Deng, R.L. A multi-vessels cooperation scheduling for networked maritime fog-ran architecture leveraging SDN. *Peer Peer Netw. Appl.* **2018**, *11*, 808–820. [CrossRef]
5. Khoda, M.E.; Razzaque, M.A.; Almogren, A.; Hassan, M.M.; Alamri, A.; Alelaiwi, A. Efficient computation offloading decision in mobile cloud computing over 5G network. *Mob. Netw. Appl.* **2016**, *21*, 777–792. [CrossRef]
6. Niu, R.F.; Song, W.F.; Liu, Y. An energy-efficient multisite offloading algorithm for mobile devices. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 72–81. [CrossRef]
7. Goudarzi, M.; Zamani, M.; Haghighat, A.T. A genetic-based decision algorithm for multisite computation offloading in mobile cloud computing. *Int. J. Commun. Syst.* **2016**, *30*, 1–13. [CrossRef]
8. Enzai, N.I.M.; Tang, M.L. A heuristic algorithm for multi-site computation offloading in mobile cloud computing. *Procedia Comput. Sci.* **2016**, *80*, 1232–1241. [CrossRef]
9. Pu, L.J.; Chen, X.; Xu, J.D.; Fu, X.M. D2D Fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 3887–3901. [CrossRef]
10. Li, Y.J.; Wang, W.Y. Can mobile cloudlets support mobile applications. In Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 1060–1068.
11. Chen, M.; Hao, Y.X.; Qiu, M.K.; Song, J.; Wu, D.; Humar, I. Mobility-aware caching and computation offloading in 5G ultra-dense cellular networks. *Sensors* **2016**, *16*, 974. [CrossRef]
12. Sun, Y.H.; Peng, M.G.; Mao, S.W. Mobility-aware caching and computation offloading in 5G ultra-dense cellular networks. *IEEE Internet Things J.* **2019**, *6*, 1960–1971. [CrossRef]
13. Zhang, H.J.; Qiu, Y.; Long, K.P.; George, K.K.; Wang, X.B.; Nallanathan, A. Resource Allocation in NOMA-Based Fog Radio Access Networks. *IEEE Wirel. Commun.* **2018**, *25*, 110–116. [CrossRef]
14. Liu, L.Q.; Chang, Z.; Guo, X.J.; Mao, S.W.; Ristaniemi, T. Multiobjective optimization for computation offloading in fog computing. *IEEE Internet Things J.* **2018**, *5*, 283–295. [CrossRef]
15. Sun, Y.X.; Guo, X.Y.; Song, J.H.; Zhou, S.; Jiang, Z.Y.; Liu, X.; Niu, Z.S. Adaptive learning-based task offloading for vehicular edge computing systems. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3061–3074. [CrossRef]

16. Li, J.; Gao, H.; Lv, T.; Lu, Y. Deep reinforcement learning based computation offloading and resource allocation for MEC. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference, Marrakech, Morocco, 15–19 April 2019.
17. Chen, X.F.; Zhang, H.G.; Wu, C.; Mao, S.W.; Ji, Y.S.; Bennis, M. Performance optimization in mobile-edge computing via deep reinforcement learning. In Proceedings of the 2018 IEEE 88th Vehicular Technology Conference, Chicago, IL, USA, 27–30 August 2018.
18. Jin, X.M.; Liu, Y.A.; Fan, W.H.; Wu, F.; Tang, B.H. Multisite computation offloading in dynamic mobile cloud environments. *Sci. China Inf.* **2017**, *60*, 089301. [CrossRef]
19. Shah-Mansouri, H.; Wong, V.W.S. Hierarchical fog-cloud computing for IoT systems: A Computation Offloading Game. *IEEE Internet Things J.* **2018**, *5*, 3246–3257. [CrossRef]
20. Mukherjee, M.; Shu, L.; Wang, D. Survey of fog computing: Fundamental, network applications, and research challenges. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 1826–1857. [CrossRef]
21. Sarkar, S.; Chatterjee, S.; Misra, S. Assessment of the suitability of fog computing in the context of Internet of Things. *IEEE Trans. Cloud Comput.* **2018**, *6*, 46–59. [CrossRef]
22. He, J.H.; Wei, J.; Chen, K.; Tang, Z.Y.; Zhou, Y.; Zhang, Y. Multitier fog computing with large-scale IoT data analytics for smart cities. *IEEE Internet Things J.* **2018**, *5*, 677–686. [CrossRef]
23. Gu, Y.N.; Chang, Z.; Pan, M.; Song, L.Y.; Han, Z. Joint radio and computational resource allocation in IoT fog computing. *IEEE Trans. Veh. Technol.* **2018**, *67*, 7475–7484. [CrossRef]
24. Sun, Y.X.; Guo, X.Y.; Zhou, S.; Jiang, Z.Y.; Liu, X.; Niu, Z.S. Learning-based task offloading for vehicular cloud computing systems. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–7.
25. Pinto, M.F.; Marcato, A.L.M.; Melo, A.G.; Honório, L.M.; Urdiales, C. A framework for analyzing fog-cloud computing cooperation applied to information processing of UAVs. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 1–14. [CrossRef]
26. Xiao, F.; Xie, X.H.; Jiang, Z.F.; Sun, L.J.; Wang, R.C. Utility-aware data transmission scheme for delay tolerant networks. *Peer Peer Netw. Appl.* **2016**, *9*, 936–944. [CrossRef]
27. Ning, Z.L.; Huang, J.; Wang, X.J. Vehicular fog computing: Enabling real-time traffic management for smart cities. *IEEE Wirel. Commun.* **2019**, *26*, 87–93. [CrossRef]
28. Aazam, M.; Zeadally, S.; Harras, K.A. Deploying fog computing in industrial internet of things and industry 4.0. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4674–4682. [CrossRef]
29. Cui, K.T.; Yang, Z.C.; Sun, W.L. The collaborative autonomy and control framework for unmanned surface vehicle. In Proceedings of the 2015 Ninth International Conference on Frontier of Computer Science and Technology, Dalian, China, 26–28 August 2015.
30. Mao, Y.Y.; You, C.S.; Zhang, J.; Huang, K.B.; Letaief, K.B. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [CrossRef]
31. Sun, Y.X.; Zhou, S.; Xu, J. EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 2637–2646. [CrossRef]