Received June 28, 2020, accepted July 2, 2020, date of publication July 7, 2020, date of current version July 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007669

Resource Management of Maritime Edge Nodes for Collected Data Feedback

YANLI XU, **(Member, IEEE)**
Department of Information Engineering, Shanghai Maritime University, Shanghai 201306, China
e-mail: ylxu@shmtu.edu.cn

This work was supported in part by the National Natural Science Foundation of China under Grant 61601283 and Grant U1701265.

ABSTRACT With the development of marine economy, more marine applications emerge leading to more deployed sensors and more generated data. However, current marine Internet only provides low transmission rate constrained by limited communication resources and bad transmission environment. How to efficiently use these resources and guarantee quality-of-service (QoS) of applications need to be addressed. This paper proposes an online optimization resource management algorithm to improve the communication resource efficiency with guaranteed QoSs for different applications. This algorithm needs neither of the resource cost function nor QoS constraint function at the resource scheduling node. Furthermore, the gradient information, which is usually needed in learning strategies, is not required either. Instead, this algorithm performs resource management of both computation and communication resources in each time slot only based on the observation of the last time slot. With slot-by-slot resource allocation, the communication cost can be minimized to be suitable for maritime scenarios. In the meantime, a long-term delay constraint can be satisfied. Results show that the proposed algorithm achieves the goal of reducing communication cost while guaranteeing the delay-constraint of different applications. Although more computation resources will improve its performance, this algorithm still can obtain the minimized cost given a low computation resource.

INDEX TERMS Online resource management, data feedback, edge computing, maritime communications.

I. INTRODUCTION

As the increase of human marine activities, more underwater devices are deployed such as various kinds of sensors and autonomous underwater vehicles (AUVs) for different applications such as fish monitoring, detection and safeguard [1]–[3]. The detected information of underwater devices is usually collected by devices at the edge-layer of the marine Internet such as buoys, ships, marine observation platforms and cellphones [4], [5]. These devices feed collected data back to remote data centers via wireless Internet access layer devices such as base stations (BSs), access points (APs) and satellites [6], [7]. Different from terrestrial communications, maritime communications are usually suffered from harsh environment and lacking infrastructure. As a result, the maritime communication channel is usually with low capacity. For example, the transmission rates of leo-Internet of things (IoT) satellite and automatic identification system (AIS) are usually tens of kHz to hundreds of kHz. For Wide-band techniques such as the fourth-generation (4G) cellular techniques and WiFi, the coverage is usually small compared to the required coverage of maritime communication scenarios. When integrate wide-band techniques into maritime communications, communication resources are mainly used for extending the coverage [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Irfan Ahmed.

To realize the feedback of booming data to application users with very limited maritime communication resources, a useful tool is resource management, which can appropriately adjust available communication resources such as transmission time and frequency bandwidth to different applications or users. Most of resource management schemes are designed for terrestrial Internet such as cellular networks and wireless local area network (WLAN). Some of the literature is based on coloring-graph algorithms. For example, in [9], authors use a vertex list-coloring graph for formulating the bandwidth allocation problem of multi-cells. A distributed coloring algorithm is proposed to maximize the use of the resources and try to keep fair among nodes. In [10], an interference graph is utilized to describe the sub-band allocation problem. To guarantee that any two neighbor device-to-device (D2D) users cannot use the same resources, an interference-degree based resource management algorithm is proposed. Similarly, the resource management problem of small cells are also considered based on list-coloring graph in [11]. Part of literature is based on the game theory. In general, via the aid of the game model, resource management decisions are regarded as actions of players, where resources such as power [12], sub-channels [13], spectrum [14] and transmission modes are allocated to maximize the utility or minimize the cost. Another method is based on the auction theory, which is a process of resource allocation and price discovery. For example, in [15], the sleep scheduling decision of BS is made base on auction. In [16], user association and scheduling problems in multi-cell multi-user multiple input multiple output (MIMO) networks are also determined by using auction and pricing. As the development of artificial intelligence (AI) and machine learning, some researches begin the study of utilizing AI for resource management. With the aid of AI, implicit characteristics such as communication channels, user demands and user distributions can be predicted. For example, in [17], the resource allocation strategy is formulated as a joint optimization problem with content caching. Then the decision on how to allocate the dynamic resources is performed by a deep learning approach. In [18], bacteria foraging algorithm is utilized to select the most appropriate combination of resource blocks for small cell users. In addition, some reinforcement learning methods such as Q-learning [19], [20] and multi-armed bandit (MAB) [21] are also utilized to improve the performance of resource management.

For maritime communications, most of common resource management schemes are designed for underwater networks. For example, in [22], time resources are determined optimally to maximize the overall channel utilization while preserving flow limitation and maintaining fairness for underwater networks. In [23], a spatial reuse resource allocation scheme is presented for underwater acoustic networks. This scheme schedules communications so as to avoid destructive collisions. In [24], authors propose a genetic algorithm (GA) based scheduling scheme to maximize the available bandwidth at each transmission period and satisfy the transmission requirements of both periodical and event-triggered information for the AUV. In [25], authors consider multi-hop underwater acoustic sensor networks suffering from low throughput caused by unreasonable bandwidth allocation. They propose a distributed traffic-based scheduling method to optimize the bandwidth allocation by letting the older packet be transmitted preferentially. In [26], the downlink of an underwater sensor network (USN) is considered, a joint sub-carrier allocation and bit loading algorithm is proposed to achieve identical data rate for each node in the USN while satisfy a targeted bit error rate.

As the development of underwater IoT, the amount of feedback data rapidly increases. This feedback problem is different with that of terrestrial internets and underwater internets due to following reasons. Compared with terrestrial internets, the communication bandwidth of over-water maritime internets is very limited, which restricts the feedback of abundant collected data at edge nodes. In addition, the feedback of data is affected by many parameters and the affecting function is difficult to obtain due to the complication of marine environment and the lack of appropriate general channel models. Compared with underwater networks, the edge nodes of network usually have stronger computation ability and computation energy, which is more suitable for performing edge computation. Due to the above differences, the problem formulation, constraint, target and available resources will be different with that of terrestrial internets and underwater internets. Under this situation, how to effectively schedule available resources at the edge node to improve the feedback efficiency with limited resources is still a problem to be solved. Considering some learning methods have natural advantages to solving problem without explicit functions, more strategies are provided to solve tough problems based on learning methods, such as MAB, Q-learning, natural networks, Support Vector Machine and ect.. In this paper, we will solve the resource scheduling problem of maritime data feedback based on the bandit learning method. Relative to existing resource management work, we consider the quality-of-service (QoS) requirements of different applications instead of that of individuals to design an appropriate resource management approach for the successful delivery of packets of different applications. Furthermore, computation resources are jointly scheduled to make the best of edge computation and minimize the communication cost. More importantly, assumptions on the acquisition of cost function and QoS constraint function are relaxed, which do not need to know many parameters about the network such as communication channels, computation patterns, network topology, and etc.. As a result, the operability and practicality of resource management can be improved. In addition, our proposed algorithm is online, which makes resource management only be based on the information of the last time slot. Thereby, it is not only having low overhead cost but also adjusting the resource management according to the communication environment and available resources dynamically. The remainder of this paper is organized as follows. Section II provides system model followed by the problem formulation of resource allocation. Section III proposes an online resource management scheme for data feedback with different QoS requirements. Section IV presents several numerical results to evaluate the proposed scheme and Section V concludes this paper.

II. SYSTEM MODEL

The notations and symbols used in this paper are listed in Table 1.

A. A MOTIVATING APPLICATION

This paper is motivated by the inconvenience and inefficiency of detection information feedback for maritime applications. For example, offshore aquaculture usually deploys sensors and cameras under water or on the water platform to monitor

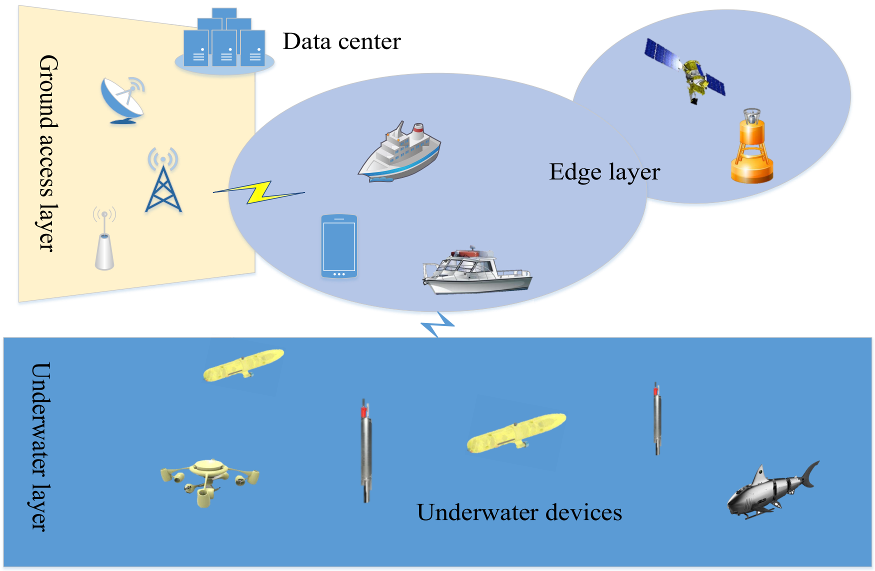

FIGURE 1. An example of maritime communication scenario.

TABLE 1. Symbols and Notations.

Symbol or Notation	Description
γ_i	data arrival rate of application i
K	number of applications
$B(t)$	the available bandwidth at time slot t
$B_i(t)$	the bandwidth allocated to application i at time slot t
$l_i(t)$	the data amount of i after compression at time slot t
$d_i^{co}(t)$	the communication latency of i at time slot t
$P_i(t)$	the allocated computation resources of i at time slot t
η_i	compression ratio per computation resource
L_i	the amount of new arrived data of application i at slot t
T	observation time
$d_i^{cp}(t)$	the computation latency of i at time slot t
$d_i(t)$	the total latency of i at time slot t
$\mathbf{x}_t$	resource allocation action set at t
$f(\mathbf{x}_t)$	cost function at t
$g(\mathbf{x}_t)$	constraint function at t
∇	gradient symbol
$\|x\|$	norm of x
$[x]^+$	$\max\{x,0\}$
A^T	the transposition of matrix A
$\mathbf{E}_x$	the expectation of x
λ_t	the Lagrangian multiplier at t
α and μ	step sizes for iteration
$\mathbf{B}_t$	the action set of bandwidth allocation at t.
$\mathbf{P}_t$	the action set of computation resource allocation at t

the situation of fish and avoid steal. However, it is very challenging for the edge node such as mobile phone and leo-IOT satellite with high latency, limited communication resources and finite computation capacity to feed enormous data back to farming enterprises. The current available approaches in practice are mainly reduced to three folds: i) feedback through leo-IOT satellite, which compresses video into low-pixels pictures and transmits them with several kHz rate, ii) feedback through existing infrastructure or opportunistic networks such as buoys or passing ships with uncertainty or high latency, and iii) send people to task locations for getting collected data back through 4G for nearsea aquaculture. In balance, the disadvantages of all above methods can be attributed to limited communication resources with low transmission rate and high latency. On the other hand, applications have different QoS requirements such as delay constraints. Some applications cannot tolerate such high latency especially for safety detection data.

One promising solution to the challenges is to compute some data at the edge node and appropriately schedule resources for each application. Meanwhile, take the fact that some applications have low-latency requirements into consideration. Since the maritime communication environment is complicated and there is few accurate channel modeling for maritime communications, it is better to perform resource management without explicit distribution expresses of channels, cost and constraint. In addition, it is necessary for the edge node to make real-time resource management decisions. It is expected that the decisions are made based on less overhead and information exchanges.

B. NETWORK MODEL

We consider a maritime communication network where underwater devices are deployed for monitor, detection and other applications (see Fig. 1 as an example). We focus on the feedback of collected data of an edge node such as buoy and ship, which collects data from nearby underwater devices. Underwater devices generate streams (a stream is the entity to which the QoSs are offered. It is a sequence of packets from a transmitter to a receiver) of different applications during our observation period T. Usually, the value of observation period value T of learning schemes is scenario related, which can be set based on cost and constraint requirements or empirical values. For example, we can use the value of one or multiple delay-QoS constraints here for the value of T. In addition, it can be optimized to maximize utility or minimize cost.

Similar to popular communication techniques such as WiFi and 4G, the transmission period is time slotted. We use a Poisson model to characterize the distribution of data generated by different applications, i.e., the arrival of application i follows Poisson distribution with rate γ_i. Different applications have different QoS requirements. For streams of K applications in the observed time slot t, the QoS requirements here use delay constraint as an example. Before feeding these data back to the data users, edge node may pre-process them to compress and reduce the data amount. However, the computation capacity of edge node is constrained leading that the compression of data usually experiences high latency, which may incur the violation of delay constraint. Thus, before discussing resource manage, we model resources at the edge node in the next subsection.

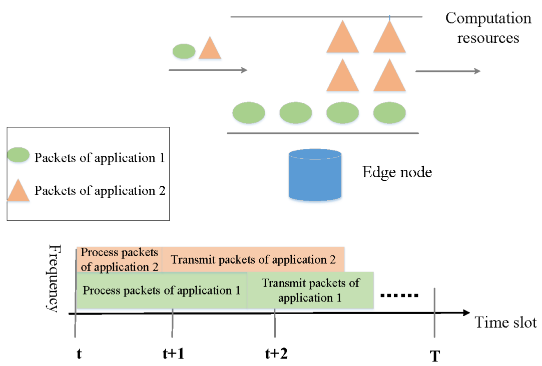

FIGURE 2. An example of two applications and their resource allocation.

C. RESOURCE AND COST MODEL

Fig. 2 gives an example of resource allocation. The available resources at an edge node include communication resources and computation resources. For the data transmission, the edge node can allocate different frequency bandwidth to different applications with total bandwidth $B(t)$ at t. Before the data transmission, the edge node also can allocate different computation resources to different applications. For application i, given the amount of transmission data l_i and allocated bandwidth B_i, the transmission tim and computation time can be determined. For example, as shown in Fig.2, application 2 is allocated with two computation process units while application 1 is allocated with one computation process unit. On the other hand, application 1 is allocated more frequency bandwidth than that of application 2. Here application 1 and 2 use 2 and 1 time slots for computation, respectively. In addition, application 1 and 2 use 1 and 2 time slots for data transmission, respectively. Since we do not constrict the processing and transmission to a time slot, the data processing and transmission which arrive at time slot t also can be processed and transmitted at time slots after t. However, the decision still can be made slot-by-slot. This assumption is made sense due to following reasons. We can know the amount of data computed and transmitted during slot t. Since the communication rate (like 4G) and computation rate is constant, the time used by communication and computation for data arriving at t can be derived. Thus, we can make decision at $t+1$ based on the information obtained at t without waiting for the accomplishment of transmission and computation. With allocated bandwidth $\{B_1(t), B_2(t) \cdots, B_K(t)\}$ at t for K applications, the communication latency of i is

$$d_i^{co}(t) = l_i(t) / h_0(B_i(t)), \tag{1}$$

where $l_i(t)$ is the amount of i after compression and $h_0(B_i(t))$ expresses the transmission rate with bandwidth B_i.

Here we do not specify a channel modeling for both of large-scale fading and small-scale fading since common maritime channel model such as physical models and stochastic models [27] are scenario-specified. The accuracy of these models is heavily discounted. Instead, we leave the channel model to be unknown and use learning method to provide a universal solution for the given optimal problem in this paper without being constrained by a specific channel model.

The compression of packets depends on the computation resource. As stated in [28], the computation cost is related to circuits, CPU frequencies, CPU cores, and memory sizes, which make different nodes have different computation ability. Given the circuit configuration of edge node, the amount of allocated computation resources leads to different compression ratio of data. Denote the allocated computation resources of i to be $P_i(t)$, the compression ratio η_i is a function of $P_i(t)$, $h(P_i(t))$. We do not specify the concrete expression of h, which will not affect our following analyses. With the compression, the amount of i can be written as

$$l_i(t) = L_i(t) * \eta_i = L_i(t) * h(P_i(t)), \tag{2}$$

where L_i is the amount of new arrived data of application i at slot t before compression. We do not assume that the stream size is restricted, i.e., the finished stream in one slot can be fractional, such that the delay computation does not need to be restricted in one time slot, which is more suitable for practical cases.

Let $d_i(t)$ denote the delay of i arriving at t, which includes communication delay (including scheduling delay here) and computation delay. As shown in (1), the communication delay is a function of $B_i(t)$ and $l_i(t)$. In addition, as shown in (2), $l_i(t)$ is a function of $P_i(t)$. Then, the communication delay can be expressed by a function of $B_i(t)$ and $P_i(t)$, say, $d_i^{co}(t) = z_1(B_i(t), P_i(t))$. The computation delay is a function of $P_i(t)$, which is expressed as $d_i^{cp}(t) = z_2(P_i(t))$. Thus, the total delay is

$$\begin{aligned} d_i(t) &= d_i^{co}(t) + d_i^{cp}(t) \\ &= z_1(B_i(t), P_i(t)) + z_2(P_i(t)), \end{aligned} \tag{3}$$

which is dependent on resource management decisions made in each slot.

D. PROBLEM FORMULATION

Intuitively, in order to reduce consumed communication resources, processing streams as many as possible will be beneficial. However, the QoS of applications (such as delay requirements, data rate and successful delivery probability) are different and computation resources are limited. To make full use of maritime communication resources, resources are managed at each slot in our scheme. The slot length can be optimized for different scenarios which is out of our scope and is fixed to be a constant in our work. At the beginning of each slot, edge node can collect new arrival application data, and determine the resource allocation for them. Denoting the resource allocation action set as $\mathbf{x}_t$, it consists of the allocated computation and communication resources for each application, i.e., $\mathbf{x}_t := \{B_1(t), P_1(t), \cdots, B_i(t), P_i(t), \cdots, B_K(t), P_K(t)\}$.

The object here is to minimize the average communication cost per stream. That is, the following cost

$$f(\mathbf{x}_t) = \sum_{t=1}^{T}\sum_{i=1}^{K} B_i(t)/L_i(t), \tag{4}$$

is targeted to be minimized.

Since there is a trade-off between computation delay and communication delay, and the total delay of an application is affected by both of them, the sum of computation delay and communication delay should be constrained to guarantee the QoSs of applications. Although data arrived in a time slot may experience different time slots for transmission and computation, the computation and communication time can be known based on information obtained at the time slot t. Then, in $t+1$, the learner can make a new decision based on these time delay and cost information obtained in t. Thus, the online total delay can be obtained slot by slot. The delay-constraint of each application can be written as $D_0 = \left\{d_1^0, \ldots d_K^0\right\}$. Denoting the constraint function as $\mathbf{g}_t(\mathbf{x}_t) = \left\{g_t^1(\mathbf{x}_t), \cdots, g_t^i(\mathbf{x}_t), \cdots, g_t^K(\mathbf{x}_t)\right\}$, the instantaneously considered constraint is not necessarily equal to the QoS threshold. Instead, a long-term constraint is common to ensure that the cumulative amount of QoS metric is no larger than the threshold at the edge node over time, i.e., $\sum_{t=1}^{T} d_i(t) \leq d_i^0$. The long-term constraint can be expressed by (5), where $g_t^i(\mathbf{x}_t)$ is the constraint of i. Given the total resources of applications at an edge node, the choice of resource allocation of different applications will affect with each other. That is, the choice of $B_i(t)$ and $P_i(t)$ will be affected by the value of $B_j(t)$ and $P_j(t)$ for all $i \neq j$. Thus, here the constraint of each application is a function of action set $\mathbf{x}_t$.

$$\sum_{t=1}^{T} g_t^i(\mathbf{x}_t) = \sum_{t=1}^{T} d_i(t) - d_i^0 \leq 0 \quad \text{for } i = 1 \cdots K. \tag{5}$$

III. ONLINE RESOURCE MANAGEMENT ALGORITHM

We target to realize online resource management, in which the edge node can determine resource allocation dynamically based on available resources, requirements of applications, prior cost and constraints. With the action set $\mathbf{x}_t$, the resource allocation decision of i is determined, which is denoted by x_i. With this allocation, the cost and constraint of application i at t can be obtained after the transmission from the following expressions,

$$f_t(x_i) = B_i(t)/L_i(t), \tag{6}$$

and

$$\begin{aligned} g_t^i(x_i) &= d_i(t) - \frac{d_i^0}{T} \\ &= z_1(B_i(t), P_i(t)) + z_2(P_i(t)) - \frac{d_i^0}{T}. \end{aligned} \tag{7}$$

Of course, we can use other metrics for $f_t(x_i)$ and $g_t^i(x_i)$, while the methodology illustrated follows will be similar. Clearly, the explicit expressions of $B_i(t)$ and $d_i(t)$ are unknown to the edge node. Thus, we assume that the detailed functions of both cost function f_t and constraint functions $\mathbf{g}_t$ are unavailable. Only the values of functions at a queried point are known since the edge node can observe the results of data delivery after the resource allocation and transmission. Although in our model, data may experience several time slots for computation or transmission, resources also can be allocated at the beginning of each slot based on the observation of the prior slot since the rates of transmission and computation of a given stream is invariant after its resource allocation.

For this problem, communication resources can be seen as a player, and its allocation can be regarded as player action. On the other hand, pre-process can be seen as an opponent, and its allocation can be regarded as its action. Their different actions lead to different results. However, they can only play an action at a time slot and their actions can only refer results observed from prior time slots. Thus, we can use online learning to solve this problem, where the resource allocation at each slot can be seen as an action set $\mathbf{x}_t$, which is decided by the learning of prior cost. The optimization problem based on above context is written as follows.

$$\begin{aligned} &\min f_t(\mathbf{x}_t) = \sum_{t=1}^{T}\sum_{i=1}^{K} B_i(t)/L_i(t) \\ &\text{s.t.} \sum_{t=1}^{T} z_1(B_i(t), P_i(t)) + z_2(P_i(t)) - d_i^0 \leq 0 \\ &\quad\ \text{for } i = 1 \cdots K. \end{aligned} \tag{8}$$

For the optimization objective $f_t(\mathbf{x}_t)$ in each slot with constraint $\mathbf{g}_t(\mathbf{x}_t)$, this online optimal problem can be solved step by step by using gradients of cost and constraint functions when the full information of functions can be obtained [29]. That is, given the primal iterate $\mathbf{x}_t$, the decision of next slot $\mathbf{x}_{t+1}$ is generated by

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in \mathbf{X}} \nabla_{\mathbf{x}}^{T} L_t(\mathbf{x}_t, \lambda_t)(\mathbf{x} - \mathbf{x}_t) + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}_t\|^2, \tag{9}$$

where α is a pre-defined constant for adjusting step size and $\mathbf{X}$ is the feasible set. $L_t(\mathbf{x}_t, \lambda_t)$ is the online Lagrangian of (8), which can be expressed as

$$L_t\left(\mathbf{x}_t, \lambda^T\right) = f_t\left(\mathbf{x}_t\right) + \lambda_t^T \mathbf{g}_t\left(\mathbf{x}_t\right). \tag{10}$$

The gradient of $L_t\left(\mathbf{x}_t, \lambda_t\right)$ with respect to the primal variable $\mathbf{x}$ at $\mathbf{x} = \mathbf{x}_t$ can be expressed as

$$\nabla_{\mathbf{x}} L_t\left(\mathbf{x}_t, \lambda_t\right) = \nabla f_t\left(\mathbf{x}_t\right) + \nabla^T \mathbf{g}_t\left(\mathbf{x}_t\right) \lambda_t. \tag{11}$$

Including (11) into (9), the minimization of (9) can be written as follows (the detailed derivations can refer to Appendix A).

$$\begin{aligned}\mathbf{x}_{t+1} &= \Phi_X\left(\mathbf{x}_t + \alpha \nabla_{\mathbf{x}} L_t\left(\mathbf{x}_t, \lambda_t\right)\right)\\ &= \Phi_X\left[\mathbf{x}_t + \alpha\left(\nabla f_t\left(\mathbf{x}_t\right) + \nabla^T \mathbf{g}_t\left(\mathbf{x}_t\right) \lambda_t\right)\right],\end{aligned} \tag{12}$$

where $\Phi_X(y) = \arg\min_{\mathbf{x} \in \mathbf{X}} \|\mathbf{x} - \mathbf{y}\|^2$ denotes the projection operator. In addition, the iteration of Lagrangian multiplier can be written as

$$\lambda_{t+1} = \left[\lambda_t + \mu\left(\mathbf{g}_t(\mathbf{x}_t) + \nabla^T \mathbf{g}_t(\mathbf{x}_t)\left(\mathbf{x}_{t+1} - \mathbf{x}_t\right)\right]^+, \tag{13}$$

where μ is a positive step size, and $\nabla_\lambda L_t\left(\mathbf{x}_t, \lambda_t\right) = \mathbf{g}_t\left(\mathbf{x}_t\right)$ is the gradient of the gradient of $\nabla_\lambda L_t\left(\mathbf{x}_t, \lambda\right)$ with respect to λ at $\lambda = \lambda_t$. $[x]^+ = \max\{x, 0\}$.

From (12) and (13), we can see that the player needs to know the gradient of $f_t(\mathbf{x}_t)$ and the constraint $\mathbf{g}_t(\mathbf{x}_t)$ at each slot t to make the decision at $t+1$. However, the functions of cost and constraints are unknown in our setup. Gradient-free methods will be leveraged to tackle this problem. The key idea here is using observations in t to estimate gradients. In the online setting, the functions change gradually over time and we only can evaluate each function once. Thus, we use a one-point estimate of the gradient in this paper. Different from stochastic gradient descent, stochastic gradient estimation evaluates gradient of all samples and estimate gradient without knowing the function of optimal problem. Intuitively, the performance of estimation will improve if multiple evaluations are available per time slot. As indicated in [30], the optimal is $d+1$. But this increase of information feedback will increase the overhead and delay. Although only one feedback is used here, it is certified that this one-point biased estimate is sufficient to approximate gradient descent on the sequence of functions in [31]. In addition, simulation results show that our scheme achieves similar performance with that of the scheme knowing the gradient. Thus, we use the value of $f_t(\mathbf{x})$ inquired at a single point $\mathbf{x}$ at a slot to estimate the gradient. In detail, for a random unit vector $\mathbf{u}$ with dimension d, and a small constant $\delta > 0$, the derivative of f_t at $\mathbf{x}$ can be approximated by [30]

$$\nabla f_t(\mathbf{x}) \approx \frac{d}{\delta} \mathbf{E}_{\mathbf{u}}\left[f_t\left(\mathbf{x} + \delta \mathbf{u}\right)\right]. \tag{14}$$

For example, for one-dimensional case ($d = 1$), (14) can be simplified as

$$\begin{aligned}\nabla f_t(x) &\approx \mathbf{E}_u\left[\frac{u}{\delta} f_t\left(x + \delta u\right)\right]\\ &= \frac{f_t(x+\delta) - f_t(x-\delta)}{2\delta}.\end{aligned} \tag{15}$$

Since the constraint function in our setup is also unknown, the gradient of constraint is similarly obtained by

$$\nabla \mathbf{g}_t\left(\mathbf{x}\right) \approx \frac{d}{\delta} \mathbf{E}_{\mathbf{u}}\left[\mathbf{g}_t\left(\mathbf{x} + \delta \mathbf{u}\right)\right]. \tag{16}$$

Based on (14) and (16), (12) can be approximated as

$$\begin{aligned}&\mathbf{x}_{t+1}\\ &= \Phi_{\beta \mathbf{X}}\left\{\mathbf{x}_t + \frac{\alpha d}{\delta}\left\{\mathbf{E}_{\mathbf{u}}\left[f_t\left(\mathbf{x}+\delta\mathbf{u}\right)\right] + \mathbf{E}_{\mathbf{u}}^T\left[\mathbf{g}_t\left(\mathbf{x}+\delta\mathbf{u}\right)\right]\lambda_t\right\}\right\},\end{aligned} \tag{17}$$

where $\beta \in [0, 1)$ is a pre-selected constant depending on δ (see Appendix B) and $\beta\mathbf{X}$ is a subset of $\mathbf{X}$. From (17), we see that the actually inquired point is $\mathbf{y}_t = \mathbf{x}_t + \delta \mathbf{u}_t$.

Similarly, we can obtain the recursion of Lagrange factor as

$$\lambda_{t+1} = \left[\lambda_t + \frac{d\mu}{\delta} \mathbf{E}_{\mathbf{u}}^T\left[\mathbf{g}_t\left(\mathbf{x} + \delta \mathbf{u}\right)\right]\left(\mathbf{x}_{t+1} - \mathbf{x}_t\right)\right]^+. \tag{18}$$

Based on (17) and (18), the player actions can be determined sequentially. Particularly, in our resource management problem, the steps of bandwidth and computation are as follows, where $\mathbf{B}_t$ and $\mathbf{P}_t$ denote communication resource allocation set and computation resource allocation set at slot t, respectively.

$$\begin{aligned}&\mathbf{B}_{t+1}\\ &= \Phi_{\beta X}\left\{\mathbf{B}_t + \frac{\alpha d}{\delta}\left\{\mathbf{E}_{\mathbf{u}}\left[f_t\left(\mathbf{B}_t+\delta\mathbf{u}\right)\right] + \mathbf{E}_{\mathbf{u}}^T\left[\mathbf{g}_t\left(\mathbf{B}_t+\delta\mathbf{u}\right)\right]\lambda_t\right\}\right\},\end{aligned} \tag{19}$$

and

$$\begin{aligned}&\mathbf{P}_{t+1}\\ &= \Phi_{\beta X}\left\{\mathbf{P}_t + \frac{\alpha d}{\delta}\left\{\mathbf{E}_{\mathbf{u}}\left[f_t\left(\mathbf{P}_t+\delta\mathbf{u}\right)\right] + \mathbf{E}_{\mathbf{u}}^T\left[\mathbf{g}_t\left(\mathbf{P}_t+\delta\mathbf{u}\right)\right]\lambda_t\right\}\right\}.\end{aligned} \tag{20}$$

The detailed proposed resource management algorithm based on above analyses is listed in Table 2, which is performed on the edge node. It shows that the online method of allocating resources, which minimize the communication cost with guaranteed QoSs of different applications. In the proposed online resource management scheme, resource allocation is performed at the edge node slot by slot to assign appropriate computation and communication resources to data of different applications collected by the edge node. To achieve this goal, edge node firstly selects a random slot as the first time slot, and the edge node will randomly select primate values of allocated computation resource and communication resource. Then the transmission delay and communication delay can be obtained based on the amount of transmitted data and computed data during the observed time slot. These time delay and occupied resources will be treated as cost and constraint, which will be used for making decision in the next time slot based on (19) and (20). When the data process ability and communication ability change,

TABLE 2. Resource Management Algorithm.

1: Initialize: primal iterate $\mathbf{y}_1 = \{B_i(1), P_i(1)\}_{i=1}^{K}$, λ_1, parameters δ and β, step size α and u.
2: For $t = 1, 2, \cdots$ do
3: The edge node determines $\mathbf{x}_t = \{B_i(t), P_i(t)\}_{i=1}^{K}$ based on iteration value $\mathbf{y}_t$ through $\mathbf{x}_t = \mathbf{y}_t + \delta\mathbf{u}_t$.
4: The edge node collects the cost value $f_t(\mathbf{x}_t)$ and constraint value. $\mathbf{g}_t(\mathbf{x}_t)$ at the queried point.
5: The edge node updates the dual variable λ_t via (18).
6: The edge node updates the $\mathbf{B}_{t+1}$ and $\mathbf{P}_{t+1}$ by (19) and (20),
7: respectively
8: End

the edge node can observe them from prior slots, and then the edge node updates them for the following cost and constraint calculation. With step by step resource allocation as shown in step 6, the online optimal problem is transferred into offline minimization problem, i.e., calculating allocated resources only with the aid of the result of resource allocation at t. It does not need the future information although it tries to obtain a guaranteed long-term QoS.

The offline minimization problem can be seen as the least distance problem (LDP) and its solution is listed in the Table 3 [32].

TABLE 3. Least Distance Problem Solving Steps.

1: Define $K+1 \times K+1$ matrix to be $\mathbf{A} = [-\mathbf{1}_{1\times K} \quad \mathbf{I}]^T$ and $K+1$ vector to be $\mathbf{e} = [0, \cdots, 0, 1]^T_{1\times K+1}$. Solve $\widehat{\mathbf{w}} = \min \|\mathbf{A}\mathbf{w} - \mathbf{e}\|$, where $\widehat{\mathbf{w}} \geq \mathbf{0}$ is the solution of non-negative least squares (NNLS), the derivation of which is listed in Table 4.
2: Compute $K+1$ vector $\mathbf{r} = \mathbf{A}\widehat{\mathbf{w}} - \mathbf{e}$.
3: If $\|\mathbf{r}\| = 0$, set $\psi = \mathbf{F}$ and goto step 7.
4: Set $\psi = \mathbf{T}$.
5: For $j = 1 \cdots K$
6: Compute $\widehat{x}_j = -r_j / r_{K+1}$.
7: End

When ψ = F, it expresses that there is no solution to the current constraint. Otherwise, set ψ = T. The solution of $\widehat{w}$ can be calculated by steps listed in the Table 4 [32], which gives a solution of non-negative least squares (NNLS) problem as follows.

$$\begin{aligned} &\min \quad \|\mathbf{Ax} - \mathbf{e}\| \\ &\text{s.t. } \mathbf{x} \geq \mathbf{0}. \end{aligned} \tag{21}$$

Until now, the resource management problem is solved. However, it is not clear the degree of the constraint satisfaction. In some practical scenarios, delay guarantee is important, e.g., safety detection. To evaluate the degree of constraint satisfaction, fit metric is usually used [33], which measures the accumulated violation of constraints, and is defined as follows.

$$\text{Fit}_T = \left\| \sum_{t=1}^{T} \mathbf{g}_t(\mathbf{x}_t) \right\|. \tag{22}$$

From the definition, we can see the smaller Fit_T is, the constraint violation is smaller. Inspired by the classic online convex optimization framework of Zinkevich [34], the online optimal problem with constraint is solved step by step here

TABLE 4. Non-negative least squares Solution.

1: Set $\psi = \emptyset$, $\mathbf{Z} = 1, 2 \cdots, K$, and $\mathbf{x} = 0$.
2: Compute the K vector $\mathbf{w} = \mathbf{A}^T(\mathbf{e} - \mathbf{Ax})$.
3: If the set Z is empty or if $w_j \leq 0$ for all $j \in \mathbf{Z}$, go to step 12.
4: Find an index $l \in Z$ such that $w_l = \max\{w_j : j \in \mathbf{Z}\}$.
5: Move the index l from set Z to set ψ.
6: Let $\mathbf{A}_\psi$ denote the $m_2 \times K$ matrix defined by $C(\mathbf{A}_\psi) = \begin{cases} C(\mathbf{A}) & j \in \psi \\ 0 & j \in \mathbf{Z}. \end{cases}$, where $C(A)$ denotes the column of $\mathbf{A}$. Compute the K vector z as a solution of the LDP $\mathbf{A}_\psi \approx e$. Note that only the components $z_j, j \in \psi$, are determined.
7: Define $z_j = 0$ for $j \in \mathbf{Z}$. If $z_j > 0$ for all $j \in \psi$, set $\mathbf{x} = \mathbf{z}$ and go to step 2. 8: Find an index $q \in \psi$ such that $x_q/(x_q - z_q) = \min\{x_j/(x_j - z_j) : z_j \leq 0, j \in \psi\}$.
9: Set $\alpha = x_q/(x_q - z_q)$.
10: Set $\mathbf{x} = \mathbf{x} + \alpha(\mathbf{z} - \mathbf{x})$.
11: Move from set ψ to set $\mathbf{Z}$ all indices $j \in \psi$ for which $x_j = 0$. Go to Step 6.
12: End

by using estimated gradients of cost and constraint functions. The essence of this optimization method is transforming the optimal problem of observed period T (shown in (8)) into the optimal problem of each time slot (shown in (9)) as proved in [29]. By minimizing the right-hand of (9), the action decision of next time slot $t+1$ will be executed towards the goal of minimizing the gap between the optimal action and the actual action for the optimal problem with guaranteed constraint in (8). The guarantee of long-term constraint of (8) is then embodied in the online Lagrangian function (shown in (10)) of the right-hand of (9), which makes accumulated violation of constraints be less than QoS requirement during period T. Thus, here the long-term constraint is satisfied for each element $\sum_{t=1}^{T} \mathbf{g}_t(\mathbf{x}_t) \leq 0$ instead of constraint in each slot.

Since the proposed scheme is a central scheme performed at the edge node, which determines the allocation of its resource pool to different applications, there is no data exchange between nodes for the information obtaining. Instead, the edge node only observes the QoS satisfaction degree and cost value without extra communication overhead. In order to perform the resource allocation scheme, the edge node should pay computation cost on it. Next, we analyze the computation cost occurs in the process. For each slot, the edge node firstly determines action (resource allocation) based on iteration value, the computation time of which is $O(K)$. Then, the cost value and constraint value are collected by the edge node. Then, the edge node updates dual variable with computation time $O(K)$. Lastly, the edge node updates iteration values with computation times $O(K) + \Lambda(LDP)$, where $\Lambda(LDP)$ is the computation time of LDP. For the last step, the update of iteration values are calculated based on the computation complexity of LDP, i.e., $\Lambda(LDP)$. From the solution step in table 3 and table 4, we see that the computation complexity of LDP is $O(K)$. Thus, the all computation time complexity is $O(K)$, which has linear complexity.

IV. NUMERICAL RESULTS

To evaluate the proposed scheme, we give some numerical results in this section. We consider 5 applications with different delay constraints which will be processed and relayed by the edge node. The delay constraints vary with different applications [35]. For example, for UMTS, the minimum delay constraint of conventional class traffic is 100 ms, that of stream class traffic can be 300 ms, and that of Interactive class traffic can be longer. For time sensitive application, the delay constraint will be 1 millisecond or more shorter [36]. Here the delay constraints are randomly selected from [0.5, 500]ms unless specified. The time slot (a resource allocation period) unit here is 10 milliseconds, which includes 10 frames (in practice, a frame usually lasts 1 millisecond [37]). The observation time is 1000ms, which is 100 resource allocation periods. If not specified, the arrived data of application i follows Poisson process with $\gamma_i = 1000 * i$. The available resources of edge node are $B = 10^4$ and $P = 8000$, with the transmitted data and processed data per unit resource are 1 and 10, respectively. The observation time $T = 1000$ ms, and the step size $\mu = \alpha = 0.05/T$. In this simulation, the observation time, resources and ability (that of communication and computing) of edge node here are only an example, the unit and amount of which can be adjusted for different scenarios. For example, for shore-based AIS networks, the unit of available bandwidth is kHz, the unit of delay constraint of safety detection can be several minutes while that of the ordinary video can be from several minutes to hours.

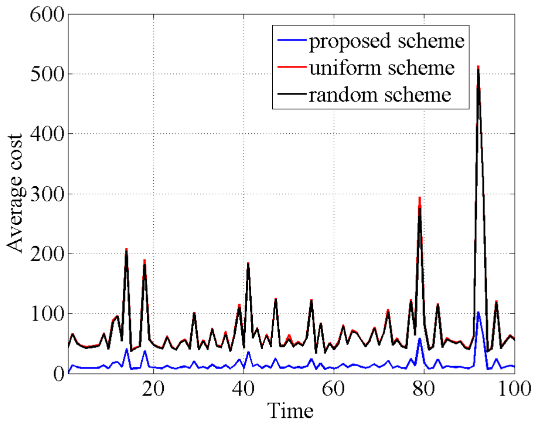

FIGURE 3. Cost comparison based on different resource management schemes.

Fig.3 compares the average cost of each slot for the proposed scheme, uniform scheme where resources are allocated equally to streams of each application, and random scheme, where resources are allocated randomly to streams of each application. In this figure, the unit of time is the number of resource allocation periods (RAPs) and the unit of cost is Hz/bit, respectively. From this figure, we can see that the proposed scheme largely reduces the average cost compared to other two schemes. At some burst points (the amount of data increases in some slots), the propose resource management can also keep lower cost. The cost of random scheme increases significantly at some points since it allocates resources at random, leading to less allocated resources to high-requirement streams. However, the proposed scheme can adjust resources based on their requirements based on the cost of last slot.

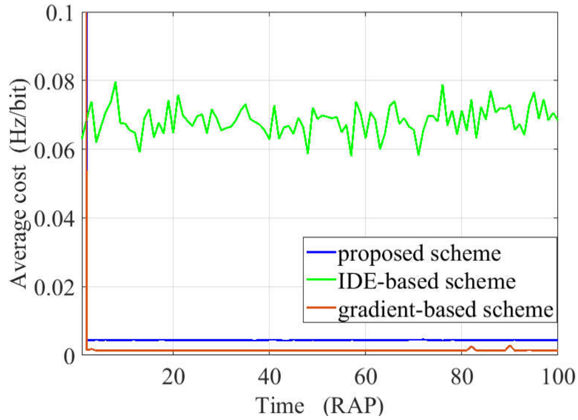

FIGURE 4. Cost comparison among proposed scheme, IDE-based scheme and gradient-based scheme.

In Fig.4, the proposed scheme is compared with other related works. Since there are few works considering the same problem, we select similar works for comparison here. One is a recent work [38], which considers the optimal allocation of computation resources and proposes an improved differential evolution (IDE) method for resource allocation (called IDE-based scheme here). In addition, gradient-based scheme is also compared here as an upper bound [39]. From this figure, we see that the cost of these schemes is low compared those lower bound scheme such as random scheme and uniform scheme. Although the gradient information cannot be obtained, the proposed scheme performs nearly to the gradient-based scheme. Compared to IDE-based scheme, the computation complexity of proposed scheme is lower while the performance is better.

Before introducing the edge computation to traditional communication network, relay nodes usually play the act of communication without computation. To evaluate the effect of computation on the cost, Fig.5 compares the scheme without computation (i.e., no compress scheme) and the proposed scheme as well as uniform scheme. The computation and communication resources are further enhanced to $3 * 10^7$ to evaluate the effect of pre-process and the effect of resource capacity. We can see that appropriately compression is very useful for reducing the cost. Comparing Fig.3 to Fig.5, we see that the proposed scheme is better when there is enough computation ability. However, the proposed scheme can also achieve the low cost although computation resource is reduced.

Fig.6 evaluates the proposed scheme under different parameters. From this figure, we can see that as the increase of applications, the cost is only slightly increased. The step size can affect the performance of scheme, as analyzed in this paper. Large step size will destroy the astringency of the

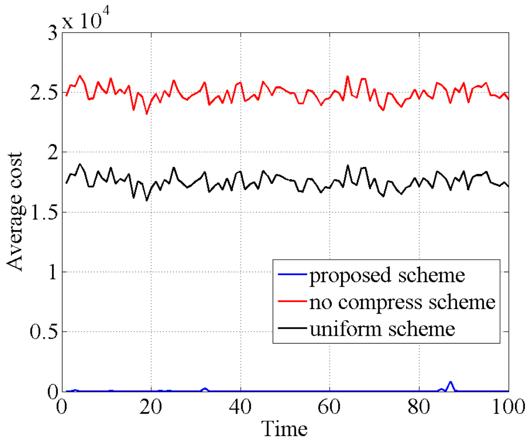

FIGURE 5. Comparison of different resource allocation schemes with no compression scheme.

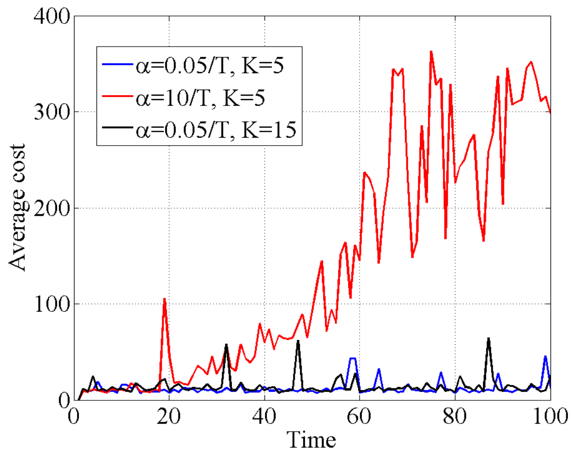

FIGURE 6. Comparison the proposed scheme at some different settings.

scheme. Thereby, appropriate selection of step size also can improve the scheme.

The fit metric give in (22) is used for evaluating the degree of constraint satisfaction. That is, low fit illustrates that constraints of the optimization problem are well satisfied. Thereby, to guarantee the delay-constraints, the fit metric is expected to be zero. From Fig.7, we see that fit values of both the uniform scheme and the proposed scheme are much less than that of random scheme. However, the value is between 30 and 40 instead of near zero. The reason is that the fit metric uses normalization in its definition. Although g^i in the proposed scheme is less than zero (i.e., satisfy the delay constraint), its normalization value is larger than zero. For more clearly show the comparison of delay satisfaction, Fig.8 shows the delay satisfaction probability in each slot. This probability is the division between the number of applications with delay guarantee and the number of all applications. It is shown that the probability of delay guarantee of proposed scheme is in close proximity to 1. Hence, it can provide the low cost with guaranteed QoS requirements.

From Fig.9, we see that the probability of delay guarantee of proposed scheme is similar to that of

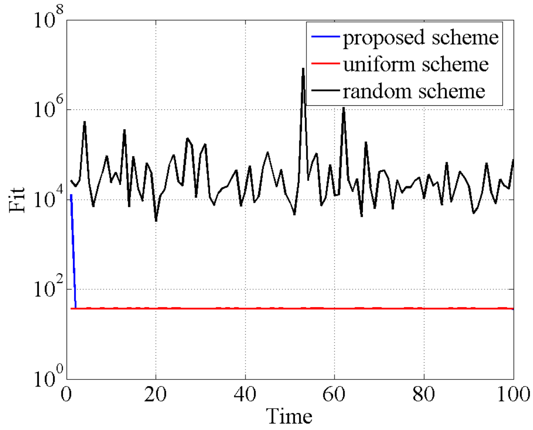

FIGURE 7. Fit comparison based on resource management schemes.

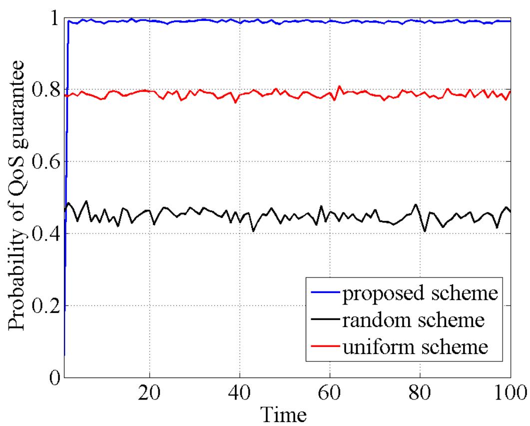

FIGURE 8. Delay constraint satisfaction of different resource management schemes.

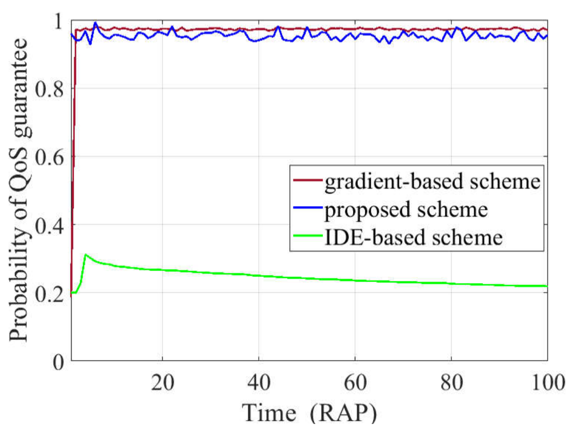

FIGURE 9. Delay constraint satisfaction of proposed scheme, IDE-based scheme and gradient-based scheme.

gradient-based scheme. However, the gradient-based scheme needs gradient information which may be not acquirable sometimes. For the IDE-based scheme, which also needs no gradient information, its delay constraint cannot be well guaranteed since it allocates resources to minimize the cost without the consideration of delay constraint.

V. CONCLUSIONS

Motivated by challenges faced by the feedback of mass of detected data for maritime applications, we study the rapid online resource management approach without the information of cost and constraint functions. It makes the implementation of resource management be easier. This scheme is more appropriate for maritime scenarios since many parameters of networking at the marine environment are difficult to obtain. We explore the following efforts to realize the above merits. Firstly, the information feedback problem is appropriately modeled. Time is slotted for easily tackling decisions at each slot. Streams of applications can be tackled and transmitted without the constraint of time slot. Secondly, the online solution to the optimization problem jointly schedules computation and communication resources while guarantees delay constraints. Last but not the least, with the aid of estimation of gradients, we can allocate resources without explicit functions of both cost and constraint, which is suitable for existing maritime communication networks.

With the consideration on some powerful underwater nodes, the proposed scheme can also be used for underwater nodes by regarding them as edge nodes. With information exchange among edge nodes and underwater powerful nodes, this scheme can be further improved by appropriately allocating the total computation resources and communication resources of these nodes, which will be further considered in the future work.

APPENDIX A

The proof generalizes the result in [29] with full-information gradient feedback. Re-written the minimization problem as follows.

$$\mathbf{x}_{t+1} = \mathtt{arg\ min}_{\mathbf{x}\in\mathbf{X}} \nabla_{\mathbf{x}}^T L_t\left(\mathbf{x}_t, \lambda_t\right)\left(\mathbf{x}-\mathbf{x}_t\right) + \frac{1}{2\alpha}\left\|\mathbf{x}-\mathbf{x}_t\right\|^2. \tag{23}$$

Let $A_t^T = 2\alpha^2\nabla_{\mathbf{x}}^T L_t\left(\mathbf{x}_t, \lambda_t\right)$, the minimization of (23) is equal to minimizing the following equation. That is, $\mathbf{x}_{t+1}$ is the optimal value of (24)

$$\min A_t^T x + \alpha\left\|\mathbf{x}-\mathbf{x}_t\right\|^2, \tag{24}$$

which can also be written as

$$\begin{aligned}
&\mathbf{x}_{t+1}\\
&= \mathtt{arg\ min}_{\mathbf{x}\in\mathbf{X}}\left\{A_t^T\mathbf{x} + \alpha\left\|\mathbf{x}-\mathbf{x}_t\right\|^2\right\}\\
&= \mathtt{arg\ min}_{\mathbf{x}\in\mathbf{X}}\left\{\frac{\left\|A_t\right\|^2}{4\alpha} + A_t^T\left(\mathbf{x}-\mathbf{x}_t\right) + \alpha\left\|\mathbf{x}-\mathbf{x}_t\right\|^2\right\}\\
&= \mathtt{arg\ min}_{\mathbf{x}\in\mathbf{X}}\left\|\sqrt{\alpha}\left(\mathbf{x}-\mathbf{x}_t\right) - \frac{A_t}{2\sqrt{\alpha}}\right\|^2\\
&= \mathtt{arg\ min}_{\mathbf{x}\in\mathbf{X}}\left\|\mathbf{x} - \left(\mathbf{x}_t + \frac{A_t}{2\alpha}\right)\right\|^2\\
&= \mathtt{arg\ min}_{\mathbf{x}\in\mathbf{X}}\left\|\mathbf{x} - \left(\mathbf{x}_t + \alpha\nabla_{\mathbf{x}}^T L_t\left(\mathbf{x}_t, \lambda_t\right)\right)\right\|^2\\
&= \Phi_X\left(\mathbf{x}_t + \alpha\nabla_{\mathbf{x}} L_t\left(\mathbf{x}_t, \lambda_t\right)\right),
\end{aligned} \tag{25}$$

which yields to (12).

APPENDIX B

Here we simply describe the principle to determine β. The fundamental purpose of selecting these parameters is to minimize the communication cost. However, there is always a gap (usually called as regret) between selected actions (decisions) and optimal actions (decisions). Thus, it is expected the smaller regret is, the better is. Let $I_t = \frac{d}{\delta} f_t\left(\mathbf{x}_t + \delta\mathbf{u}_t\right)\mathbf{u}_t + \lambda_t\frac{d}{\delta} g_t\left(\mathbf{x}_t + \delta\mathbf{u}_t\right)\mathbf{u}_t$, and then the iteration can be written as

$$\mathbf{x}_{t+1} = \Psi_{\beta X}\left(\mathbf{x}_t + \alpha I_t\right), \tag{26}$$

where $\alpha = \nu\delta/d$ and ν is a constant. Since

$$\left\|I_t\right\| \leq \frac{dC}{\delta} = G, \tag{27}$$

where C is the bound of $L_t(\mathbf{x}_t, \lambda_t)$. By appropriate choice of ν, we have $\alpha = R/G\sqrt{T}$. Then the regret is bounded by $\frac{RdC\sqrt{T}}{\delta}$. Integrating it to the regret expression, we can get the bound of regret and the corresponding parameter values, i.e., $\delta = \left(\frac{a^2}{bc}\right)^{\frac{1}{3}}$, $\beta = 1 - \left(\frac{ab}{c^2}\right)^{\frac{1}{3}}$, where $a = RdC\sqrt{T}$, $b = 6CT/r$ and $c = 2CT$.

Here we do not give the concrete derivation of regret bound since it is not our focus in this paper, which can be referred to Theorem 1 of [30].

REFERENCES

[1] S. Jiang, "On securing underwater acoustic networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 729–752, 1st Quart., 2019.
[2] O. L. Osen, S. O. Liavåg, L. E. Sætre, O. Morken, and H. Zhang, "Integration and performance study of full functional sea farm inspection platform for aquaculture application," in *Proc. MTS/IEEE SEATTLE OCEANS*, Oct. 2019, pp. 1–9.
[3] M. Carreras, J. D. Hernández, E. Vidal, N. Palomeras, D. Ribas, and P. Ridao, "Sparus II AUV—A hovering vehicle for seabed inspection," *IEEE J. Ocean. Eng.*, vol. 43, no. 2, pp. 344–355, Apr. 2018.
[4] T. Yang, Y. Guo, Y. Zhou, and S. Wei, "Joint communication and control for small underactuated USV based on mobile computing technology," *IEEE Access*, vol. 7, pp. 160610–160622, 2019.
[5] A. Signori, F. Campagnaro, D. Zordan, F. Favaro, and M. Zorzi, "Underwater acoustic sensors data collection in the robotic vessels as-a-service project," in *Proc. OCEANS*, Jun. 2019, pp. 1–9.
[6] Y. Xu, S. Jiang, and F. Liu, "A LTE-based communication architecture for coastal networks," in *Proc. 11th ACM Int. Conf. Underwater Netw. Syst. (WUWNet)*, 2016, pp. 1–2.
[7] S. Jiang, "Fostering marine Internet with advanced maritime radio system using spectrums of cellular networks," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, Dec. 2016, pp. 1–6.
[8] Y. Xu, "Quality of service provisions for maritime communications based on cellular networks," *IEEE Access*, vol. 5, pp. 23881–23890, 2017.
[9] L. Liu, V. Garcia, L. Tian, Z. Pan, and J. Shi, "Joint clustering and inter-cell resource allocation for CoMP in ultra dense cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 2560–2564.
[10] Y. Xu, D. Li, and Y. Liu, "Resource management for interference mitigation in device-to-device communication," *IET Commun.*, vol. 9, no. 9, pp. 1199–1207, Jun. 2015.
[11] Y. Zhao, H. Xia, Z. Zeng, and S. Wu, "Joint clustering-based resource allocation and power control in dense small cell networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Nov. 2015, pp. 1–5.
[12] N. Trabelsi, C. S. Chen, R. El Azouzi, L. Roullet, and E. Altman, "User association and resource allocation optimization in LTE cellular networks," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 2, pp. 429–440, Jun. 2017.
[13] M. Wang, H. Tian, and G. Nie, "Energy efficient power and subchannel allocation in dense OFDMA small cell networks," in *Proc. IEEE 80th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2014, pp. 1–5.

[14] Z. Xiao, J. Yu, T. Li, Z. Xiang, D. Wang, and W. Chen, "Resource allocation via hierarchical clustering in dense small cell networks: A correlated equilibrium approach," in *Proc. IEEE 27th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2016, pp. 1–5.

[15] A. Bousia, E. Kartsakli, A. Antonopoulos, L. Alonso, and C. Verikoukis, "Multiobjective auction-based switching-off scheme in heterogeneous networks: To bid or not to bid?" *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 9168–9180, Nov. 2016.

[16] M. Xie, T.-M. Lok, and Q. Yang, "User association and scheduling based on auction in multi-cell MU-MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4150–4162, Jun. 2018.

[17] Y. He, Z. Zhang, and Y. Zhang, "A big data deep reinforcement learning approach to next generation green wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[18] H. Lalin, I. W. Mustika, A. Bejo, and S. Alam, "Optimal resource allocation scheme in femtocell networks using bacterial foraging optimization," in *Proc. Int. Comput. Sci. Eng. Conf. (ICSEC)*, Dec. 2016, pp. 1–5.

[19] N. Aihara, K. Adachi, O. Takyu, M. Ohta, and T. Fujii, "Q-learning aided resource allocation and environment recognition in LoRaWAN with CSMA/CA," *IEEE Access*, vol. 7, pp. 152126–152137, 2019.

[20] C. Qiu, H. Yao, F. R. Yu, F. Xu, and C. Zhao, "Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5871–5883, Jun. 2019.

[21] M. A. Qureshi and C. Tekin, "Fast learning for dynamic resource allocation in AI-enabled radio networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 1, pp. 95–110, Mar. 2020.

[22] F. Campagnaro, P. Casari, M. Zorzi, and R. Diamant, "Optimal transmission scheduling in small multimodal underwater networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 368–371, Apr. 2019.

[23] R. Diamant, P. Casari, F. Campagnaro, and M. Zorzi, "Leveraging the near–far effect for improved spatial-reuse scheduling in underwater acoustic networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1480–1493, Mar. 2017.

[24] L. Li, W. Liu, L.-E. Gao, Y. Zhang, Z. Li, and Z. Zhang, "A novel network scheduling approach based on genetic algorithm for autonomous underwater vehicle control," *IEEE Access*, vol. 7, pp. 111053–111064, 2019.

[25] Z. Liao, D. Li, and J. Chen, "Joint bandwidth optimization and media access control for multihop underwater acoustic sensor networks," *IEEE Sensors J.*, vol. 15, no. 8, pp. 4292–4304, Aug. 2015.

[26] M. Elamassie, M. Karbalayghareh, F. Miramirkhani, M. Uysal, M. Abdallah, and K. Qaraqe, "Resource allocation for downlink OFDMA in underwater visible light communications," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, Jun. 2019, pp. 1–6.

[27] J. Wang, H. Zhou, Y. Li, Q. Sun, Y. Wu, S. Jin, T. Q. S. Quek, and C. Xu, "Wireless channel models for maritime communications," *IEEE Access*, vol. 6, pp. 68070–68088, 2018.

[28] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[29] M. J. Neely and H. Yu, "Online convex optimization with time-varying constraints," 2017, *arXiv:1702.04783*. [Online]. Available: http://arxiv.org/abs/1702.04783

[30] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," 2004, *arXiv:cs/0408007*. [Online]. Available: https://arxiv.org/abs/cs/0408007

[31] A. Agarwal, O. Dekel, and L. Xiao, "Optimal algorithms for online convex optimization with multi-point bandit feedback," in *Proc. Colt Conf. Learn. Theory*, 2009, pp. 28–40.

[32] C. Lawson and R. Hanson, *Solving Least Squares Problems*. New York, NY, USA: Prentice-Hall, 1974.

[33] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: Online convex optimization with long term constraints," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 2503–2528, 2012.

[34] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 928–936.

[35] *Quality of Service (QoS) Concept and Architecture (Release 15)*, document TS 23.107, 3GPP, 2018.

[36] *IEEE Standard for a Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks*, IEEE Standard 1722-2016, 2016.

[37] *Physical Channels and Modulation (Release 15)*, document TS 38.211, 3GPP, 2020.

[38] P. Zhang, M. Zhou, and X. Wang, "An intelligent optimization method for optimal virtual machine allocation in cloud data centers," *IEEE Trans. Autom. Sci. Eng.*, early access, Mar. 9, 2020, doi: 10.1109/TASE.2020.2975225.

[39] Q. Fan and N. Ansari, "On cost aware cloudlet placement for mobile edge computing," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 4, pp. 926–937, Jul. 2019.

YANLI XU (Member, IEEE) received the Ph.D. degree from the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China, in 2012. From 2012 to 2015, she worked as a Research Scientist with the Bell Labs, Alcatel-Lucent, China, with a focus on the standardization of 5G cellular networks. She is currently an Associate Professor with Shanghai Maritime University. Her research interests are in 6G, maritime communications, and the IoT, with a current focus on edge computation. She is also an Associate Editor of IEEE Access, and serves as a TPC Member or a Reviewer for many conferences and journals, such as the IEEE Journal on Selected Areas in Communications (JSAC), TCOM, TWC, TSP, IET Com, and Globecom.