Massive Maritime Path Planning: A Contextual Online Learning Approach

Pan Zhou, *Member, IEEE*, Weiguang Zhao, Jianghui Li, Ang Li, *Student Member, IEEE*, Wei Du, and Shiping Wen

***Abstract*—The ocean has been investigated for centuries across the world, and planning the travel path for vessels in the ocean has become a hot topic in recent decades as the increasing development of worldwide business trading. Planning such suitable paths is often based on big data processing in cybernetics, while not many investigations have been done. We attempt to find the optimal path for vessels in the ocean by proposing an online learning dispatch approach on studying the mission–executing–feedback (MEF) model. The proposed approach explores the ocean subdomain (OS) to achieve the largest average traveling feedback for different vessels. It balances the ocean path by a deep and wide search, and considers adaptation for these vessels. Further, we propose a contextual multiarmed bandit-based algorithm, which provides accurate exploration results with sublinear regret and significantly improves the learning speed. The experimental results show that the proposed MEF approach possesses 90% accuracy gain over random exploration and achieves about 25% accuracy improvement over other contextual bandit models on supporting big data online learning pre-eminently.**

***Index Terms*—Big data, contextual multiarm bandit (CMAB), maritime path planning, online learning.**

I. Introduction

IN RECENT years, the ocean has been increasingly explored and investigated both on the sea surface and in the water column [1]–[9]. Such exploration involves path planning for vessels, e.g., in cybernetics [10]–[15]. Meanwhile, automatic identification system (AIS) data equipped in vessels are growing rapidly [16], e.g., over two million records have been produced in a single month within only 300 km in the Yangtze River [17].

Recently, the mixed human–machine systems and mastering information dominance for effective context-driven operations have drawn interest from various countries, because of their tempting effectiveness on organizing vessel paths in a channel [18]. However, constructing such a system to solve the path planning problem is still not an easy task, due to the complex and interaction influences. The main goal of this article is to choose a better ocean area for different kinds of vessels between two sites. Some effective solutions have been presented in [19]–[22], while the character of vessels (contexts) has been neglected. In order to successfully incorporate context into the decision-making process, we implement a contextual algorithm to solve the path problem.

As the ocean traffic continuously creates massive data, big data processing approaches are then needed to be developed [23]–[25]. For this kind of processing, researchers have considered the data transmission and feedback [26]–[28]. Bidkar *et al.* [29] proposed a new control plan based on two schemes for segment routing (SR) implementation, which considers a tradeoff between the routing distance of the path and the routing-table size of the nodes. These proposed methods can be used for virtual routing, while entity routing is more complex in terms of the energy and environment limitation. To tackle the energy-aware routing problem, Lent [19] proposed a cognitive packet network to choose the best link. Then, Pavone *et al.* [21] presented a dynamic version with a known distribution. To maximize the steady service of the fraction of demands, Bopardikar *et al.* [20] presented an appropriate acyclic graph structure which computes long path routing efficiently.

After the automatic identification system (AIS) became a standard for ship-to-ship, ship-to-shore, and shore-to-ship communication of information, Zhu [30] used association rules to discover the positions of the vessels. Ahmad *et al.* [31] developed a bridging distribution method based on the Bayesian inference and Kalman filtering, and Han *et al.* [32] used dynamic list planning which interacts with humans to dynamically allocate hierarchically organized assets. However, they mainly concentrate on normal trajectories and patterns.

To efficiently plan paths and avoid obstacles, Qingji *et al.* [33] used the A-star algorithm and Shan *et al.* [34] used the rapidly exploring random tree (RRT) to choose the path. As artificial intelligence (AI) progresses, Lin and Goodrich [35] proposed a heuristic algorithm-based

Manuscript received April 12, 2019; revised September 1, 2019; accepted December 9, 2019. Date of publication February 26, 2020; date of current version December 22, 2021. This work was supported in part by NSFC under Grant 61972448, and in part by the European Union's Horizon 2020 Research and Innovation Programme (STEMM-CCS) under Grant 654462. This article was recommended by Associate Editor S. X. X. Yang. *(Corresponding author: Jianghui Li.)*

Pan Zhou is with the School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: panzhou@hust.edu.cn).

Weiguang Zhao is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: u201613519@hust.edu.cn).

Jianghui Li is with the Institute of Sound and Vibration Research, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: j.li@soton.ac.uk).

Ang Li is with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA (e-mail: al380@duke.edu).

Wei Du is with the Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701 USA (e-mail: wd005@email.uark.edu).

Shiping Wen is with the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: shiping.wen@uts.edu.au).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2019.2959543.

Digital Object Identifier 10.1109/TCYB.2019.2959543

hierarchical system for unmanned aerial vehicles (UAVs) and Eichhorn [36] presented a solution using the fast graph algorithm for autonomous underwater vehicles (AUVs). For complex underwater networking problem, Basagni *et al.* [37] presented a reinforcement-learning (RL) approach which selects the best forwarding relay known as MARLIN. All of this research has made great progress, while they only consider the intractable problems, and big data approaches have not been considered.

To conduct the ocean path planning, the multiarm bandit (MAB) algorithm has been proposed [38]–[40], which is one of the simplest and best online learning algorithms. Compared to the RL algorithm, MAB guarantees the optimal path in any finite time slots [41], which is suitable for the mission-critical maritime path planning. Awerbuch and Kleinberg [42] used the MAB to solve the routing problem, but it ignores the shortest path, that is, the best one is not chosen. To tackle the big data problem, Song *et al.* [43] proposed an adaptive clustering recommendation algorithm (ACR) to construct an infinite binary tree from *bottom-to-up*, which can only support fixed-size datasets, and Gift [38] used the MAB to develop an interdiction model in the ocean counter-smuggling domain. However, these MAB algorithms are not context awareness, which means that only explored targets are considered and internal attribution of the subjects is ignored.

In this article, we present a contextual MAB algorithm (CMAB) using the context of vessels. The vessels are classified in speeds, from which we recommend different path types. The context accelerates the exploration speed by recommending the suitable path for vessels. Further, we create a predecision method to decide which macroscopic ocean that we will exploit. In addition, when vessels are traveling on the ocean, they should contact with the data center which analyzes the feedback to submit traveling condition. Therefore, we use a sensor network to transmit information timely. Based on the MAB algorithm, we propose a tree-based mission–executing–feedback (MEF) system, in which the Mission (M) and Executing (E) parts are used to balance the exploring and exploiting, and Feedback (F) part is used to transfer the data and make resource allocation. The contribution of the proposed approach is concluded as follows.

1) The proposed MEF can handle big data exploration with an expanding top-to-down tree structure and find optimal paths with a minimum 20% faster than other compared algorithms.
2) We use the vessel contexts to accurately recommend ocean subdomains (OS) (contextual recommendation).
3) The predecision algorithm forms a closed-loop to guarantee the stable running.

This article is organized as follows. In Section II, we formulate the model and show the workflow. In Section III, we define the performance metric and notations. In Section IV, the proposed algorithm is described. Numerical results are shown in Section V. Section VI concludes this article.

II. System Model

This section describes the system model that we are using to propose the algorithm. When planning the path in the ocean, we consider the vessels, the ocean, and the decision. The model mainly contains three components: 1) the data center to make predecision; 2) the vessel's navigation (vessel context), e.g., in the left part of Fig. 1; and 3) the ocean (OS), e.g., in the right part of Fig. 1.

A. Ocean SubDomain

Different ocean areas have different feedback and a certain ocean's feedback varies according to a constant average feedback μ. Adjacent subdomains' μ of the ocean is continuous. Thus, we use the Lipschitz condition (Assumption 2 in Section III) to limit the average feedback change and to search the subdomain with the peak value μ_i called μ^*.

To support the big data analysis, we build an infinite binary tree to achieve accurate and fast exploration for the ocean area. Let O represent the entire area of the ocean, and every OS is in O, that is, $OS \in O$, where O represents the vector of an OS, which is drawn from a d_O-dimensional space O, describing one of its subdomain properties. The ancestor node of infinite *OS-cluster tree* structured for the OS space O is denoted by T^O. As shown in Fig. 1, the right part of it is our binary tree, and the ith node at depth h from the root node in T^O is denoted by (h, i). The index i of all nodes at depth h is constrained by $1 \leq i \leq 2^h$.

Each node in T^O represents an OS cluster in O. For example, the root node (0, 1) covers the entire OS space O, while its left child (1, 1) and right child (1, 2) cover two subspaces measured by dissimilarity (definition 1) d_O of O. We define the depth of tree T^O as $H = \max_{(h,i)\in T^O} h$. The region associated with cluster (h, i), which is denoted by $\mathcal{R}_{h,i}$, satisfied the constraints: $\forall h \geq 0, i \geq 1, i' \leq 2^h, \mathcal{R}_{h,i} \cap \mathcal{R}_{h,j} = \emptyset$, $\mathcal{R}_{h,i} = \mathcal{R}_{h+1,2i-1} \cup \mathcal{R}_{h+1,2i}$. One significant benefit of the OS-cluster tree structure is that it recommends OSs on a cluster level instead of a single OS level, which reduces the time complexity from $O(n)$ to $O(\log n)$.

To obtain more accurate feedback and more confident experience, we set a threshold τ for each subdomain which prevents the algorithm changing the node that we are exploring. Moreover, we propose an accelerated algorithm such that the accurate context can be partitioned, and the τ can be decreased for expanding the binary tree, as shown in (5).

B. Vessel Context

Vessels possess their own identity number W_{ID} and have their attribution called context. Denote W as the set of vessels in the target space with cardinality $|W|$. For each $w \in W$, let WC_v denote the contexts of the vessel. The contexts involve the vessel's speed, fire equipment, and ship board personnel. The contexts influence dependency among the vessels due to the similar properties. For example, if a vessel possesses a speed of 18 knots per hour, it has a good performance for a transmission task in an OS, while another vessel possesses a speed of 16 knots per hour for the same task in a similar subdomain. The feedback is likely to be slightly lower. Thus, we aggregate all vessels' contexts to feature as a context vector C, e.g., $C = \{WE_1, WC_1, \ldots, WE_w, WC_w, \ldots, WE_{|W|}, WC_{|W|}\}$, where WE means the math expectation feedback of vessel W. We treat it a particular dimension of the vessel context because it will change in our exploring process and finally tend to the optimal feedback.

The vessel context vector c is drawn from a d_C-dimensional context space C, where each dimension describes one feature of vessels. The vessels with different features should have different missions, thus we build a tree of oceans in each context partition and for each partition we find the adjacent data to build the tree. The data of vessels' contexts partition is demonstrated in the left part of Fig. 1. When the number of vessels in context is large enough, we divide the context space (one big square to four small squares), and reconstruct a new binary OS

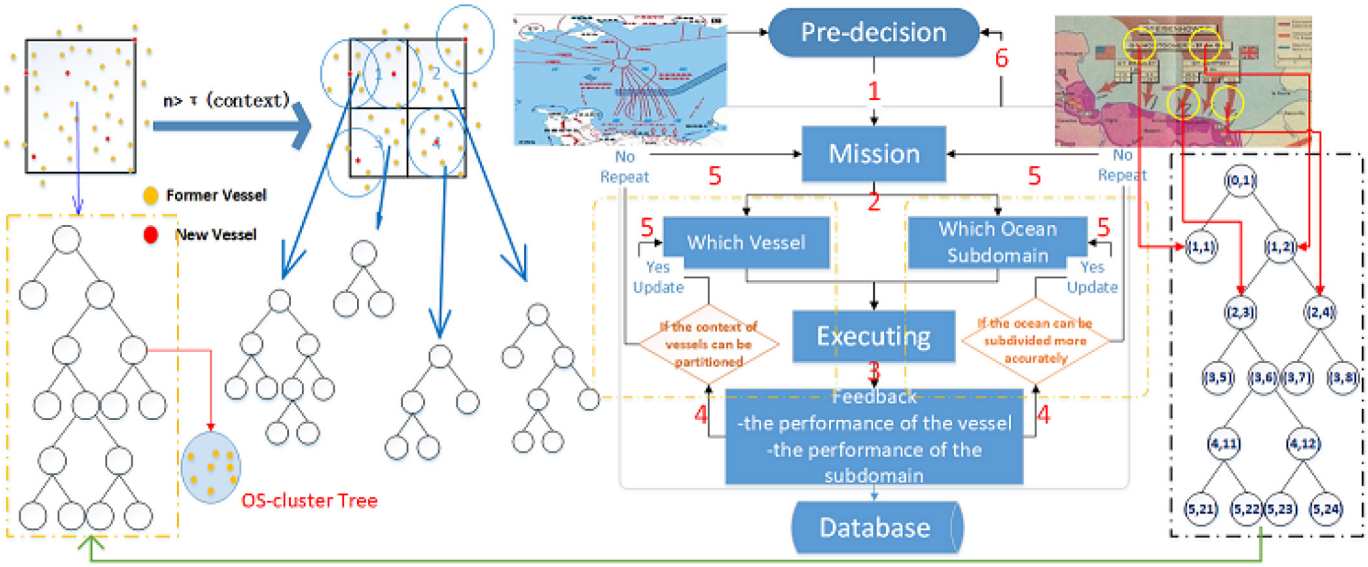


Fig. 1. MEF model workflow with the approach of tree-based ocean exploration and context partition. Maps by Google map.

tree for each context. When we calculate the property for each small square, we not only calculate the points in the square but also the *friends* of the points which are sometimes out of the square, e.g., we draw a circle to calculate their adjacent points.

C. Predecision Model

The predecision model consists of two layers, that is, the decision layer and the deployment layer. In the decision layer, we analyze the data using LP and deliver it to the commander. Then the commander makes a subjective decision for the exploration in certain ocean areas. In the deployment layer, we allocate the vessels and wait for the next interval's decision. The rough process of the model and the decision process are then demonstrated.

We formulate a scheme to predict the feedback for each channel by LP from the data center. Let $\mathrm{FB}(t)$ be the final feedback for our military operation of a certain macroscopic ocean area. Then to find the influence factors of $\mathrm{FB}(t)$, we constrain the value range of influence coefficients. We denote the d-dimensional influence coefficients by $\mathrm{IF} = \{if_1, if_2, \ldots, if_d\}$. For the constraint part, $\mathrm{OS}(IF, t)$ ($\mathrm{ES}(IF, t)$) is the function of our strength (enemy strength) with factors and time. $\mathrm{Dos}(IF, t)$ ($\mathrm{Des}(IF, t)$, $\mathrm{Denv}(IF, t)$) is the function of the coasts distance. In the channel, the force projection limitation is a crucial factor and we set $\mathrm{fpl_{os}}(t)$ ($\mathrm{fpl_{es}}(t)$) to be our (enemy) force projection limitation function. $\mathrm{ENS}(IF, t)$ is the disturbing function of military strength. We suppose that the military strength in the channel cannot exceed the force limitation and environment cannot change discontinuously as

$$\begin{cases} \mathrm{OS}(IF, t) + \mathrm{Dos}(IF, t) \le \mathrm{fpl_{os}}(t) \\ \mathrm{ES}(IF, t) + \mathrm{Des}(IF, t) \le \mathrm{fpl_{es}}(t) \\ \mathrm{EN}(IF, t) + \mathrm{Denv}(IF, t) \le \max\left(\mathrm{average}(t + \Delta t)\right). \end{cases}$$

We suppose the military strength both ours and enemies in the channel cannot exceed the force limitation, thus we can obtain the first and second formulas. The environment cannot change discontinuously, thus we can obtain the third formula like the weak Lipschitz condition. The three formulas provide us a limitation.

D. Mission–Executing–Feedback Workflow

Fig. 1 shows the workflow of the MEF model as in the center part, in which six main processes are labeled from step 1 to step 6. The workflow from the top to the bottom (steps 2–5) is the exploration part. The negative feedback part (steps 1 and 6) along with the aforementioned part form a closed loop system, which adjusts the ocean selection and allocates the vessels. When there are massive feedbacks from a certain ocean area, we allocate more vessels to explore it; and *vice versa*.

Specifically, the workflow contains six steps as follows.

1) The data center makes predecision, including analyzing the historical data and updating the vessel context C and OS $\mathcal{O}$ to decide the explored ocean.
2) Find the coming vessel with context c belongs and appoint an ocean $\mathcal{R}_{h,i}$ for the vessel. Then the vessel executes the mission. We call this process the Mission (M) allocation.
3) If there is no emergency in the OS, the vessels cannot change the mission and we can obtain a reasonable feedback transforming to the radar station by space communication, known as Executing (E).
4) The sending feedback is analyzed in the data center, judging if the vessel context can be partitioned/divided and if the OS can be subdivided.
5) If the number of vessels is larger than a threshold τ_c or the OS has been explored enough times τ_h, we divide the context space (e.g., vessel speed $\in$ [10, 20] knots is divided into [10, 15] knots and [15, 20] knots) or update the binary tree and send new missions to the data center. If not, we maintain the origin condition and repeat the same mission, known as Feedback (F).
6) For every once that we check if the mission is still feasible. We construct an exploiting closed loop for the ocean exploring. We review the explored ocean and redistribute the mission and update the vessel types. If the ocean state is not good enough, we decrease the number of vessels allocated to the ocean; and *vice versa*.

In summary, by using the MEF model, we are able to plan the optimal path in a microscopical ocean, and work out in the macro ocean selection.

III. Performance Metric

In this section, we introduce performance matrices, involving dissimilarity and diameter, regret analysis, and tree-based exploration structure and Lipschitz condition. Regret analysis can provide us a criterion for comparison of different algorithms. First, it sets a genuine optimal path and compares the total loss between our path and the optimal path during our exploration. The loss is called the regret analysis. Then we calculate the largest value of the regret called regret bound. We compare the worst case of different algorithms to find the optimal algorithms.

A. Dissimilarity and Diameter

We first define the dissimilarity of user context space. The vessel context space C is assumed to own a dissimilarity function d_C such that $d_C(c, c) = 0$ and $d_C(c, c') \geq 0$ for any $c, c' \in C^2$ which $C = [0, 1]^{d_C}$. In the Euclidean space $\mathcal{C}$, the functions d_C can be the norm of Euclidean. For any measurable space [e.g., $(\mathcal{O}, l)$ and $(\mathcal{C}, d_C)$], once given the element set $\mathcal{O}$ and the distance function f, we let the maximum dissimilarity of two elements in a subset be $A \subseteq \mathcal{O}$ as $\text{diam}(A) \coloneqq \sup_{x,y\in A} f(x, y)$, and $\mathcal{CIR}(o, \rho) \coloneqq \{o' \in \mathcal{O} : f(o, o') \leq \rho\}$ denotes the subset containing all similar vessel elements whose distance to $o \in \mathcal{O}$ is no longer than a constant $\rho > 0$. For vessel context space, we call $\mathcal{CIR}(c(t), \rho)$ as a d_C circle with radius $\rho > 0$ and center $c(t)$ to identify useful historical results of current dispatch. Similarly, the OS context space O is assumed to be equipped with a dissimilarity function D_o. In the right part of Fig. 1, we normalize the $d_C = 2$.

B. Regret Analysis

We use the notation $fb(o, c) \in [0, 1]$ to relate feedback fb to the ocean o browsed and vessels' context c (particularly, $fb(o, c) = 0$ means the vessel be devastated in this subdomain). The basic idea is that the stochasticity of feedback mainly comes from different vessels' diverse equipment conditions, and a certain OS with corresponding context will lead to a specific distribution. Thus, we assume the distributions of fb are independent and identically distributed (i.i.d.), which only depend on o and c. We use $\mu_{o,c} \coloneqq \mathbb{E}\{fb(o, c)\}$ to represent the expectation feedback, which can be obtained from the OS o with vessel context c.

Here, we use a fixed option to bound the worst case in the theoretical analysis. However, in practice, the algorithm seldom achieves that bound. The choice of $o_{h,i}$ changes at every round in experiments and practical applications. Assuming the maximizer $o_t^* = \arg\max_o \mu_{o,c(t)}$ exists, we denote the corresponding maximum $\mu_{o_t^*,c(t)}$ by μ_t^*. To measure the accuracy of the MEF model at step t, we denote the regret at t by $\Delta_t = \mu_t^* - fb_t$. The over n steps expectation regret RE_n is defined as

$$\text{RE}_n = \mathbb{E}\sum_{t=1}^{n} \Delta_t = \sum_{t=1}^{n}\left(\mu_t^* - \mathbb{E}r_t\right) = \sum_{t=1}^{n}\left(\mu_t^* - \mu_{h_t,i_t}(t)\right) \quad (1)$$

where $\mu_{h_t,i_t}(t)$ is a short notation for $\mu_{o_{h_t,i_t},c(t)}$ which means that the mean value of ocean experience feedback at time t.

A tighter upper bound for a near-optimal ocean subset is required in the theoretical analysis. Let $\epsilon > 0$, the subset of ϵ-optimal oceans is defined as $\mathcal{O}_\epsilon = \{o \in \mathcal{O} : \mu_t^* - \mu_{o,c(t)} \leq \epsilon\}$. In the next section, we attempt to prove that our approaches output near-optimal OS with high probability. Then, we can characterize the *scale* of the problem, which means how *large* the set of ϵ-optimal subdomains in $\mathcal{O}$ is, by the concept of *packing number*. For $\epsilon' < \epsilon$, there exists a *packing constant* C_O such that $\mathcal{N}(\mathcal{O}_\epsilon, l, \epsilon') \leq C_O(\epsilon')^{-d_O}$, where $\mathcal{N}(\mathcal{O}_\epsilon, l, \epsilon')$ is the maximum number of circle areas with the radius ϵ' that are disjoint with each other in the region $\mathcal{O}_\epsilon$ of the distance measure l. Similarly, we denote the minimum number of circle areas by $\mathcal{N}_\rho(\mathcal{C})$, with the radius $\rho > 0$ that covers the context space $\mathcal{C}$ of the distance measure d_C. By analogy we have $\mathcal{N}_\rho(\mathcal{C}) \leq C_C\rho^{-d_C}$, where C_C is the *covering constant* of $\mathcal{C}$.

TABLE I
Algorithm Notation

t	number of explorations
Δt	interval to inspect the whole mission. Redo Pre-decision
(h_t, i_t)	cluster selected by MEF at time t
o_t	exploration ocean result at time t
$c(t)$	context of the vessel at time t
fb_t	feedback received at time t
ν_1	maximum distance in the ocean space
ν_2	limited distance in the ocean space
$\mathcal{R}_{h,i}$	corresponding ocean sub-domain of node (h, i)
$w(t)$	dispatched vessel at time t
$\mathcal{T}_t$	ocean exploration tree already built at time t
$H(t)$	depth of the binary tree $\mathcal{T}_t$
R_t	ergodic path from the root node to (h_t, i_t)
$N_S(c(t))$	number of vessel contexts in the square partitioned area belonging to $c(t)$
$\mathcal{CIR}(c(t), \rho_t^c)$	a d_C-circle with radius ρ_t^c and center $c(t)$
$\mathcal{T}(\mathcal{CIR}(c(t), \rho_t^c))$	set of context arriving times in $\mathcal{CIR}(c(t), \rho_t^c)$
$T_{h,i}(t)$	number of times the cluster (h, i) has been explored in $\mathcal{CIR}(c(t), \rho_t^c)$
$\mathbb{I}$	indicator function
$\widehat{\mu}_{h,i}(t)$	empirical feedback of the cluster (h, i) at time t
$U_{h,i}(t)$	estimated feedback upper-bound of node (h, i) at time t
$\bar{\delta}(t)$	$min\{c_1\delta/t, 1\}$ ($c_1, \delta > 0$ are constants)
t^+	$2^{\lfloor log(t)\rfloor+1}$
$B_{h,i}(t)$	tighter feedback bound of node (h, i) at time t
$\tau_h(t)$	threshold of explored times for each ocean node
$\tau_c(t)$	subdivision threshold for each vessel context

C. Tree-Based Exploration Structure and Lipschitz Condition

Here, we use two assumptions to describe the tree-based exploration structure and the Lipschitz condition.

Assumption 1 (Tree-Based Exploration Structure): There exist $\nu_1 > \nu_2 > 0$ and $0 < \rho < 1$ such that for any node $(h, i), (h, j) \in \mathcal{T}$

$$(a) \quad \nu_2\rho^h \leq \text{diam}(\mathcal{R}_{h,i}) \leq \nu_1\rho^h$$
$$(b) \quad \exists o_{h,i}^o \in \mathcal{R}_{h,i} \text{ s.t. } \mathcal{S}_{h,i} \coloneqq \mathcal{S}\left(o_{h,i}^o, \nu_2\rho^h\right) \subset \mathcal{R}_{h,i}.$$

This assumption is the addicted experimental characteristics of the ocean space. Note that constants ν_1, ν_2, and ρ depend on dissimilarity function D_o and the original data characteristics. Practically, ρ is often around 0.3. Two similar subdomains of ocean o_2 and o_3 in the same node should satisfy $D_o(o_2, o_3) \leq \nu_1\rho^{h_1}$, and $\text{diam}(\mathcal{R}_{h_1,i_1}), \text{diam}(\mathcal{R}_{h_1,i_2}) \geq \nu_2\rho^{h_1}$.

Assumption 2 (Lipschitz Condition): Given two contexts $c, c' \in \mathcal{C}$, for all OSs $o \in \mathcal{O}$ and $t > 0$, we have $|\mu_{o,c} - \mu_{o,c'}| \leq L_C d_C(c, c')$ and $\mu_t^* - \mu_{o,c(t)} \leq D_o(o_t^*, o)$, where L_C is the Lipschitz constant in the context space.

This assumption for contexts L_C is only required for theoretical analysis. Meanwhile, the Lipschitz condition for the ocean space only needs the expectation feedback function to be Lipschitz with the maximum $|\mu_{o,c} - \mu_{o,c'}| \leq \max D_o(o_t^*, o), \mu_c^* - \mu_{o,c'}$, which is weaker than the original one and is so-called *local smoothness* assumption.

Algorithm 1: MEF Model

Input: IF, Δt, $\nu 1 > \nu 2 > 0$, $\rho \in (0,1)$, d_O, d_C, confidence $\delta \in (0,1)$, tree-based exploration structure $(\mathcal{R}_{h,i})_{h\geq 0, 1\leq i\leq 2^h}$.

1 **Initialize** : $t = 1$, $\mathcal{T}_t = \{(0,1),(1,1),(1,2)\}, H(t) = 1, U_{1,1}(t) = U_{1,2}(t) = infinity$;
2 **Auxiliary procedure** : Predecision, OptTree, Optpath;
3 **for** *t = 1, 2, 3, do* **do**
4 **if** $t = \Delta t, 2\Delta t, 3\Delta t, \ldots$ **then**
5 Predecision;
6 A vessel $w(t_n)$ with context $c(t_n)$ at time slot t_n is in order to be dispatchted to the OS. Find the context subspace C_n which $c(t_n)$ belongs to and search for the tree $\mathcal{T}_t$;
7 $\mathcal{T}_t \leftarrow OptTree(c(t_n))$;
8 **for** *all calculated OS* $(h,i) \in \mathcal{T}_t$ *backward from the leaf* $(H(t),i)_{1\leq i\leq 2^{H(t)}}$ **do**
9 Update estimated node's upper-bound $U_{h,i}(t)$ in Eq. (3);
10 Update the tighter exploration feedback bound $B_{h,i}(t)$ in Eq. (4);
11 $(h_t, i_t), P_t \leftarrow Optpath(\mathcal{T}_t)$;
12 The vessel explore the ocean corresponding to o_{h_t,i_t} and gives its feedback fb_t;
13 **if** $T_{h_t,i_t}(t) \geq \tau_{h_t}(t) and (h_t,i_t) \in leaf(\mathcal{T}_t)$ **then**
14 $\mathcal{T}_t \leftarrow \mathcal{T}_t \bigcup \{(h_t+1, 2i_t-1), (h_t+1, 2i_t)\}$;
15 $U_{h_t+1,2i_t-1}(t) = U_{h_t+1,2i_t}(t) = infinity$;
16 Enter into the next moment $t = t+1$;

IV. Proposed Algorithm

In this section, we present the proposed MEF algorithm and other auxiliary algorithms. Table I introduces notations for the algorithm analysis, in which we provide the mathematical theorem of the algorithm in the regret analysis.

A. Algorithm Description of MEF Algorithm

Algorithm 1 shows the elaborate MEF algorithm, where there are three main steps: 1) preparation for the entire algorithm (line 5); 2) vessel context analysis and tree option (lines 6 and 7); and 3) OS option and tree exploration (lines 8–16).

1) Preparation: In accordance with the practical situation and experience, we input the adjustable parameters, where IF is the coefficient of the LP for the predecision model; Δt is the interval that we evaluate and redeploy the vessels. For example, we set $\Delta t = 10^3$, when $t = 10^3, 2\times 10^3, 3\times 10^3, \ldots$, run the predecision model, and check the mission loop to redeploy the vessels. δ decides the exploration rate in the ocean, known as the confident coefficient. The higher δ is, the quicker we can explore the ocean. Then we construct the ocean domain tree $\mathcal{T}_t$, initiate the depth of the tree $H(t) = 1$ and their confidence. Load three auxiliary functions (lines 1 and 2), we start the main algorithm. After a certain operation time, we use the predecision function to evaluate the practical reward and redeploy the military strengths (e.g., increase a certain kind of vessels) (lines 3–5).

2) Vessel Context Analysis and Tree Option: At each moment t, model receives a mission inquired by a vessel with its context $c(t)$. To accelerate exploration speed, the MEF utilizes historical exploration feedback that has a similar context with $c(t)$. We assume $d_C = 2$, and use a square to divide the entire context area. When there are enough explored vessel points in the square ($N_S(c(t)) \geq \tau_c$), we subdivide the square. In every subdivision, the length of the small square's side becomes half of the large square and the large square is divided into four (2^{d_C}, and for simplicity we select $d_C = 2$) small squares for more accurate vessel mission allocation. For example, before we dispatch enough vessels to the ocean, we have an initial division method that the vessels are divided into two types: 1) on the sea surface like aircraft and 2) underwater like a submarine. While with more and more vessels executing the mission, the initial division way is not efficient enough. Thus, we subdivide the sea surface type into two parts: 1) relatively large vessels like aircraft and 2) relatively small vessels like destroyers.

It is obvious that there are various ocean adaptations between vessel speed $\in [10, 20]$ knots and $[20, 30]$ knots. Thus, for each small division, we build an ocean exploration tree to accurately explore them. There is also a negative feedback between the vessel points and division threshold. As the kind of vessel grows, we subdivide them for accuracy. In contrast, as the vessels have been divided more accurately, the subdivision threshold becomes larger for reinforce exploration. Thus, we choose

$$\tau_c(t) = \lambda 2^{2d_C k} \ln n \tag{2}$$

where k is the partitioned times, and λ is a control parameter.

3) Ocean Subdomain Option and Tree Exploration: After the dispatched vessel is decided, we select OS to pass based on (2). This process involves three parts: 1) renewal; 2) searching output OS; and 3) expanding.

a) Renewal: The MEF renews the estimated feedback of the entire tree-based exploration structure on the square context partition. We define a practical notation, the number of times that the OS (h,i) has been explored in the square context $T_{h,i}(t) = \sum_{\tau\in\mathcal{T}(\mathcal{CIR}(c(t),\rho_t^c))} \mathbb{I}\{h_\tau = h, i_\tau = i\}$, where $\mathbb{I}$ is the indicator function. We can see that the context partition could be improved, because a vessel with the thickness of the deck 11 mm is in a tree between [10, 20] mm. Because 11 mm is more close to 8 mm than 17 mm, we set $\mathbb{I}$ function to calculate the feedback of vessel points both in the square and in a range out of it. We draw a circle with radius ρ_t^c to find their neighbor points and calculate the average OS selection times. If the OS divisions are different, we include them to the exploration tree's partition node. Then the empirical exploration feedback $\widehat{\mu}_{h,i}(t)$ of $o_{h,i}$ is computed as $\widehat{\mu}_{h,i}(t) = (1/[T_{h,i}(t)]) \sum_{\tau\in\mathcal{T}(\mathcal{CIR}(c(t),\rho_t^c))} fb_\tau$, where $\mathcal{T}(\mathcal{CIR}(c(t),\rho_t^c))$ means the points in the square partition and circle neighbor finder. Then we bring in an upper bound $U_{h,i}(t)$

$$U_{h_t,i_t}(t) = \widehat{\mu}_{h_t,i_t}(t) + \nu_1 \rho^{h_t}\left(2^{kd_C}\right)^r + c\sqrt{\frac{\log\left(1/\widetilde{\delta}\left(t^+\right)\right)}{T_{h_t,i_t}(t)}}. \tag{3}$$

For each node $(h,i) \in \mathcal{T}_t$, we estimate the feedback upper bound of node (h,i) like the one above. The first term is the feedback of practical exploration. We introduce the third uncertainty term. The sum of the above two terms is counted as the *upper confidence bound* of $\mu_{o,c}$. The second term is the maximum node size, where k is the times the square has been divided, and r is the adjustable parameter relating to d_C and d_O. We use it to balance the rate and accuracy of finding optimal traveling ocean, neither too quick nor too precise to regret.

While the MEF algorithm introduces the tighter feedback bound $B_{h,i}(t)$ to balance the father nodes and the child nodes.

To calculate the B-value of node (h, i)

$$B_{h,i}(t) = \begin{cases} U_{h,i}(t), & (h,i) \in \text{leaf}(\mathcal{T}_t) \\ \infty, & T_{h,i}(t) = 0 \\ \min\left[U_{h,i}(t), \max\limits_{j\in\{2i-1,2i\}} B_{h+1,j}(t)\right], & \text{otherwise.} \end{cases} \tag{4}$$

We need to obtain its child's B-value first. Thus, the updating process should begin from the leaf nodes $(H(t), i)_{1\le i\le 2^{H(t)}}$ of the tree $\mathcal{T}_t$ (line 6), and the updating process should repeat at each time slot.

b) Searching output ocean subdomain: The proposed algorithm gives consideration for both the high feedback and the unexplored area. To seek the most confident OS, MEF calls the *OptPath* function (auxiliary) to select the child node possessing the maximum B-value from the root node, which will be demonstrated in the auxiliary algorithm part.

c) Expanding: Reconsider (3), the last two terms calculate the uncertainty of the experienced feedback. The second term is on behalf of the most possible gap of two OSs in the same ocean cluster, and the last term denotes the contingency because of the limited exploration times which decreases with the number of explorations. When OSs in a leaf node have been explored adequately, we explore it in a more accurate way to subdivide the leaf node into two child nodes of $\mathcal{T}_t$. Thus, we set a threshold $\tau_h(t)$ to control the subdivision of ocean, and the expanding happens when $T_{h,i}(t) \ge \tau_h(t)$.

This expansion is a dynamic and top-to-down (from the father node to the child node) process. It only considers the range that each node has so that it can search for better path more accurately in OS to *ocean subdomain* but no influence on the binary tree. Note that the ocean is wide enough and the vessel's navigation width is small, even for a long time running the OS can still voyage for many vessels. Thus, the balance between $\tau_{h,k}$ and t, k can be expressed as

$$\begin{aligned} \nu_1 \rho^h \left(2^{d_C k}\right)^r &= c\sqrt{\frac{\ln\left[1/\widetilde{\delta}\left(t^+\right)\right]}{\tau_h(t)}} \Rightarrow \tau_{h,k}(t) \\ &= \frac{c^2 \ln\left(\frac{1}{\widetilde{\delta}(t^+)}\right)\rho^{-2h}}{{\nu_1}^2 2^{k d_C r}}. \end{aligned} \tag{5}$$

B. Algorithm Description of Auxiliary Algorithm

1) Predecision: We first show how predecision algorithm works in Algorithm 2. Then, a robust model possesses a man–machine interaction which helps us control the process and minimize the errors. Thus, we set a negative feedback by adding human manipulating for the MEF model. We collect the data every Δt time interval and analyze it by LP, submitting and waiting for the decision. After we input the total feedback and coefficient *IF*, in line 1, we use LP to analyze the battlefield situation and analyze in the data center. In line 2, we verify if our vessels are enough. If not, we inform the data center and ask for the supplementary, e.g., requiring more vessels for reinforce exploration or stopping this ocean's exploration and selecting another.

2) OptTree: The auxiliary algorithm (Algorithm 3) can be seen as a partition for vessel contexts and build a tree for each division. When a vessel with $c(t)$ asks for dispatching, we receive $c(t)$ and search for the small partition that it belongs based on historical data in the context set (line 1). We set m to count the partition times and initialize it to zero. We verify that if the partition $c(t)$ belonging has established a cover tree and recorded in the storage *built-up-context* (line 3). If yes, we directly output the established point set $\mathcal{T}_t$ (line 4). If not, we proceed the partition process. We calculate the points in the undivided area, and if there are enough points in the area, we divide it equally into two parts and increase the threshold (line 6). The judgment begins with the most significant context dimension d_1 to d_{d_C}. Every time we subdivide the partition, we add m and use m' to denote the subdividing context dimension (lines 7–9). Finally, we save this built-up-context to the storage for next time search and output the point set $\mathcal{T}_t$. In this case, $\mathcal{T}_t$ cannot be recognized as a tree because it only contains some points and after updating we call it a binary tree.

3) Optpath: We propose the *Optpath* function in Algorithm 4 to search the child node possessing the maximum B-value from the ancestor node, which means our most confident ocean. We first receive a point set $\mathcal{T}$, and our task is to construct a binary tree for these points. Then we construct the root and initialize it with the entire ocean space set (line 14). Afterward, we construct a traversal path R_t and end at (h_t, i_t), which should end at a leaf node (line 2). Thus, we propose the threshold as

$$\tau_{h,k}(t) = \frac{c^2 \ln\left(1/\widetilde{\delta}\left(t^+\right)\right)\rho^{-2h}}{{\nu_1}^2 2^{k d_C r}} \tag{6}$$

which is used to decrease the depth of tree $H(t)$ and guarantee each node explored adequately such that we are more

Algorithm 2: Predecision

Input: *IF*
1 Linear Programming for data analysis and Strategy decision making;
2 **if** $F(IF) \ge Constrain(IF)$ **then**
3 Update the vessels : $w_{updated} = w_{remained} \bigcup w_{supplement}$;

Algorithm 3: OptTree

Input: $c(t)$
1 There are some tested vessels' feedback and historical records, then a new vessel with $c(t)$ comes, searching for its context domain and building a cover tree **Initialize** : m = 0 (m is the partition times);
2 **if** $c(t) \in$ *built-up-context* **then**
3 **output** : $\mathcal{T}_t$ and exit the procedure;
4 **else**
5 **while** $N_S(c(t)) \ge \tau_c^m(t)$ **do**
6 **if** $N_S^{m'}(c(t)) \ge \tau_c^m(t)$ **then**
7 dichotomize the context subdomain in m';
8 $m = m + 1$; $m' = m \mod d_C$;
9 built-up-context $\leftarrow \mathcal{T}_t \leftarrow \mathcal{T}(\mathcal{CIR}(c(t), \rho_t^c))$;
10 **output** : $\mathcal{T}_t$ and exit the procedure;

Algorithm 4: OptPath

Input: Tree $\mathcal{T}_t$
1 **Initialize** : $(h, i) \leftarrow (0, 1)$, $R \leftarrow (0, 1)$, $T_{0,1}(t) = \tau_h(t) = 1$;
2 **while** $T_{h,i}(t) \ge \tau_h(t)$ *AND* $(h, i) \notin leaf(\mathcal{T})$ **do**
3 **if** $B_{h+1,2i-1} \ge B_{h+1,2i}$ **then**
4 $(h, i) \leftarrow (h + 1, 2i - 1)$;
5 **else**
6 $(h, i) \leftarrow (h + 1, 2i)$;
7 $R \leftarrow R \bigcup \{(h, i)\}$;
8 **Output** : (h, i) and R;

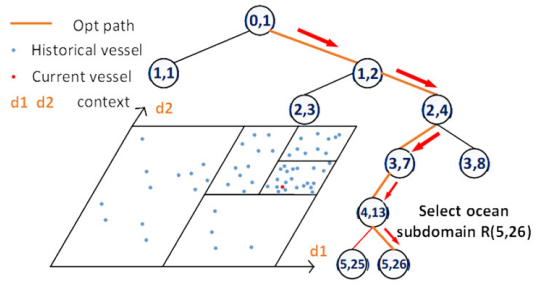


Fig. 2. Tree-based ocean and node selection.

confident about the acquired feedback $\widehat{\mu}_{h_t,i_t}(t)$. As the experience of vessels and ocean feedback increases, $\tau_h(t)$ becomes smaller. For example, in Fig. 2, when a vessel with context $c(t)$ comes, we compare the B-value and choose a confident path (lines 16–19). The most confident ocean o_t^* in $\mathcal{R}_{5,26}$, but the optimal OS is $\mathcal{R}_{5,25}$. The MEF selects $\mathcal{R}_{5,26}$ as output and obtains the path from $\mathcal{R}_{0,1}$ to $\mathcal{R}_{5,26}$ (lines 20 and 21), which will cause the deviation called regret.

C. Regret Analysis

In this section, we set the regret bound, which is the upper limitation to the regret, guaranteeing the performance in the worst case. We first prove that the cover tree will not expand too fast to try sufficiently for every subdomain, which may cause a huge deviation to the best path.

Lemma 1: Given the exploration threshold $\tau_h(t)$ in (5), the depth of the tree $\mathcal{T}_n$ can be bounded as

$$H(n) \leq \frac{1}{2(1-\rho)} \ln\left(\frac{n\nu_1^2(\frac{n}{\lambda})^{\frac{r}{2d_C}}}{(c\rho)^2}\right).$$

Proof: See proof in the supplementary material. ■

Lemma 1 proves that the exploration cover tree expands at the speed $\mathcal{O}(\log n)$, which promises sublinear cost to traverse the cover tree and bound the regret. The smaller threshold may lead to higher cost on exploring a suboptimal branch or a longer traverse path, thus the exploration accuracy is also guaranteed by the moderate speed.

Lemma 2: We define the set for all nodes which is possible in trees under the maximum depth $H_{\max}(t)$ as $\mathcal{N}_t = \bigcup_{\mathcal{T}:\text{Depth}(\mathcal{T})\leq H_{\max}(t)} Nodes(\mathcal{T})$. The high probability event can be defined as

$$\varepsilon_t = \left\{ \forall (h,i) \in \mathcal{N}_t \quad \forall T_{h,i}(t) = 1, \ldots, t : \right.$$
$$\left. \left|\widehat{\mu}_{h,i}(t) - \mathbb{E}\widehat{\mu}_{h,i}(t)\right| \leq c\sqrt{\frac{\log\left[1/\widetilde{\delta}(t)\right]}{T_{h,i}(t)}} \right\}.$$

If $c = 2\sqrt{1/1-\rho}$ and $\widetilde{\delta}(t) = \delta\sqrt[8]{\rho/3\nu_1}/t$, the event ε_t holds with probability with a minimum value $1-(\delta/t^5)(1/\lambda)^{(1/d_C)}$.

Proof: See proof in the supplementary material. ■

By using Lemma 2, we limit the gap between the estimated feedback and the genuine feedback of the suboptimal nodes. We use probability estimation to assure the circumstance in which the exploration can hardly happen, and even if it happens the average impact is also controllable. Using the above two lemmas, we obtain the regret bound in Theorem 1.

Theorem 1 (Regret Bound of MEF): Let oceans' feedback be i.i.d., Assumptions 1 and 2 hold at each moment t. Under the same condition in Lemma 2, the genuine regret of MEF RE_n up to time n is

$$\begin{aligned}
\text{RE}_n &\leq \text{RE}_n(\text{single}-\text{context}-\text{partition}) \times 2^k \\
&\leq 2^k \times \left(\frac{5\delta}{4}\left(\frac{1}{\lambda}\right)^{\frac{1}{d_C}}\left(\frac{n}{\lambda}\right)^{\frac{1}{2d_C}} + 6L_c\left(1+\sqrt{d_C}\right)\left(\frac{n}{\lambda}\right)^{\frac{1}{2d_C}}\right) \\
&\quad + \sqrt{\frac{C_V C_C \rho^{d_O-1}\nu_2^{-d_O}}{(1-\rho)\left(1-\rho^{(d_O+1)}\right)}\ln\left(2\frac{\sqrt[8]{3\nu_1}n}{\delta\sqrt[8]{\rho}}\right)n\left(\rho_n^c\right)^{-d_C}\rho^{-Hd_O}} \\
&\quad \times 2^k \times 96\left(\frac{n}{\lambda}\right)^{\frac{1}{2d_C}} \\
&\leq \frac{5\delta}{4}\left(\frac{1}{\lambda}\right)^{\frac{1}{d_C}}\left(\frac{n}{\lambda}\right)^{\frac{1}{2d_C}} + 6L_c\left(1+2\sqrt{d_C}\right)\left(\frac{n}{\lambda}\right)^{\frac{1}{2d_C}} \\
&\quad + 96\left(\frac{n}{\lambda}\right)^{\frac{1}{2d_C}}\left(\frac{C_V C_C \rho^{d_O-1}\nu_2^{-d_O}}{(1-\rho)\left(1-\rho^{(d_O+1)}\right)}\ln\left(2\frac{\sqrt[8]{3\nu_1}n}{\delta\sqrt[8]{\rho}}\right)\right)^{\frac{2}{d_O+2}} \\
&\quad \times \left(\frac{4}{\nu_1^2}\right)^{-\frac{d_O}{d_O+2}} \lambda^{-\left(\frac{(r+1)d_O+2}{4(d_O+2)}\right)}\sqrt{n^{\left(\frac{3}{2}+\frac{(r+1)d_O}{2(d_O+2)}+\frac{1}{d_C}\right)}}. \qquad (7)
\end{aligned}$$

Proof: See proof in the supplementary material. ■

Equation (7) in Theorem 1 upper bounds the gap between the optimal choice and the genuine feedback. We have proved that the MEF Algorithm achieves sublinear regret $\text{RE}_n = \mathcal{O}(n^{([3/4]+([(r+1)d_O]/[4(d_O+2)])+[1/2d_C])})$, which promises when the parameters values' are suitable, it converges in terms of the average feedback. As the equation shows, when $d_C > 2$ and $0 \leq r \leq ([2d_C - 2d_O - 4]/[d_C d_O])$, the algorithm satisfies the condition of sublinear, which means that we can use this algorithm when the context is large. In fact, when we classify a kind of vessel, d_C is always large, e.g., speed, the thickness of the deck, and the artillery configuration. In contrast, in the ocean exploration, we abstract $d_O = 1$ (longitude and latitude) to satisfy the bound condition. We set the confidence degree $\delta = 0.03$ empirically to guarantee the running performance of the MEF model. Typically, we set $\nu_1 = \sup_{o_1,o_2\in\mathcal{O}} f(o_1,o_2)$. Moreover, $\mu_t^* - \mu_{h_t,i_t}(t) \leq 6L_c(1+2\sqrt{d_C}) + 6\nu_1\rho^{h_t^p}(2^{kd_c})^r$ shows the regret increases with the larger ν_1, which means that we should also consider the ocean's changing rate of genuine feedback called gentle degree, that is, the more gentle the ocean is, the more accurate and efficient our algorithm is.

V. Numerical Results

In this section, we verify the performance of the proposed regret bound for the MEF algorithm with experimental results based on the big data datasets. We assess the MEF algorithm in three aspects: 1) learning performance includes the speed of MEF enhanced; 2) individual exploration (context awareness) includes the accuracy and recall of MEF enhanced by the aggregation and partition of context; and 3) acceleration of the context partition.

The intention of 1) is to compare the MEF algorithm with other online learning and hashing approaches. From comparisons, we show the outstanding learning performance of the MEF as an online learning algorithm. Considering the logistical challenges of establishing a system to run the MEF algorithms by live data, the experiments are run on offline data. We establish a simulation platform [45], involving an ocean space, a set of dispatching situations and a set of assumption

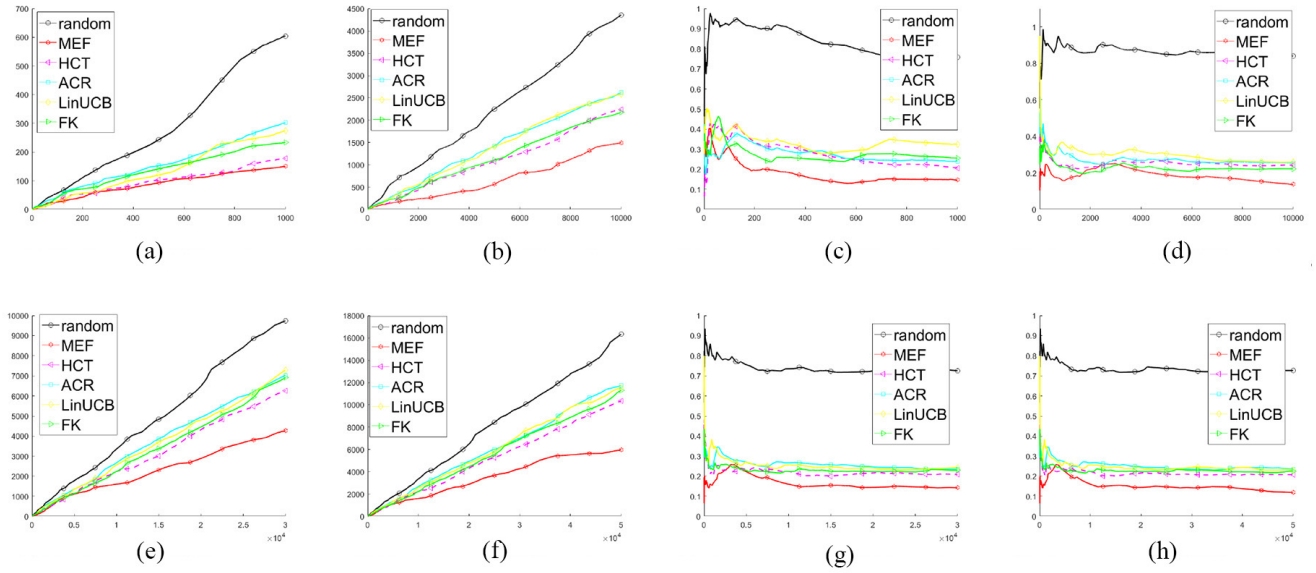


Fig. 3. Performance of MEF and other algorithms under context-free circumstance. Different vessel types execute one mission. (a) Total regret: 1000 try times. (b) Total regret: 10 000 try times. (c) Average regret: 1000 try times. (d) Average regret: 10 000 try times. (e) Total regret: 30 000 try times. (f) Total regret: 50 000 try times. (g) Average regret: 30 000 try times. (h) Average regret: 50 000 try times.

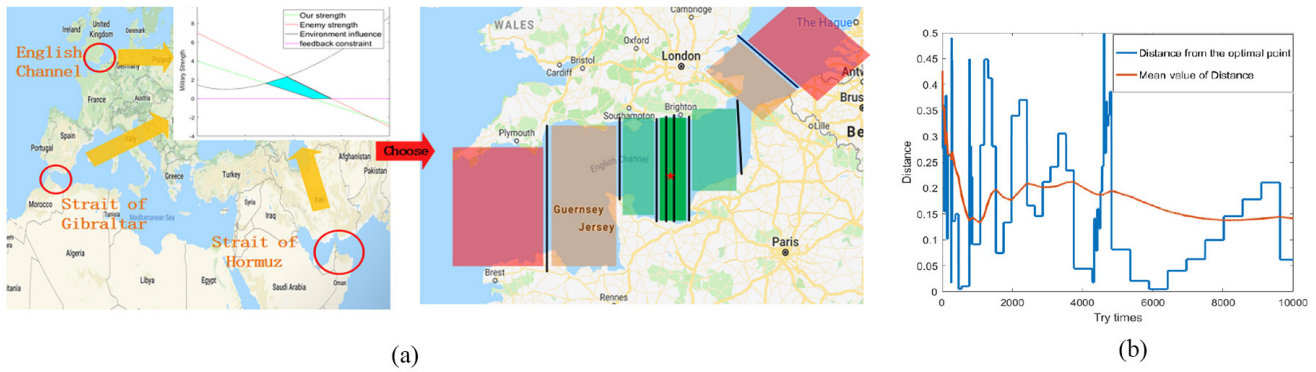


Fig. 4. Experiment on the picture of real product, ocean division, and distance to the optimal path. Maps by Google map. (a) OS in the practical situation. (b) Distance to the optimal path (the path through right star).

vessel context and ocean environment, to evaluate the accuracy and speed of ocean exploration. All the experiments are implemented and operated on our high-performance computing platform, whose GPU reaches 18.46 TFlops, and SSD cache is 1.00 TB. We assume the ocean is 400-miles long, 85-miles wide in the English Channel. The weather on the ocean is changing in a random range. We treat the ocean as an entire area and explore it. Every vessel is seen at one point. Afterward for the intention of 2), to evaluate the improvement of utilizing the vessel context, we test the context with real-world vessel configurations. For 3), we compare the MEF algorithm with context-aware algorithm ACT and SACT [44], which shows that MEF achieves improvements in both convergence speed and exploration accuracy.

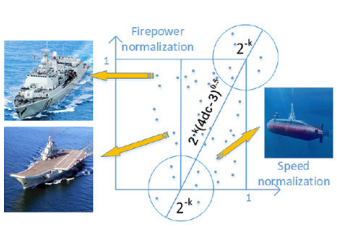
A. Datasets Description

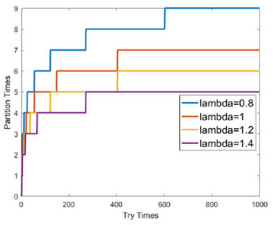
In this article, we use the data from U.S. Government's public data.[1] We analyze the character of vessels from the total tonnage (foreign and domestic) of commodities carried on commercial waterways and the freight analysis framework. Then, we refine the features of the vessel by the latent Dirichlet allocation (LDA) method to obtain a 10-D vessel context. To obtain the ocean characters, we find the data from USCG vessel, including the marine casualty and pollution data files, which provides details about marine casualty and pollution incidents investigated by Coast Guard Offices throughout the United States. Considering that the main part of it is useless, we also refine the features out of it and build a 3-D ocean environment.

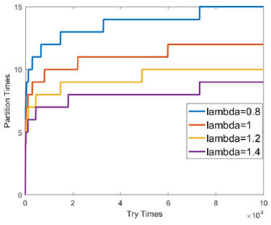
For the vessel context experiment, we abstract the vessel features, including tonnage, vessel speed, vessel type (e.g., cargo or passenger ship), the thickness of the deck, personnel allocation, tags, nationality, fire power allocation, vessel materials, and vessel volume, recording the distribution characters, that is, $d_C = 10$. The example of abstraction is in Fig. 5 (only showing 6-D). Then we run the code with the distribution characters to demonstrate the MEF algorithm. As for the ocean features, we abstract $d_C = 3$, including

[1]https://www.data.gov/maritime/ & https://www.data.gov/ocean/

	Kitty Hawk	Ticonderoga	Spruance-class
Nationality	USA	USA	USA
Vessel type	Aircraft carrier	Guided missile cruiser	Guided missile destroyer
Tonnage	81780t	7652t	8040t
Vessel speed	32kn	30.5kn	33kn
Vessel length	318.5m	172.8m	171.6m
Vessel width	39.6m	16.8m	16.76m







(a) (b) (c) (d)

Fig. 5. Demonstrations of context partition approach and partition times. (a) Vessel character for context partition. (b) Vessel context partition. (c) Max context partition times of the first 1000 try times. (d) Max context partition times of 10^5 try times.

the depth of water, the longitude, and the latitude from the ocean dataset.

We first normalize the abstract data to [0, 1]. While for the aforementioned features, the high-dimensional features are difficult to simulate. Thus, we adopt the mean values [46] to reduce the dimension of features for simplicity. Finally, we obtain an 8-D feature vector of a vessel, and a 1-D feature vector of an ocean area. On the basis of those simplified vectors, we set the parameters $\nu_1 = 2\sqrt[4]{18}$ to assure Lemma 1. When expanding the binary exploration cover tree, we equally divide each OS cluster, hence $\rho = 1/2$. As for the intention to assure the performance, we empirically set the confidence parameter $\delta = 0.06$. For simplicity, we utilize the Euclidean norm as the above-mentioned dissimilarity functions for the two spaces.

B. Experimental Setting

To evaluate the exploration quality, we compare our algorithm with several existing congeneric algorithms [47], including the most classical MAB algorithm with finite arms [48], and two bandit algorithms supporting big data exploitation, that is, high confidence tree algorithm (HCT) [41] and ACR algorithm [43]. For a recent modified algorithm, we present a relevant feedback algorithm, that is, the Fisher kernel (FK) relevance approach [49]. Note that the HCT is a context-free algorithm, we treat all inputs to it as one type. For the ACR, the feedback $E = \min\{L, \lfloor \log_2(T) \rfloor\}$. Since each leaf node in the ACR only includes one item, the depth of the tree is $L = 13$, and the time horizon $T = 10^4$, thus $E = \lfloor \log_2(T) \rfloor = 12$. For the hashing algorithm, we take 10% of datasets as the request sets and the rest as the training and exploration sets. For the FK RF algorithm, we choose their global FK RBF framework [49] to test due to their slow frame aggregation.

1) Learning Ability Examining: To achieve the improvement in individuation, we decrease the collected vessels' characteristics and exploration time as an 8-D vectors and aggregate them with a request as the context input. The context input is similar to the Mission request, which means after a vessel with context requesting for a mission, we send the abstract 8-D vectors to the center. Under the datasets aforementioned, we run the five algorithms *without context*, respectively, setting $n = 100\,000$.

2) Individuation Exploration: Under the datasets aforementioned, we run the eight algorithms *with context* (contextual) and set $T = 100\,000$. For the contextual task, vessels' characters input are reduced to $d_C = 8$ as a contextual group with the same process mentioned above. Then to eliminate the influence of the number of context dimensions, it was generated as $d_C = 8$ by setting all vessels' characters to the same for a context-free group. The two groups will be utilized as input to explore the ocean dataset ($d_O = 1$), respectively. To obtain an unbiased estimation of the proposed online learning algorithms by this offline simulation, we conduct an experiment [46], that is, at each moment we repeatedly pick exploring events randomly until the output w_t being the part of the corresponding exploring result. Then, we set accurate (feedback) $fb_t = \mathbb{I}\{w_t$ was selected$\}$.

3) Acceleration Partition: To prove the acceleration ability of the proposed MEF algorithm, we compare the MEF with context partition algorithm SACT [44] and other general contextual algorithms. Because of the different context using ways, we mainly show how much improvement we have made to compare with the ACT and SACT. Moreover, we use a random algorithm which randomly selects an OS at each round as the worst algorithm in contrast.

C. Testing Learning Performance

In this section, we show our learning performance mainly via two standards: 1) the MEF algorithm's regret is sublinear and the time average regret converges to 0 and 2) the MEF algorithm disposes various request vessel types. First, we run the experiment in an ocean dataset. The results are shown in Fig. 3. Dispatching the different types of vessels to the same OS for the same exploration mission. For example, we send destroyers and aircrafts to the same path, which is somewhat inefficient. The first 1000 points we can see in the down right corner, which cannot completely embody MEF's advantage because of the exploiting feature of the algorithm, but when we try more times, the regret possesses an obvious low regret. Under this circumstance, we conclude that we have the minimum total regret and average regret among the six algorithms, which proves standard 2). In the center of Fig. 3, it is the total regret in the 10^4 exploration times, and we can see that the red curve is the proposed algorithm's excellent regret, about 1500 in 10^4 iteration times. When we try more times [Fig. 3(e)–(h)], we can see the MEF has a smoother regret curve and performs the best. The trend of average regret converges to 0, that is, all the five algorithms except random can do this but we have the most efficient method, which proves standard 1).

Fig. 4(a) shows that the OS division in the practical ocean. Red subdomains means that it is more dangerous and has low feedback with our exploration in the ocean. The bottle green OS means that it is safer and the path through red star is the

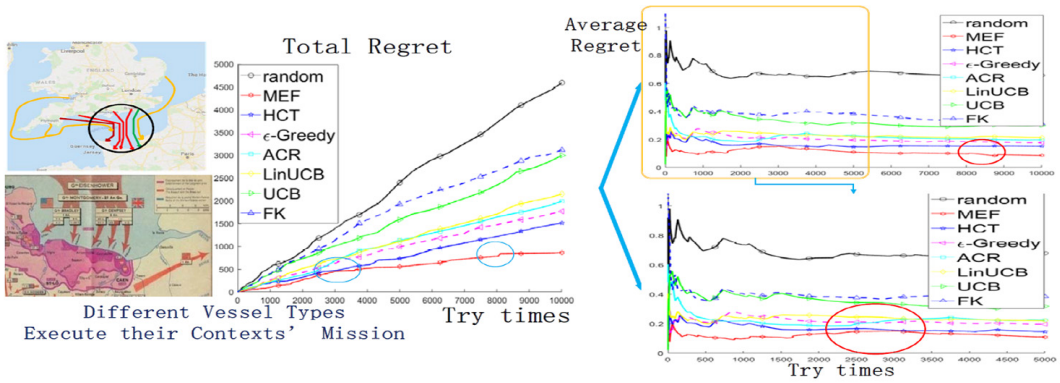


Fig. 6. Performance of MEF and other algorithms under contextual circumstance. Maps by Google map.

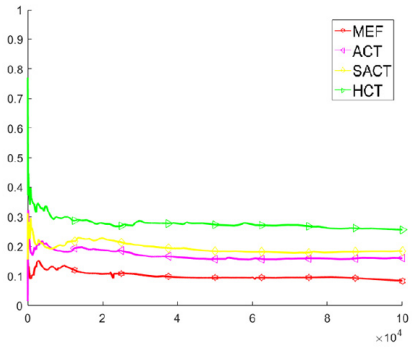


Fig. 7. Average regret for four algorithms.

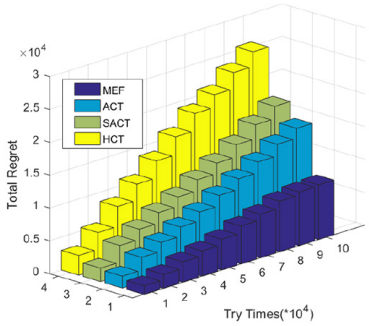


Fig. 8. Regret of four algorithms with the 3-D performance of variation trend.

optimal path. Fig. 4(b) shows the distance from the path we choose to the optimal path. We can see that at the first 5000 iteration times, the distance to the optimal path there is a great fluctuation around 0.2 and our choice is more accurate with larger trying times.

D. Testing Individuation Exploration

In the proposed algorithm, we improve the performance in the context space, considering the different vessel types for different individual missions. We practically classified the vessels contexts in Fig. 5(a) and partition the context space shown in Fig. 5(b). Fig. 6 shows that the destroyers and aircrafts have their own individual mission, e.g., aircrafts searching for the short path in the middle English channel while the destroyers to the longer periphery path. Thus, we demonstrate the proposed algorithm in the regret of MEF and in the adaptive ability to the vessel context.

1) Regret of MEF: The MEF experiences a remarkable decrease in the total regret and average regret. In the total regret figure, the saltus steps are at about 3000 and 8000 iteration times, which is caused by a special partition and threshold. This means that when we subdivide the partition area, causing the number of points to decrease and thus reduce the accuracy temporarily. We can endure the accuracy decreasing because the transient decrease can lead to larger gain. The experiment shows that after the transient accuracy decrease, we have a lower slope of the total regret curve at about 20%.

2) Adaptive Ability to the Vessel Context: In a long-time big data exploration, we have a pre-eminent regret curve which has significant advantages among eight algorithms compared with the context-free model. This is MEF's contextual advantages. We have increased accuracy by 19%, which shows the outstanding adaptive ability to the context in Table II. (In the 0.1×10^4 column.) Fig. 5(a) and (c) shows that when we try enough times, the maximum partition times in the context space are about 10. It means that in our vessels' context space, the smallest vessel context has been partitioned ten times. Take one dimension context as an example, the initial tonnage partition is $[10, 20]$ ton and the final is $[10, 10 + (10/2^{10})]$ ton. Moreover, different initial parameter causes different partition rate. The smaller λ means that we are more confident with the vessels' character classification. Moreover, the partition times' increasing speed becomes lower as the iteration times increase, which is caused by the increasing $\tau_c(t)$ (2).

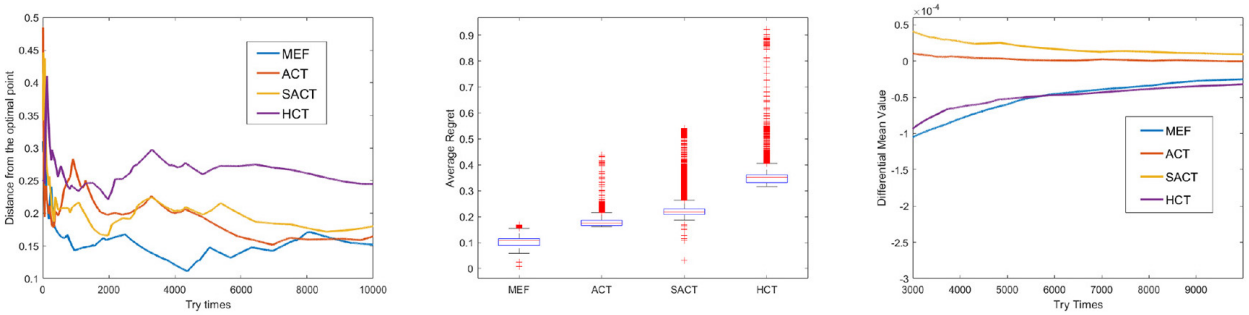
E. Testing the Acceleration of Threshold Change

Fig. 7 shows the average regret for the four algorithms. We can see the MEF possesses the lowest regret, which embodies our algorithm's acceleration character. Fig. 8 shows the total regret of the four algorithms, and we can see the MEF's advantage by comparison. Then, for quantitative analysis, we propose the magnification in Table II. We define the accuracy magnification of algorithm ξ_1 to the algorithm ξ_2 as $(AC_{\xi_1,t} - AC_{\xi_2,t})/AC_{\xi_2,t}$, where $AC_{\xi,t}$ indicates the average accuracy of that algorithm at time t.

Under context-free circumstance, we try enough times (10×10^4), and have only 30% gain over HCT and even have the same gain with ACT and SACT. While under contextual circumstance, when we try 10×10^4 times, the MEF outperforms about 60% over HCT and 30% over ACT and SACT, which indicates our algorithm is relatively more dependent

TABLE II
INFLUENCE OF CONTEXT TO THE FOUR ALGORITHMS

Task	Algorithm	Retrieval Times $\times 10^4$ (context-free)					Retrieval Times $\times 10^4$ (contextual)				
		0.1	1	5	8	10	0.1	1	5	8	10
Average Accuracies	MEF	55.23%	63.99%	67.58%	71.31%	71.61%	74.58%	82.63%	88.38%	91.20%	91.45%
	ACT	52.38%	60.82%	64.38%	67.63%	68.91%	59.21%	66.39%	72.64%	73.32%	74.35%
	SACT	50.92%	59.36%	62.92%	66.17%	67.46%	57.75%	64.93%	71.18%	71.86%	72.89%
	HCT	43.31%	49.81%	53.58%	55.23%	55.66%	47.46%	51.38%	54.81%	56.32%	58.16%
Gain	MEF over HCT	27.52%	28.47%	26.69%	27.85%	29.20%	57.14%	60.82%	60.28%	61.19%	62.54%
	MEF over SACT	8.46%	7.79%	7.88%	6.71%	6.59%	29.14%	27.25%	24.16%	26.91%	25.46%
	MEF over ACT	5.44%	5.21%	5.20%	4.34%	4.21%	25.96%	24.46%	21.67%	24.39%	23.14%



(a) (b) (c)

Fig. 9. Differential value of the four compared algorithms. (a) Distance to the optimal point. (b) Dispersion degree. (c) Mean differential average regret.

on the context. In our experiment, the vessel context has a high randomness which means the vessels may have a peculiar character. Nevertheless, in the real world, the produced vessels have a specification, which makes the learning process easier.

Fig. 9(a) shows the distance from the optimal OS to the subdomain that we choose. The MEF possesses the lowest distance to the optimal ocean at around 0.15. The box figure of four algorithms in the middle of Fig. 9(b) shows the dispersion of average regret. We can see that the MEF possesses the lowest mean regret value of around 0.1 (center of the square) and the comparative stable regret distribution which is a little more unstable than the ACT with regret about 0.18. In Fig. 9(c), we can see the MEF does not have the lowest average regret growth degree, while it has the lowest average regret because of the outstanding exploring in the first 3000 iteration times. This means that we choose a well-behaved ocean at the beginning.

VI. CONCLUSION AND DISCUSSION

In this article, we proposed a contextual online learning approach MEF for solving the big data ocean exploration problem in cybernetics. We use a novel top-to-down tree structure-based CMAB algorithm and improve its accuracy from 71% to 91% by adding the practical vessel context space and adaptively adjusting the exploration threshold. Further, when we explore the ocean enough times, the proposed MEF algorithm achieves a minimum 25% improvement in performance in accuracy over other existing contextual algorithms used for comparison. With the contexts fully utilized, numerical results have shown that the learning performance of the proposed approaches have been optimized constantly.

However, in the real world, we need an extensive assistance from all departments, which can be belated and complex. Moreover, we assume we have all kinds of infinite vessels that can be dispatched, which is limited by real-world production. We also assume the ocean be an explored area. Our vessels are the executing points. The vessels travel on the ocean to explore the optimal area and then find the optimal path. In addition, we use vessels contexts to select better vessel paths for different kind of them. In the experiment, we assume the environment of the ocean is changing but in the time duration, the expectation value of the feedback is constant. In Figs. 3–9, we can see in the English Channel there is an increasing accuracy in the exploration of the ocean. In Fig. 4, we can see under our ocean assumption, the central ocean channel with a star is the best path for the vessels. Although we have made some progress in the exploration while we assume the ocean's feedback is constant, while if there is a changing in the ocean's feedback, we may need an adaptive algorithm to solve this problem, e.g., treat it as a random process problem.

REFERENCES

[1] M. G. Parsons, A. C. Chubb, and Y. Cao, "An assessment of fuzzy logic vessel path control," *IEEE J. Ocean. Eng.*, vol. 20, no. 4, pp. 276–284, Oct. 1995.

[2] Y. Shu, W. Daamen, H. Ligteringen, and S. Hoogendoorn, "Vessel speed, course, and path analysis in the Botlek area of the Port of Rotterdam, Netherlands," *Transp. Res. Rec.*, vol. 2330, no. 1, pp. 63–72, 2013.

[3] J. Li and Y. V. Zakharov, "Efficient use of space-time clustering for underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 43, no. 1, pp. 173–183, Jan. 2018.

[4] J. Li, Y. V. Zakharov, and B. Henson, "Multibranch autocorrelation method for Doppler estimation in underwater acoustic channels," *IEEE J. Ocean. Eng.*, vol. 43, no. 4, pp. 1099–1113, Oct. 2018.

[5] J. Li, P. R. White, J. M. Bull, and T. G. Leighton, "A noise impact assessment model for passive acoustic measurements of seabed gas fluxes," *Ocean Eng.*, vol. 183, no. 1, pp. 294–304, 2019.

[6] F. Yuan, Z. Jia, J. Li, and E. Cheng, "STLFM signal based adaptive synchronization for underwater acoustic communications," *IEEE Access*, vol. 7, pp. 28734–28748, 2019.

[7] J. Li, L. Liao, and Y. V. Zakharov, "Space-time cluster combining for UWA communications," in *Proc. IEEE OCEANS Shanghai*, 2016, pp. 1–6.

[8] J. Li, "DOA tracking in time-varying underwater acoustic communication channels," in *Proc. IEEE OCEANS Aberdeen*, 2017, pp. 1–9.

[9] Y. Zhang, T. Wu, Y. Zakharov, and J. Li, "MMP-DCD-CV based sparse channel estimation algorithm for underwater acoustic transform domain communication system," *Appl. Acoust.*, vol. 154, pp. 43–52, Nov. 2019.

[10] Z. Zeng, K. Sammut, L. Lian, A. Lammas, F. He, and Y. Tang, "Rendezvous path planning for multiple autonomous marine vehicles," *IEEE J. Ocean. Eng.*, vol. 43, no. 3, pp. 640–664, Jul. 2018.

[11] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for UAV navigation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 6, pp. 898–912, Dec. 2003.

[12] H. Kim, D. Kim, J.-U. Shin, H. Kim, and H. Myung, "Angular rate-constrained path planning algorithm for unmanned surface vehicles," *Ocean Eng.*, vol. 84, pp. 37–44, Jul. 2014.

[13] T. Oral and F. Polat, "MOD* Lite: An incremental path planning algorithm taking care of multiple objectives," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 245–257, Jan. 2016.

[14] A. Macwan, J. Vilela, G. Nejat, and B. Benhabib, "A multirobot path-planning strategy for autonomous wilderness search and rescue," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1784–1797, Sep. 2015.

[15] S.-H. Ryu, Y. Kang, S.-J. Kim, K. Lee, B.-J. You, and N. L. Doh, "Humanoid path planning from HRI perspective: A scalable approach via waypoints with a time index," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 217–229, Feb. 2013.

[16] J. Wang *et al.*, "Doppler shift estimation for space-based AIS signals over satellite-to-ship links," *IEEE Access*, vol. 7, pp. 76250–76262, 2019.

[17] F. Deng, S. Guo, Y. Deng, H. Chu, Q. Zhu, and F. Sun, "Vessel track information mining using AIS data," in *Proc. IEEE Int. Conf. Multisensor Fusion Inf. Integr. Intell. Syst. (MFI)*, 2014, pp. 1–6.

[18] D. Sidoti *et al.*, "Context-aware dynamic asset allocation for maritime interdiction operations," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[19] R. Lent, "Linear QoS goals of additive and concave metrics in ad hoc cognitive packet routing," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1255–1260, Dec. 2006.

[20] S. D. Bopardikar, S. L. Smith, and F. Bullo, "On dynamic vehicle routing with time constraints," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1524–1532, Dec. 2014.

[21] M. Pavone, N. Bisnik, E. Frazzoli, and V. Isler, "A stochastic and dynamic vehicle routing problem with time windows and customer impatience," *Mobile Netw. Appl.*, vol. 14, no. 3, pp. 350–364, 2009.

[22] D. Sidoti *et al.*, "A multiobjective path-planning algorithm with time windows for asset routing in a dynamic weather-impacted environment," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 12, pp. 3256–3271, Dec. 2017.

[23] D. Kumar, J. C. Bezdek, M. Palaniswami, S. Rajasegarar, C. Leckie, and T. C. Havens, "A hybrid approach to clustering in big data," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2372–2385, Oct. 2016.

[24] J. M. Luna, F. Padillo, M. Pechenizkiy, and S. Ventura, "*A priori* versions based on MapReduce for mining frequent patterns on big data," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2851–2865, Oct. 2018.

[25] T. Lai *et al.*, "Efficient robust model fitting for multistructure data using global greedy search," *IEEE Trans. Cybern.*, to be published.

[26] H. Li, X. Jing, H.-K. Lam, and P. Shi, "Fuzzy sampled-data control for uncertain vehicle suspension systems," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1111–1126, Jul. 2014.

[27] Z.-P. Wang and H.-N. Wu, "On fuzzy sampled-data control of chaotic systems via a time-dependent Lyapunov functional approach," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 819–829, Apr. 2015.

[28] H. Modares, F. L. Lewis, and Z.-P. Jiang, "Optimal output-feedback control of unknown continuous-time linear systems using off-policy reinforcement learning," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2401–2410, Nov. 2016.

[29] S. Bidkar, A. Gumaste, P. Ghodasara, A. Kushwaha, J. Wang, and A. Somani, "Scalable segment routing—A new paradigm for efficient service provider networking using carrier Ethernet advances," *J. Opt. Commun. Netw.*, vol. 7, no. 5, pp. 445–460, 2015.

[30] F. Zhu, "Mining ship spatial trajectory patterns from AIS database for maritime surveillance," in *Proc. 2nd IEEE Int. Conf. Emerg. Manag. Manag. Sci.*, 2011, pp. 772–775.

[31] B. I. Ahmad, J. K. Murphy, P. M. Langdon, and S. J. Godsill, "Bayesian intent prediction in object tracking using bridging distributions," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 215–227, Jan. 2018.

[32] X. Han *et al.*, "Optimization-based decision support software for a team-in-the-loop experiment: Multilevel asset allocation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 8, pp. 1098–1112, Aug. 2014.

[33] G. Qingji, Y. Yongsheng, and H. Dandan, "Feasible path search and optimization based on an improved algorithm," *China Civil Aviation College J.*, vol. 23, no. 4, pp. 42–44, 2005.

[34] E. Shan, B. Dai, J. Song, and Z. Sun, "A dynamic RRT path planning algorithm based on B-spline," in *Proc. 2nd Int. Symp. Comput. Intell. Design*, vol. 2, 2009, pp. 25–29.

[35] L. Lin and M. A. Goodrich, "Hierarchical heuristic search using a Gaussian mixture model for UAV coverage planning," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2532–2544, Dec. 2014.

[36] M. Eichhorn, "Solutions for practice-oriented requirements for optimal path planning for the AUV 'SLOCUM glider'," in *Proc. MTS/IEEE OCEANS Seattle*, 2010, pp. 1–10.

[37] S. Basagni, V. Di Valerio, P. Gjanci, and C. Petrioli, "Finding MARLIN: Exploiting multi-modal communications for reliable and low-latency underwater networking," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2017, pp. 1–9.

[38] P. Gift, "Planning for an adaptive evader with application to drug interdiction operations," Naval Postgrad. School, Monterey, CA, USA, Rep. 5161, 2010.

[39] M. Laskey *et al.*, "Multi-arm bandit models for 2D sample based grasp planning with uncertainty," in *Proc. IEEE Conf. Autom. Sci. Eng. (CASE)*, 2015, pp. 1–8.

[40] M. Phillips, V. Narayanan, S. Aine, and M. Likhachev, "Efficient search with an ensemble of heuristics," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 1–8.

[41] M. G. Azar, A. Lazaric, and E. Brunskill, "Online stochastic optimization under correlated bandit feedback," in *Proc. ICML*, 2014, pp. 1557–1565.

[42] B. Awerbuch and R. D. Kleinberg, "Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches," in *Proc. 36th Annu. ACM Symp. Theory Comput.*, 2004, pp. 45–53.

[43] L. Song, C. Tekin, and M. van der Schaar, "Online learning in large-scale contextual recommender systems," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 433–445, May/Jun. 2016.

[44] Y. Feng, P. Zhou, J. Xu, S. Ji, and D. O. Wu, "Video big data retrieval over media cloud: A context-aware online learning approach," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1762–1777, Jul. 2019.

[45] D. Vallet, P. Castells, M. Fernández, P. Mylonas, and Y. Avrithis, "Personalized content retrieval in context using ontological knowledge," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 3, pp. 336–346, Mar. 2007.

[46] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 661–670.

[47] J. Pietz, "A generalized orienteering problem for optimal search and interdiction planning," Naval Postgrad. School, Monterey, CA, USA, Rep. 37694, 2013.

[48] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, nos. 2–3, pp. 235–256, 2002.

[49] I. Mironică, B. Ionescu, J. Uijlings, and N. Sebe, "Fisher kernel temporal variation-based relevance feedback for video retrieval," *Comput. Vis. Image Understand.*, vol. 143, pp. 38–51, Feb. 2016.