An Adaptive Ant Colony System Based on Variable Range Receding Horizon Control for Berth Allocation Problem

Rong Wang, *Student Member, IEEE*, Fei Ji, *Member, IEEE*, Yi Jiang, *Student Member, IEEE*, Sheng-Hao Wu, *Student Member, IEEE*, Sam Kwong, *Fellow, IEEE*, Jun Zhang, *Fellow, IEEE*, and Zhi-Hui Zhan, *Senior Member, IEEE*

***Abstract*— The berth allocation problem (BAP) is an NP-hard problem in maritime traffic scheduling that significantly influences the operational efficiency of the container terminal. This paper formulates the BAP as a permutation-based combinatorial optimization problem and proposes an improved ant colony system (ACS) algorithm to solve it. The proposed ACS has three main contributions. First, an adaptive heuristic information (AHI) mechanism is proposed to help ACS handle the discrete and real-time difficulties of BAP. Second, to relieve the computational burden, a divide-and-conquer strategy based on variable-range receding horizon control (vRHC) is designed to divide the complete BAP into a set of sub-BAPs. Third, a partial solution memory (PSM) mechanism is proposed to accelerate the ACS convergence process in each receding horizon (i.e., each sub-BAP). The proposed algorithm is termed as adaptive ACS (AACS) with vRHC strategy and PSM mechanism. The performance of the AACS is comprehensively tested on a set of test cases with different scales. Experimental results show that the effectiveness and robustness of AACS are generally better than the compared state-of-the-art algorithms, including the well-performing adaptive evolutionary algorithm and ant colony optimization algorithm. Moreover, comprehensive investigations are conducted to evaluate the influences of the AHI mechanism, the vRHC strategy, and the PSM mechanism on the performance of the AACS algorithm.**

Manuscript received 9 November 2021; revised 22 March 2022; accepted 25 April 2022. Date of publication 18 May 2022; date of current version 7 November 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102102, in part by the National Natural Science Foundation of China (NSFC) under Grant 62176094 and Grant 61873097, in part by the Key-Area Research and Development of Guangdong Province under Grant 2020B010166002, in part by the Guangdong Natural Science Foundation Research Team under Grant 2018B030312003, and in part by the National Research Foundation of Korea under Grant NRF-2021H1D3A2A01082705. The Associate Editor for this article was L. Li. *(Corresponding authors: Fei Ji; Zhi-Hui Zhan.)*

Rong Wang and Fei Ji are with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: eefeiji@scut.edu.cn).

Yi Jiang, Sheng-Hao Wu, and Zhi-Hui Zhan are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, also with the Pazhou Laboratory, Guangzhou 510330, China, and also with the Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, Guangzhou 510006, China (e-mail: zhanapollo@163.com).

Sam Kwong is with the Department of Computer Science, City University of Hong Kong, Hong Kong.

Jun Zhang is with Zhejiang Normal University, Jinhua 321004, China, and also with Hanyang University, Ansan 15588, South Korea.

This article has supplementary downloadable material available at https://doi.org/10.1109/TITS.2022.3172719, provided by the authors.

Digital Object Identifier 10.1109/TITS.2022.3172719

***Index Terms*— Berth allocation problem (BAP), ant colony system (ACS), evolutionary computation (EC), variable-range receding horizon control, adaptive heuristic information.**

I. Introduction

THE globalization of the world economy has stimulated the rapid increase in import and export volumes of cargo and resulted in the development of the international logistics industry. Maritime transport is a vital link in the global logistics supply chain [1]. The United Nations Conference on Trade and Development indicates that approximately 80% of goods of international trade are carried by sea, and this figure is even higher among developing countries [2]. Container terminals are one of the critical components of the maritime transport system. Considering the ever-increasing demand for maritime transport, increases in container terminals' operational efficiency and productivity are in great need. Directly expanding the scale of container terminals or increasing the terminal resources is considered an unrealistic option because of natural and economic constraints. Instead, it is highly preferred to improve the operational efficiency of container terminal by reasonably allocating and scheduling existing resources. Since berths are a bottleneck resource in container terminal, berth allocation plays an essential role in operational efficiency.

Therefore, this paper studies the berth allocation problem (BAP) and proposes an efficient algorithm to allocate berthing berth and berthing time for each vessel. According to the actual situation of berths and vessels, BAP models can be categorized into four types: continuous BAP, discrete BAP, dynamic BAP, and static BAP. In a continuous BAP case, the container terminal is a whole platform, and multiple vessels can be berthed at a time. In a discrete BAP case, the container terminal is divided into several berths (i.e., multi-berth), and each berth serves only one vessel at a time. In the dynamic BAP case, vessels arrive at the container terminal at any time. However, in the static BAP case, the allocation is conducted after all vessels have arrived at the container terminal, which therefore incurs more management costs and requires a larger waiting area to berth all vessels. In addition, a continuous berth can be divided into several discrete berths according to the loading and unloading equipment of the container terminal. Therefore, this paper focuses on the discrete and dynamic BAP.

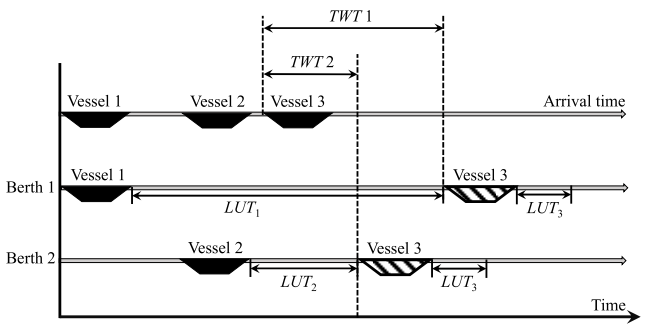

Fig. 1. Example of different vessels' berthing modes.

For the discrete dynamic BAP, various formulations to model the problem have been proposed. For instance, the minimization of the total cost was used as the objective in [3]. In [4], since the handling speed of each berth was different, minimizing the total service time which included all vessels' waiting time and handling time was used as an objective to evaluate the berth allocation plan. Minimization of the port penalty cost as the optimization objective was proposed in [5]. As environmental issues are taken seriously, carbon dioxide emissions costs due to container handling have also been used to assess berth allocation plans in [6]. Furthermore, in addition to berths, some recent studies also considered to schedule and allocate other resources at the container terminal, such as ship-to-ship cranes [7], quay cranes [8], yard cranes [9], containers [10], vehicles [11], and labor [12]. The availability of these equipment resources is a key factor in determining the speed of loading and unloading. In practice, the allocation of these equipment resources is usually the same at every discrete berth. Among the above existing BAP models, the waiting time of the vessel plays a key role in the objective function of each BAP model. Therefore, the minimization of total waiting time (*TWT*) is adopted as the optimization objective of our BAP model.

The BAP can be regarded as a kind of permutation-based combinatorial optimization problem, which is known to be NP-hard [13], [14]. A solution of the BAP is a permutation containing the berthing time and berthing berth of all vessels. Different berths and different berthing sequences will both affect the performance quality of the solution (i.e., the fitness value). For example, Fig. 1 shows the BAP case with three vessels and two discrete berths. Because the loading and unloading time (*LUT*) of vessel 1 is longer than that of vessel 2, the *TWT* can be reduced by berthing vessel 3 at berth 2 rather than at berth 1.

As an important branch of BAP research, many efforts on algorithm research have been conducted. A straightforward method to solve this problem is the first come first served (FCFS) approach. However, FCFS always obtains a fixed permutation with a poor objective value *TWT* [15]. In most cases, the *LUT* required for each vessel is different. If a vessel with a larger *LUT* value berths first, other vessels that berth later than this vessel will have a longer total waiting time. Therefore, instead of FCFS, various meta-heuristic algorithms have been proposed to solve the BAP. Cordeau *et al.* [16] presented two tabu search heuristics for solving the discrete BAP and the continuous BAP separately. Oliveira *et al.* [17] proposed a clustering search method in which a meta-heuristic simulated annealing was applied to generate BAP solutions. Furthermore, two simulated annealing variants improved by the restart strategy were proposed to solve the dynamic BAP in [18]. In addition, genetic algorithm (GA) and its variants, which are population-based meta-heuristics, have also been used for solving BAP [19], [20]. Dulebenets [19] proposed a novel evolutionary algorithm with an adaptive mechanism to solve BAP. The adaptive mechanism was a parameter control strategy developed for the mutation operator, in which the mutation rate was updated based on feedback from the search. Golias *et al.* [20] presented a GA-based heuristic was used to solve the discrete and dynamic BSP for the minimization of the total waiting and delayed departure time for all vessels.

Swarm intelligence algorithms are a class of meta-heuristic algorithms that imitate the foraging behavior of animals, which have also been utilized to solve the BAP, such as particle swarm optimization [21], artificial fish swarm optimization [22], and ant colony optimization (ACO) [23], [24]. However, the ACO variant proposed in [23] mainly focused on the continuous BAP, which was unsuitable for the discrete BAP. In [24], a greedy mechanism was used in ACO to help choose the berth that first became idle, which was suitable for the discrete BAP. Moreover, by using such a greedy algorithm, it is easy to become trapped in local optima. Therefore, to overcome the above drawbacks, we propose to use an ant colony system (ACS) algorithm to solve the discrete BAP. As an efficient and widely-used ACO variant, ACS has more extensive population diversity and better problem-solving ability [25]. ACS has been widely used in various discrete combinatorial optimization problems, such as the traveling salesman problem (TSP) [25], supply chain management [26], cloud computing resource scheduling [27], [28], aircraft arrival sequencing and scheduling problem [29], workflow scheduling [30], and the new energy vehicle dispatch problem [31]. Based on the above, the ACS has enormous potential for solving the discrete BAP efficiently.

However, as the scale of the BAP increases, the computational burden of ACS for solving BAP also increases significantly. Receding horizon control (RHC) is a strategy to assist decision feedback, in which a large-scale problem is divided into smaller ones to reduce the computational burden and improve computational efficiency [32]. As a divide-and-conquer strategy, RHC has been applied to solve or help solve various optimization problems. In [33], the RHC strategy was integrated into a GA to solve the multiairport capacity management problem. Zhan *et al.* [29] applied the RHC to divide the arrival aircraft into different receding horizon regions to help the ACS schedule the sequencing more efficiently. Inspired by the application of RHC in the combinatorial optimization problem, we introduce the RHC strategy to improve ACS for solving the BAP. Note that, to avoid the situation in which some aircraft will be missed when the receding horizon moves processes, the basic information of these aircraft is modified in [29]. For example, the predicted arrival time of those aircraft

that are not scheduled in the current horizon will be modified to ensure that these aircraft will be considered in the following horizons. However, when we consider multi-berth BAP in this paper, we think it is not a good idea to modify the vessels' basic information (e.g., the predicted arrival time). Changing the predicted arrival time of a vessel may make this vessel unsuitable for some berths. Therefore, different from the modification of the predicted arrival time used in [29], an improved RHC strategy is proposed to ensure that no vessels are missed when the receding horizon moves forward.

In view of the characteristics of BAP, which is multi-berth, real-time, and multi-constrained, we propose three improvement strategies for ACS to efficiently solve the BAP, which are outlined as follows:

1) We propose an adaptive ACS (AACS) to solve the BAP for higher solution accuracy. As the multi-berth feature is considered in our BAP model, the AACS adopts an adaptive heuristics information (AHI) mechanism to update the heuristics values based on feedback from berth information and vessel arrival time.
2) To meet the real-time requirements of BAP, a variable-range RHC (vRHC) strategy is proposed to cooperate with AACS to address the BAP as several smaller sub-BAPs to enhance computational efficiency. The time intervals of allocation windows in receding horizons are adjusted according to the arrival time of vessels so that every vessel can be considered in a certain horizon.
3) A partial solution memory (PSM) mechanism is proposed for AACS to accelerate the convergence speed in the optimization process of the next receding horizon. The memory mechanism adjusts the initial individuals according to the best solution of the previous receding horizon.

The rest of this paper is organized as follows. Section II provides a detailed description of the BAP and gives the mathematical definition of the BAP. The proposed AACS based on AHI, vRHC, and PSM is then given in Section III. Furthermore, Section IV introduces how to generate experimental instances, which is followed by the analysis of experimental results. Finally, Section V concludes this paper and provides a discussion on future works.

TABLE I
LIST OF MATHEMATICAL SYMBOLS

Symbol	Description
	Sets
$B = \{1, \ldots, m\}$	The set of discrete berths, m is number of berths
$V = \{1, \ldots, n\}$	The set of vessels in BAP, n is number of vessels
$V_K = \{1, \ldots, nk\}$	The subset of V ($V_K \subset V$), containing vessels in the Kth sub-BAP, nk is number of vessels in Kth sub-BAP
πV	The set of edges on the optimal solution of the BAP
πV_K	The set of edges on the optimal solution of the Kth sub-BAP
πV_{PSM}	The set of edges whose pheromone are enhances by PSM mechanism
N	The set of positive integers
Ω	The best solution of the BAP, containing ABT_v and ABB_v of all vessels in the BAP, $v \in V$
	Known variables
PAT_v	$v \in V$, the predicted arrival time of vessel v
LUT_v	$v \in V$, the loading and unloading time of vessel v
WT_v	$v \in V$, the waiting time of vessel v
	Auxiliary parameters
x_{vb}	$v \in V$, $b \in B$, equal to 1 if vessel v is assigned for service at berth b; 0 otherwise;
ABT_v	$v \in V$, the assigned berthing time of vessel v in berth b
ABB_v	$v \in V$, the assigned berthing berth of vessel v
BST_b	$b \in B$, the stopped operating time of berth b, the initial value is set to 0
S_b^v	$v \in V$, $b \in B$, $S_b^v \in N$, the serial number of berthing ascending sequence of vessel v in berth b
	vRHC parameters
N_{RHC}	The time intervals' number of the receding horizon, set $N_{\mathrm{RHC}} = 4$
K	$K \in N$, the Kth receding horizon
W_t^K	$1 \le t \le N_{\mathrm{RHC}}$, $t \in N$, $K \in N$, the time interval of the t-th allocation window in the Kth receding horizon
W_0	The initial width of the allocation window, $W_t^1 = W_0$ $(1 \le t \le N_{\mathrm{RHC}})$, and $W_t^K = W_0$ $(1 < t \le N_{\mathrm{RHC}})$
T_K	The starting time of the Kth receding horizon
	Objective parameter
TWT	The total vessels' waiting time

II. PROBLEM DESCRIPTION AND FORMULATION

A. Problem Description

In this paper, the BAP model is a single-user model. The single-user refers to a shipping company that schedules all the vessels. Therefore, the different waiting times for each vessel can be tolerated. A vessel with an earlier arrival time is sometimes the last vessel to berth. This situation does not affect the satisfaction of the vessels because they all belong to the same company. Table I summarizes the notations and definitions of the BAP model including some sets, known variables, auxiliary parameters, vRHC parameters, and objective parameters. Some of the important symbols of the BAP model are detailed described as follows.

Firstly, the set $V = \{1, \ldots, n\}$ denotes a set of vessels arriving at the container terminal and n denotes the number of vessels. All vessels belong to the same shipping company. Let $B = \{1, \ldots, m\}$ be a set of available discrete berths, where m denotes the number of berths that this shipping company can use on the terminal (i.e., the number of berths offered to the shipping company by the terminal manager). In addition, it is assumed that all berths meet the draft and the length of all vessels (i.e., any vessel can be served at any provided berth).

Secondly, the predicted arrival time PAT_v is the accurate arrival port time for vessel v provided by the shipping company. Uncertain circumstances, such as the vessel being delayed to the terminal due to adverse weather conditions or other factors, are not considered in this model. The loading and unloading time LUT_v is a known condition that is provided by the shipping company. The number of quay cranes at every berth is fixed, and quay cranes always maintain a stable handling rate for vessel service. The handling rate is the same at each berth. The number of cargoes to be loaded and unloaded for each vessel is known, it is easy to calculate the vessel loading and unloading time LUT_v at the berth according to these known conditions. Moreover, the BAP model does not

consider the distance from berths to yards, which is another factor of the vessel's processing time.

Thirdly, some auxiliary parameters are described as follows. When vessel v arrives at the terminal, if its assigned berthing berth ABB_v is now serving another vessel, vessel v will wait at the waiting area (such as vessel 3 and vessel 4 in Fig. S.1 in the supplementary material). The waiting time of vessel v in the waiting area will be added to the total vessels' waiting time *TWT*. In another situation, when vessel v arrives at the container terminal and the assigned berthing berth ABB_v is idle, the assigned berthing time (*ABT*) of vessel v is $ABT_v = PAT_v$. When the loading and unloading service is finished, the vessel should leave the berth immediately. Fragments of time (which may be caused by the vessel that is towed from the waiting area to its assigned berth, time of quay crane startup, or operating time of vessel departure from berth) are ignored in this paper. Also, we define the berth stopped operating time (BST_b) and the berthing sequence (S_b^v) as auxiliary variables. BST_b denotes the stopped operating time of berth b, and its initial value is set to 0. S_b^v denotes the berthing sequence index of vessel v on berth b. For instance, $S_{b1}^{v1} = 1$ means that vessel $v1$ is the first vessel at the berth $b1$ sequence. $S_{b2}^{v3} - S_{b2}^{v2} = 1$ means that the next berthing vessel of vessel $v2$ is vessel $v3$ at berth $b2$.

B. Objective Formulation

The objective of the BAP model is to find a berth schedule that minimizes the *TWT*. The berth schedule includes ABT_v and ABB_v of all vessels that need to be served in BAP. A minimum *TWT* implies a minimum total operation time of all vessels. The metric of the *TWT* emphasizes the best utilization of berths and the maximum throughput of the container terminal.

The BAP model can be formulated as:

$$\begin{aligned} min\ f &= min\ TWT \\ &= min\ \sum\nolimits_{b\in B}\sum\nolimits_{v\in V}(ABT_v - PAT_v)\cdot x_{vb} \end{aligned} \tag{1}$$

$$\text{subject to}: \quad \sum\nolimits_{b\in B} x_{vb} = 1,\ \forall v \in V \tag{2}$$

$$\sum\nolimits_{v\in V} x_{vb} \le 1,\ \forall b \in B \tag{3}$$

$$ABT_v \ge PAT_v,\ \forall v \in V \tag{4}$$

$$x_{vb} \in \{0,1\},\ \forall v \in V, \forall b \in B \tag{5}$$

Equation (1) is the objective function of the BAP that minimizes the total vessels' waiting time *TWT*. Constraint (2) indicates that each vessel must be served only once at one of any berths. Constraint (3) ensures that only one vessel can be served at a given berth at a time. Constraint (4) guarantees that vessels are served after their arrivals. Constraint (5) shows decision variables x_{vb} of the BAP model. The assigned berthing time ABT_v of Equation (1) for each vessel to be served at berth b ($b \in B$) is calculated by

$$ABT_v = \begin{cases} PAT_v, & \text{if } S_{ABB_v}^v = 1 \\ max\ \left\{PAT_v, BST_{ABB_{v'}}\right\}, & \text{otherwise} \end{cases} \tag{6}$$

where vessels v and v' satisfy:

$$ABB_{v'} = ABB_v = b,\ v' \in V,\quad v \in V,\quad b \in B \tag{7}$$

$$\left(S_{ABB_v}^v\right) - \left(S_{ABB_{v'}}^{v'}\right) = 1,\ v' \in V,\quad v \in V \tag{8}$$

$$BST_{ABB_{v'}} = ABT_{v'} + LUT_{v'},\ v' \in V \tag{9}$$

$$0 < S_b^v \le m \quad \forall v \in V,\ \forall b \in B \tag{10}$$

$$ABB_v \in B \quad \forall v \in V \tag{11}$$

Equation (6) calculates the value of assigned berthing time ABT_v. When vessel v is the first moored at the same berth b, $ABT_v = PAT_v$. Otherwise, the value of ABT_v is the maximum value between PAT_v and $BST_{ABB_{v'}}$. Constraint (7) indicates that vessel v' and vessel v choose the same berth b. Constraint (8) guarantees that vessel v' is the previous vessel that is berthed at berth b before vessel v. Constraint (9) calculates the BST_{ABB_v}, of vessel v' choosing berth $ABB_{v'}$, which is a recursive formula. The value of $BST_{ABB_{v'}}$ is related to all vessels that are moored at berth $ABB_{v'}$ before vessel v. In general, when vessel v arrives at the port and the selected berth $ABB_{v'}$ is idle, vessel v will be moored immediately. If vessel v' is operating at berth $ABB_{v'}$, vessel v needs to wait until the loading and unloading operation of vessel v' is completed. Constraints (10) and (11) define the respective domains of the variables in the BAP model.

III. The AACS Algorithm

This section introduces the proposed AACS algorithm with the AHI mechanism, the vRHC strategy, and the PSM mechanism to solve the BAP.

A. vRHC Strategy

The proposed vRHC is a divide-and-conquer strategy. The BAP is divided into several smaller sub-BAPs by receding horizon. In the Kth receding horizon, the AACS is used to solve the sub-BAP to obtain a partial solution. Then, the optimization process repeats for the $(K+1)$th sub-BAP until a complete solution for the BAP is built. The relevant parameters of vRHC are defined in Table I.

The first step of the vRHC strategy is to find all vessels in the current sub-BAP (i.e., the Kth receding horizon). In the Kth sub-BAP, the vessel set V_K includes all vessels whose PAT_v is within the Kth receding horizon. Then, the AACS algorithm is executed for the Kth sub-BAP to obtain the ABT_v and ABB_v of the vessels in the set V_K. The second step of the vRHC strategy is to save the solution of the current sub-BAP into the best solution Ω to complete the solution. Each receding horizon is divided into N_{RHC} time intervals, which are also named allocation windows. Note that although all the vessels in all the N_{RHC} allocation windows of the Kth sub-BAP are considered in the AACS scheduling, only the ABT_v and ABB_v of the vessels whose ABT_v are within the first allocation window can be saved into the best solution Ω. These saved vessels are regarded as having been scheduled and will not be added to the next receding horizon. The third step of the vRHC strategy is to find the range of the next sub-BAP (i.e., the $(K+1)$th receding horizon). The receding horizon moves one

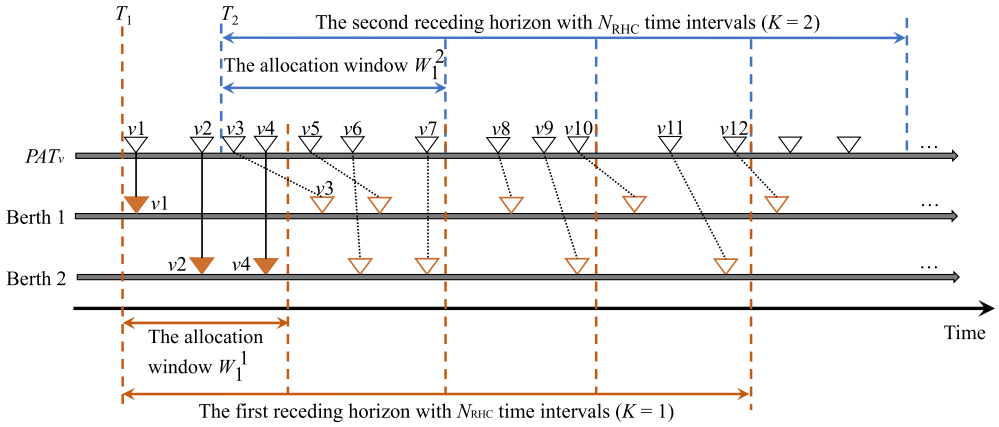

Fig. 2. Example of variable domain RHC for BAP ($N_{\mathrm{RHC}} = 4$).

allocation window every time. The N_{RHC} allocation windows initially have a fixed width W_0 in the first receding horizon (i.e., the first sub-BAP). However, some vessels may have their PAT_υ within the first allocation window, but their ABT_υ are not within the first allocation window. In such cases, these vessels cannot be saved into the best solution Ω and they should be included in the next sub-BAP. However, as pointed out in the Introduction, modifying the PAT_υ of these vessels may make them unsuitable for some berths. Therefore, unlike [29], to make these vessels available in the next receding horizon, the width of the first allocation window W_1^K of the following receding horizon (i.e., $K \geq 2$) is variable. That is, the starting time of the first allocation window W_1^K ($K \geq 2$) should be pushed forward to include those vessels that are not scheduled in the current receding horizon. Specifically, if some vessels whose PAT_υ are within the first allocation window of the Kth receding horizon but cannot be scheduled because their ABT_υ are not within the first allocation window, the starting time of the first allocation window of the $(K+1)$th receding horizon T_{K+1} is set as the smallest PAT of these vessels. Otherwise, the width of the first allocation window of the $(K+1)$th receding horizon remains unchanged. Note that the widths of the other $N_{\mathrm{RHC}} - 1$ allocation windows remain the same as W_0 when the receding horizon moves forward. Then, the optimization process repeats for the $(K+1)$th sub-BAP until all vessels in the BAP are allocated.

An example of vRHC is given in Fig. 2. Given a sequence containing PAT of all vessels, the vRHC strategy begins the scheduling of the first sub-BAP as the first receding horizon. We set $T_1 = PAT_{\upsilon 1}$, $K = 1$, and $N_{\mathrm{RHC}} = 4$. In the first receding horizon, all allocation windows have the same width $W_{\mathrm{t}}^1 = W_0$. According to the PAT_υ of the vessels, there are 12 vessels (i.e., $\upsilon 1, \upsilon 2, \ldots, \upsilon 12$ in Fig. 2) in the first sub-BAP. After the AACS algorithm is executed on the first receding horizon, the berthing plan is shown in Fig. 2, where $\upsilon 1$, $\upsilon 3$, $\upsilon 5$, $\upsilon 8$, $\upsilon 10$, and $\upsilon 12$ are assigned to berth 1 while $\upsilon 2$, $\upsilon 4$, $\upsilon 6$, $\upsilon 7$, $\upsilon 9$, and $\upsilon 11$ are assigned to berth 2. Since $ABT_{\upsilon 1}$, $ABT_{\upsilon 2}$, and $ABT_{\upsilon 4}$ are in the first allocation window, the berthing plan of $\upsilon 1$, $\upsilon 2$, and $\upsilon 4$ is saved into Ω. The saved vessels are marked as solid triangles in Fig. 2. Since vessel $\upsilon 3$ cannot be saved in the first ($K = 1$) receding horizon, it will be added to the second receding horizon. The second receding horizon is marked in blue in Fig. 2. To make $\upsilon 3$ available in the second receding horizon, the starting time of the second allocation window is set as $T_2 = PAT_{\upsilon 3}$ (with $PAT_{\upsilon 3}$ being the smallest among the remaining unallocated vessels). Therefore, the width of the first allocation window of the second receding horizon is changed (i.e., $W_1^2 \neq W_0$).

Obviously, some vessels (e.g., $\upsilon 3$, $\upsilon 5$, $\upsilon 6, \ldots,$ and $\upsilon 12$ in Fig. 2) are included in two adjacent sub-BAPs. Therefore, the allocation plan of these vessels, which has been obtained in the previous sub-BAP, can be utilized to accelerate the convergence speed of AACS in solving the current sub-BAP. Therefore, we propose the PSM mechanism by which the partial best allocation plan of the previous sub-BAP is reserved. The partial best allocation plan is a set of arcs that is also contained in the current sub-BAP. The PSM mechanism adds the pheromone to the arcs of the current sub-BAP, which is introduced in Section III-B-1).

B. AACS for Solving Sub-BAP

The proposed AACS is used to find the optimal berthing plan of the vessels in the Kth sub-BAP. The objective of the AACS is to minimize the TWT by the optimal berthing plan. The objective function for the Kth sub-BAP is given as

$$min\ TWT_K = \sum_{b=1}^{m} \sum_{\upsilon=1}^{nk} (ABT_\upsilon - PAT_\upsilon) \cdot x_{\upsilon b} \quad (12)$$

where TWT_K denotes the total vessels' waiting time in the Kth sub-BAP, and nk is the number of vessels in the Kth sub-BAP. The pseudo-code of the solution process of the Kth sub-BAP is given in **Algorithm 1**. Detail information is described below.

1) Initialization and Updating Rules of Pheromone: The pheromone $\tau(i, j)$ is a desirability measure of edges visited by ants. In the directed graph of the BAP, the pheromone on every directed edge between two vessels is not the same (i.e., $\tau(i, j) \neq \tau(j, i), \forall i \in V_K, \forall j \in V_K$). The design of the initial pheromone in the AACS algorithm includes two steps: pheromone initial value assignment based on FCFS and pheromone enhancement by the PSM mechanism, which are shown in step 3 and step 5 of Algorithm 1, respectively.

In the sub-BAP, each edge has an initial pheromone $\tau_0(i, j)$ ($\forall i \in V_K, \forall j \in V_K$). In step 3 of Algorithm 1, the FCFS algorithm is used to allocate berths and obtain a fitness

Algorithm 1 Construct Solution of the Kth Sub-BAP

1. Initialize V_K and the number of vessels nk via vRHC;
2. Initialize the number of ants NA, maximal generation NG;
3. Initialize pheromone;
4. **If** $K \geq 2$
5. Enhance the pheromone by the PSM mechanism via Eq. (13);
6. **End if**
7. **For** each gen in $\{1, \ldots, NG\}$ **Do**
8. Randomly select the 1st of vessel for each ant;
9. **For** each j in $\{1, \ldots, NA\}$ **Do**
10. **For** each $\bar{v}$ in $\{1, \ldots, nk\}$ **Do**
11. Construct the whole solution for ant j according to state transition rule;
// The AHI mechanism is used in this step.
12. Perform the local pheromone updating;
13. **End for**
14. **End for**
15. Calculate the fitness of every ant;
16. Update the historical best solution $gbest$;
// The best solution includes $\pi gbest$, $ABB_{\bar{v}}$ and $ABT_{\bar{v}}$ of all vessels.
17. Perform the global pheromone updating;
18. **End for**
19. **Return** $gbest$.

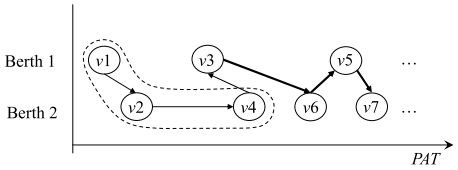

Fig. 3. Partial best path of the 1st receding horizon in Fig. 2.

value f_{FCFS} of the objective function (12). Then the initial pheromone of all edges is set to $\tau_0(i, j) = (nk \times f_{\mathrm{FCFS}})^{-1}$.

To accelerate the convergence speed of AACS in the Kth ($K \geq 2$) sub-BAP, the PSM mechanism is used to enhance the pheromone on the best path of the $(K-1)$th receding horizon. In step 5 of Algorithm 1, the pheromone is enhanced as

$$\tau_0(i, j) = (1 + \rho) \cdot \tau_0(i, j), \quad (i, j) \in \pi V_{\mathrm{PSM}} \tag{13}$$

where πV_{PSM} is a set of unsaved vessel pairs in the best path in the $(K-1)$th receding horizon, and ρ ($0 < \rho < 1$) is the pheromone enhancement parameter. According to the example in Fig. 2, the allocation plan of $v1$, $v2$, and $v4$ has been saved into Ω and is indicated by grouping them in the dotted line circle, as shown in Fig. 3. The unsaved vessel pairs of the best path in the first receding horizon are $\pi V_{\mathrm{PSM}} = \{(3, 6), (6, 5), (5, 7), \ldots\}$, as shown in Fig. 3. Then, we add pheromones to these arcs in the second sub-BAP.

During the AACS process, a local pheromone updating rule and a global pheromone updating rule are performed to update the pheromone. These two pheromone updating rules are used to guide the search direction of ants. The local pheromone update is performed to reduce the probability of each ant choosing the same path to ensure the diversity of the population. The local pheromone updating rule is operated on each vessel pair in the completed scheduled solution as

$$\tau(i, j) = (1 - \rho) \cdot \tau(i, j) + \rho \cdot \tau_0(i, j) \tag{14}$$

where ρ is set as the same value in the PSM mechanism.

The global pheromone updating rule is used to accelerate the speed of convergence by guiding the search in a more promising direction of ants. The global pheromone updating is performed after all ants have built their solutions in the current generation. The pheromone on each vessel pair is increased as

$$\tau(i, j) = (1 - \varepsilon) \cdot \tau(i, j) + \varepsilon \cdot \tau', \quad \text{if } (i, j) \in \pi gbest \tag{15}$$

where $\tau' = 1/f_{\pi gbest}$. $\pi gbest$ is a set of the vessel pairs of best paths in the current generation. Then, $f_{\pi gbest}$ is the fitness of $\pi gbest$ which is calculated by Eq. (12). ε is the pheromone enhancement parameter.

2) State Transition Rule: In AACS, the first vessel of each ant is randomly selected before solution construction. Then, the state transition rule is used to guide the iterative process of the ants by assigning the following vessels to ants one by one. When an ant completes the allocation of vessel i, it then chooses the next vessel j according to the state transition rule given by

$$j = \begin{cases} arg\ max_{j \in Ai} \left\{ \tau(i, j) \cdot \eta(i, j)^{\beta} \right\}, & \text{if } q < q_0 \\ J, & \text{otherwise} \end{cases} \tag{16}$$

where Ai is a set of unvisited vessels reachable by vessel i. q_0 is a parameter in interval [0,1] used to control the exploitation and exploration behaviors of the ant. A number $q \in [0,1]$ is randomly generated. If $q < q_0$, then the ant chooses the next vessel whose product of pheromone τ and heuristic information η is maximal, measured by $\tau(i, j) \cdot \eta(i, j)^{\beta}$, where β ($\beta > 0$) is a parameter for exploitation. The heuristic information η is adaptively controlled by the AHI mechanism in this paper as introduced in Section III-B-3). Otherwise, the next vessel j can be determined according to a random variable J, which is selected from Ai according to a probability distribution selection as

$$P(i, j) = \begin{cases} \dfrac{\tau(i, j) \cdot \eta(i, j)^{\beta}}{\sum_{u \in Ai} \tau(i, u) \cdot \eta(i, u)^{\beta}}, & \text{if } j \in Ai \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

Probability P is used to ensure better exploration ability when the solution is feasible.

3) Adaptive Heuristic Information: Heuristic information is an important component in the state transition rule of the AACS algorithm. The heuristic information $\eta(i, j)$ is the inverse of the distance between two nodes, and the heuristic information $\eta(i, j)$ in BAP is designed as

$$\eta(i, j) = \frac{1}{dis(i, j)} \tag{18}$$

where $dis(i, j)$ is the "distance" between vessel i and vessel j, and the "distance" is the time interval between two vessels. Similar to the pheromone, the heuristic information on every directed edge between two vessels is not the same (i.e., $\eta(i, j) \neq \eta(j, i)$). In the AACS algorithm, the AHI mechanism is designed to ensure a reasonable time interval (i.e., heuristic information $\eta(i, j)$) between different vessels and berths.

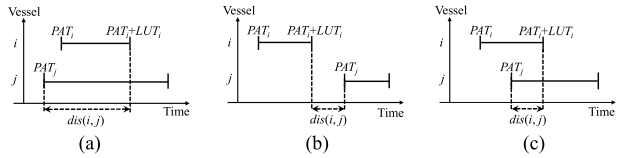

Fig. 4. Adjacent vessels i and j berthing at the same berth. (a) vessel i had not yet arrived when vessel j arrived. (b) the berth is idle when vessel j arrives. (c) vessel i was being served in the berth when vessel j arrived.

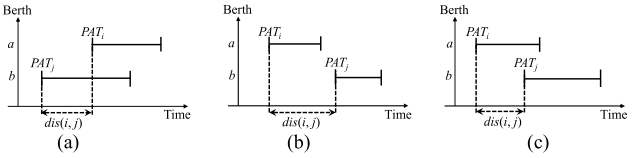

Fig. 5. Adjacent vessels i and j berthing at different berths. (a) vessel i had not yet arrived at berth a when vessel j arrived. (b) vessel i had left berth a when vessel j arrived. (c) vessel i was being served at berth a when vessel j arrived.

According to the characteristics of the BAP, three factors affect the "distance" between two vessels: *PAT*, *LUT*, and *ABB*. A sequence number $sort(i)$ is assigned to each vessel i after organizing in ascending order vessels according to their *PAT*s. As illustrated in Fig. 2, $sort(v1) = 1$, $sort(v2) = 2$. Then the adaptive heuristic information is given in Eq. (19), where $|sort(i) - sort(j)| = 1$ indicates that the *PAT*s of vessel i and vessel j are adjacent (vessel i and vessel j are time-oriented nearest neighbors). Here if vessel i is the previous vessel of vessel j, vessel i and vessel j are called adjacent vessels. The heuristic information considers four kinds of vessel pairs: adjacent vessels berthing at the same berth, adjacent vessels berthing at different berths, not-adjacent vessels berthing at the same berth, and not-adjacent vessels berthing at different berths.

a) Adjacent vessels berthing at the same berth: When vessel i leaves berth ABB_i after finishing the loading and unloading operations, vessel j can be moored at berth ABB_i. Therefore, the loading and unloading time of vessel i must be considered. The order of arrival of the two vessels is also an influencing factor. As shown in Fig. 4(a), vessel j arrives before vessel i, and vessel i must wait until vessel j leaves. In Fig. 4(b), vessel i leaves the berth, vessel j has not arrived, and the berth is idle. In Fig. 4(c), vessel i is operates when vessel j arrives, and vessel j can be berthed if vessel i leaves (19), as shown at the bottom of the page.

b) Adjacent vessels berthing at different berths: At different berths, vessel i and vessel j do not affect each other. The "distance" $dis(i, j)$ is designed as the idle time of the berth. According to the arrival time of the two vessels, it can be divided into three situations. If $PAT_i \geq PAT_j$, $dis(i, j)$ is designed as the idle time of ABB_i. Otherwise, $dis(i, j)$ is designed as the idle time of ABB_j. As illustrated in Fig. 5(a), the selected berth is ABB_i which is the idle time of berth a. Fig. 5(b) and Fig. 5(c) show the idle time of the berth selected by vessel j.

c) Non-adjacent vessels berthing at the same berth or at different berths: In Eq. (19), *VI* is a set of vessels that arrive at the port between the arrivals of vessel i and vessel j. Since these vessels in *VI* may all need to wait until vessel i and vessel j complete operation and depart, the operating time of these vessels is added in $dis(i, j)$ without loss of generality.

C. Complete AACS Algorithm With vRHC Strategy and PSM Mechanism

The pseudo-code of the complete AACS algorithm is shown in **Algorithm 2**.

Algorithm 2 Complete AACS

1. Initialize N_{RHC}, W_1^1, and W_0;
2. $K = 1$; $T_1 = PAT_{v1}$;
3. **While**$(V \neq \emptyset)$ **Do**
4. $\pi V_K \leftarrow \emptyset$; $V_K \leftarrow \emptyset$;
5. $V_K \leftarrow \arg_v \{T_K \leq PAT_v < T_K + W_1^K + (N_{\mathrm{RHC}} - 1) \times W_0\}$; // $v \in V$
6. Construct solution of the Kth sub-BAP via **Algorithm 1**;
7. $\pi V_{\mathrm{PSM}} \leftarrow \emptyset$;
8. **For** each $\bar{v}$ in V_K **Do** //$\bar{v} \in V_K$
9. **If** $ABT_{\bar{v}} \leq T_K + W_1^K$ **Then**
10. Store $ABB_{\bar{v}}$ and $ABT_{\bar{v}}$ to Ω;
11. $V \leftarrow V - \{\bar{v}\}$;
12. $V_K \leftarrow V_K - \{\bar{v}\}$;
13. Store path of $\bar{v}$ to πV;
14. Delete path of $\bar{v}$ from πV_K;
15. **End if**
16. **End for**
17. **If** $\min(PAT_{\forall v''}) > W_1^K + T_K$ **Then** // $v'' \in V_K$
18. $T_{K+1} \leftarrow W_1^K + T_K$;
19. $W_1^{K+1} \leftarrow W_0$;
20. **Else**
21. $W_1^{K+1} \leftarrow W_1^K + T_K - \min\left(PAT_{\forall v''}\right) + W_0$;
22. $T_{K+1} \leftarrow \min\left(PAT_{v''}\right)$;
23. **End if**
24. $\pi V_{\mathrm{PSM}} \leftarrow \pi V_K$;
25. $K \leftarrow K+1$;
26. **End while**
27. **Return** Ω and πV.

In the first receding horizon (i.e., $K = 1$), the time interval of each allocation window is the same. We therefore set $W_1^1 = W_0$ in step 1. The W_0 in every receding horizon is the same, and we simply use W_0 to represent all the $W_t^K (t > 1)$ herein and in Algorithm 2. The W_1^K $(K \geq 2)$ can be varied in different receding horizons due to the variable range characteristic of vRHC when the receding horizon moves forward. Then, the

$$dis(i, j) = \begin{cases} \left|PAT_i + LUT_i - PAT_j\right|, & \text{if } |sort(i) - sort(j)| = 1 \text{ and } ABB_i = ABB_j \\ \left|PAT_i - PAT_j\right|, & \text{if } |sort(i) - sort(j)| = 1 \text{ and } ABB_i \neq ABB_j \\ \left|PAT_i + LUT_i - PAT_j\right| + \sum_{d \in VI} LUT_d, & \text{if } |sort(i) - sort(j)| \neq 1 \text{ and } ABB_i = ABB_j \\ \left|PAT_i - PAT_j\right| + \sum_{d \in VI} LUT_d, & \text{if } |sort(i) - sort(j)| \neq 1 \text{ and } ABB_i \neq ABB_j \end{cases} \tag{19}$$

procedure enters the loop (step 3 - step 26) until $V \neq \emptyset$, where V is reduced through an iterative process. In step 4, πV_K and V_K is reinitialized in every sub-BAP. Next, in step 5, the set V_K of the Kth receding horizon is determined. The solution of the Kth sub-BAP is constructed via Algorithm 2 in step 6. Then the loop (step 8 - step 16) is used to store the solution of vessels in the first allocation window W_1^K. In step 17, the changing criterion of the time interval W_1^K is determined. In this step, V_K is a set of unsaved vessels in the Kth receding horizon. If the changing criterion of the time interval is satisfied, the receding horizon moves forward directly with the same allocation windows (step 18 and step 19). Otherwise, the allocation window W_1^{K+1} in the $(K+1)$th receding horizon is determined by the PAT of the earliest vessel in V_K (step 21 and step 22). In step 24, πV_{PSM} is determined by the Kth receding horizon. πV_{PSM} is a set of leftover vessels in the Kth receding horizon, which is used in the $(K+1)$th receding horizon. The procedure is terminated when all vessels find their ABB_v and ABT_v.

IV. Experimental Studies

A. Parameter Configurations

Experimental tests are conducted in this section to evaluate the effectiveness and efficiency of the proposed AACS. These numerical comparison tests are implemented in Visual Studio C++ and run on a PC with Dell Intel(R) Core-i7 and 8.0 GB RAM.

References in scheduling application literature [6], [19] and actual data [2], test cases with integer numerical data were generated by different distributions. The exponential distribution is used to assign the predicted arrival time of the vessel, which was marked as $PAT_v = E(60)$, where randomly generated data conform to an exponential distribution (E -) with a mean value of 60 minutes. To facilitate the algorithm expression, the test cases are normalized, that is, the predicted arrival time is represented by an increasing integer. For example, in Test Case 1, the arrival time of the first vessel is standardized to 0, and the arrival time of other vessels is the incremental data plus the time interval. For example, if the arrival time of vessel 1 is 8:00 am, then vessel 2 arrives at 9:25 am after an interval of 85 minutes. The loading and unloading time of the vessel occupying the berth was generated as $LUT_v = U(120, 300)$ or $U(45, 240)$, meeting the uniform distribution (U -) between 2 hours and 5 hours or 45 minutes and 4 hours. Different value ranges are used to diversify cases and better verify each algorithm.

By adjusting the number of vessels, 20 test cases are created using the generated data, from 10 vessels to 120 vessels. In these test cases, the AACS is compared with FCFS, the adaptive evolutionary algorithm (AEA) [19], and ACO [24], which have proven to be suitable for solving the discrete BAP model with minimum delay time. FCFS is a fast method, in which the vessel with earlier arrival time will give priority to the earlier free berth. AEA is a well-performing algorithm for BAP with adaptive parameter control mechanism. Moreover, the ACO is compared because the ACS is a variant of ACO. The AACS applies the ACS algorithm to solve the BAP with the AHI mechanism, the vRHC strategy, and the PSM mechanism. The pseudo-code of FCFS and AEA can be found in [19].

TABLE II
Parameter Configurations for the AACS Algorithm

ACS parameters						vRHC parameters	
NP	NG	q_0	β	ε	ρ	N_{RHC}	W_0
$7 \times nk$	$3 \times nk$	0.1	1.0	0.1	0.9	4	$m \times PAT_{Vn}$ / n

The parameter configurations for AACS are shown in Table II. The ACS-related parameters are the population size $NP = 7 \times nk$ and maximal generation number $NG = 3 \times nk$. They are set to seven times and three times the number of vessels nk in the current receding horizon. nk is changed according to vRHC-related parameters. The other ACS-related parameters are $q_0 = 0.1$, $\beta = 1.0$, $\varepsilon = 0.1$, and $\rho = 0.9$. The vRHC-related parameters are $N_{\text{RHC}} = 4$, $W_0 = m \times PAT_{Vn}/n$. m is the number of berths in test case, and n is the number of vessels in test case. Vn is the last arrived vessel in test case. Since the number of vessels and their arrival time are different in each test case, it is more reasonable to divide the allocation window W_0 according to the arrival time of the last vessel Vn. In addition, other parameter configurations are based on empirical studies presented in Sections IV-D and E.

B. Test Case 1:20-Vessel

Table III gives an example with 20 vessels in Test Case 1, which is generated by the method mentioned in Sections IV-A, and $LUT_v = U(120, 300)$. The first three columns of Table II are vessel v, PAT_v and LUT_v. These vessels are numbered in order of PAT_v. The berth schemes of FCFS, AEA, ACO, and AACS are presented, which include the berthing sequence of every vessel, ABB_v, ABT_v. The vessels are sorted by their ABT_v values. For the three evolutionary algorithms, the berth schemes and TWT values are the best solutions among 30 independent runs. The average TWT values of the solutions obtained by the 30 runs are given in the last row.

As shown in Table III, compared with other evolutionary algorithms, the FCFS algorithm has the worst results. Sorting in chronological order is the most direct, simple, and time-efficient method, but it often fails to achieve a good arrangement in the global view. The same number of vessels are equally allocated to each berth one by one in the FCFS scheme. In the three evolutionary algorithms, the number of vessels assigned to each berth is different. The table lists the best allocation schemes obtained by the AEA, ACO, and AACS among their 30 runs, with each single run of each algorithm under the same number of generations. The optimal solution of the AEA algorithm is worse than that of the ACO algorithm and AACS algorithm. Conversely, AACS obtains the best results on TWT and ensures stability with the smallest mean TWT and the smallest worst TWT among their 30 runs. Moreover, the mean and standard deviation show that AACS obtains a better solution than AEA and ACO, whereas AACS obtains this best solution with the "Best Ratio" of 60%. A deep observation on the waiting time of each vessel shows that there are two vessels that do not have to wait in FCFS (i.e., the WT_v

TABLE III
EXPERIMENTAL RESULT COMPARISONS IN TEST CASE 1 WITH 20 VESSELS

Data			FCFS				AEA				ACO				AACS			
v	PAT_v	LUT_v	v	ABB_v	ABT_v	WT_v	v	ABB_v	ABT_v	WT_v	v	ABB_v	ABT_v	WT_v	v	ABB_v	ABT_v	WT_v
1	0	133	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
2	83	128	2	2	83	0	2	2	83	0	3	2	94	0	2	2	83	0
3	94	144	3	1	133	39	3	1	133	39	2	1	133	50	3	1	133	39
4	134	262	4	2	211	77	4	2	211	77	6	2	238	23	6	2	215	0
5	206	278	5	1	277	71	6	1	277	62	5	2	380	174	7	1	277	53
6	215	142	6	2	473	258	7	1	419	195	7	1	261	37	4	2	357	223
7	224	142	7	1	555	331	8	2	473	179	8	1	403	109	8	1	419	125
8	294	232	8	2	615	321	10	1	561	31	11	1	635	96	11	2	619	80
9	327	284	9	1	697	370	11	1	685	146	10	2	658	128	10	1	651	121
10	530	124	10	2	847	317	5	2	705	499	14	2	782	27	14	2	755	0
11	539	128	11	1	971	432	9	1	813	486	15	1	765	0	15	1	775	10
12	642	284	12	2	981	339	14	2	983	228	16	2	913	69	17	2	886	23
13	654	272	13	1	1099	445	19	1	1097	90	17	1	905	42	18	1	931	0
14	755	131	14	2	1265	510	15	2	1114	349	18	1	1157	226	19	1	1080	73
15	765	140	15	1	1371	606	18	1	1229	298	20	2	1172	158	16	2	1138	294
16	844	259	16	2	1396	552	17	2	1254	391	19	1	1306	299	20	1	1212	198
17	863	252	17	1	1511	648	16	1	1378	534	12	1	1438	796	13	2	1397	743
18	931	149	18	2	1655	724	20	2	1506	492	13	1	1722	1068	9	1	1445	1118
19	1007	132	19	1	1763	756	13	1	1637	983	9	2	1405	1078	12	2	1669	1027
20	1014	233	20	2	1804	790	12	2	1739	1097	4	2	1689	1555	5	1	1729	1523
Best *TWT*			7586				6176				5935				**5650**			
Worst *TWT*							6770				7059				**6040**			
Mean of 30 *TWT*							6267.47				6483.73				**5779.37**			
Std.Dev							177.369				466.072				**175.156**			
Best Ratio							40%				23.3%				**60%**			

is 0), and the number of vessels that do not have to wait in AEA, ACO, and AACS are 2, 3, and up to 5, respectively.

The global search capability of the AACS algorithm is also verified in Test Case 1 that the vessels with early *PAT* can be arranged later to reduce the *TWT*. As shown in the AACS scheme columns, vessel 5 with an earlier arrival time is the last vessel to berth, and vessel 4 is the last vessel to berth in the ACO scheme. In addition, the ability of ACS to jump out of the local optimal solution is better than that of ACO. The vRHC strategy further improves the global search capabilities of ACS. Moreover, the vRHC strategy reduces the computational burden by dividing the receding horizon and simultaneously enabling vessels arriving earlier to be scheduled in the latter receding horizon. In contrast, the AEA algorithm initializes the population based on the FCFS policy, and although this step increases the convergence speed, this is a limitation that hinders the diversity of the population.

C. Test Case 2:30-Vessel

We also conduct experiments on Test Case 2 with 30 vessels and the experimental results are listed in Table S.I in the supplemental material due to the page limitation. Unlike Test Case 1, the LUT_v of Test Case 2 is generated by $U(45, 240)$. It can be observed from the table that the time gap between each vessel's berthing time has become larger. The number of vessels has increased, but the value of *TWT* has decreased. In practice, this situation is more common in small bulk cargo terminals.

The results of FCFS, AEA, ACO, and AACS are presented and compared in Test Case 2. Similar to the experimental results in Test Case 1, the FCFS solution keeps the same number of vessels allocated to each position, and still yields the worst result. However, due to the large variance in the berthing times of the vessels, FCFS is no longer alternately allocated to the berth for each vessel. In Test Case 2, AEA obtains better results than ACO and the mean value remains stable. However, the optimal solution is still obtained by AACS. From the mean value of AACS, it can be verified that the algorithm remains stable as the number of vessels increases. A deep observation on the waiting time of each vessel shows that there are five vessels that do not have to wait in ACO and FCFS, and the number is up to seven in AEA and AACS.

D. Analysis of ACS Parameters

The parameters in AACS include the population size *NP*, the maximal generations of the receding horizon *NG*, and the performance-related parameters q_0, ρ, ε, and β. Note that when one parameter is investigated, the others remain the same as in Table II.

The number of ants *NP* and the number of generations *NG* are the basis of the algorithm and directly affect the entire algorithm. We test these two parameters simultaneously in Test Case 2. We set *NP* from $1 \times nk$ to $10 \times nk$ with a step length of $1 \times nk$ and *NG* from $1 \times nk$ to $5 \times nk$ with a step length of $1 \times nk$, where *nk* is the number of vessels in every receding horizon. The mean result of 30 independent runs is plotted in Fig. S.2 in the supplemental material. Fig. S.2(a) and Fig. S.2(b) show the influences of the parameters on the mean *TWT* and the mean CPU time, respectively. As observed from Fig. S.2(a), the smallest value is staying at $NP = 7 \times nk$, $NP = 9 \times nk$, $NG = 3 \times nk$, and $NG = 4 \times nk$. Moreover, as shown in Fig. S.2(b), the larger the values of *NP* and *NG* are, the longer the CPU running time is. Therefore, we set $NP = 7 \times nk$ and $NG = 3 \times nk$.

The next parameter to be tested is q_0. The algorithm converges too slowly when q_0 is set to 0, and a better result cannot be obtained when the number of generations is not enough. Furthermore, setting q_0 to 1 will affect the exploration ability of the algorithm. Therefore, we set q_0 from 0.1 to

TABLE IV
EXPERIMENTAL RESULTS OF AACS ON DIFFERENT STRATEGIES SITUATIONS IN TEST CASE 3 TO TEST CASE 10

Test Case	n	FCFS	AEA			ACO			ACS			AACS-w/o-AHI			AACS-w/o-PSM			AACS		
			Min	Mean	CPU	Min	Mean	CPU	Min	Mean	CPU	Min	Mean	CPU	Min	Mean	CPU	Min	Mean	CPU
3	10	451	**302**	**302**	338	**302**	306	227	**302**	323	334	**302**	470	128	**302**	311	130	**302**	**302**	**127**
4	10	658	**461**	**461**	366	**461**	561	226	**461**	477	336	**461**	489	112	**461**	466	**111**	**461**	**461**	116
5	20	5275	3903	4063	711	4087	4586	984	3901	4165	961	3538	3561	**447**	3546	3817	453	**3525**	**3540**	459
6	20	2402	2104	2160	720	2155	2385	936	2226	2228	981	1964	1972	**495**	**1952**	2063	506	1955	**1959**	501
7	30	1531	1383	1489	920	1739	2624	1346	1853	2047	1420	1493	1523	613	**1331**	1499	**609**	**1331**	**1403**	628
8	30	8027	6483	6845	941	7079	8038	1431	7895	7957	1449	6283	6384	674	**6140**	6913	**671**	6159	**6401**	677
9	40	6602	6201	6410	1215	6794	8758	1942	6452	6726	2006	5508	5513	**933**	5324	6427	942	**5244**	**5442**	952
10	40	10438	8650	9215	1418	9981	10964	1831	9008	10398	1798	7510	7762	959	7533	8270	**925**	**7167**	**7235**	961

CPU time is measured as millisecond

0.9 with a step length of 0.1. The investigation results are plotted in Fig. S.3 in the supplemental material. Herein, both Test Case 1 and Test Case 2 are applied to test the performance of different values of q_0. It can be seen from Fig. S.3 that a better solution can be obtained when the value of q_0 is set as 0.1. Therefore, q_0 is set to 0.1 in this paper. The PSM mechanism is used to accelerate the convergence speed of the optimization process. Therefore, a small value of q_0 is applied to enhance the exploration ability.

Then, parameter β is investigated. This parameter has also been tested on Test Case 1 and Test Case 2, and the values range from 1 to 9. The investigation results are plotted in Fig. S.4 in the supplemental material. It can be observed that when the value of β is 1, the algorithm obtains a better solution. Moreover, the value of *TWT* when $\beta = 0$ is too large to plot within the scale of Fig. S.4, which indicates that the heuristic information plays an important role in AACS algorithm.

Finally, parameter ρ for local updating and parameter ε for global updating are investigated. We set ρ and ε from 0 to 1 with a step length of 0.1. The mean *TWT* results are plotted in Fig. S.5 in the supplemental material. The performance is verified in both Test Case 1 and Test Case 2. Fig. S.5(a) shows the test results of Test Case 1, and Fig. S.5(b) shows the test results of Test Case 2. The poor performance of the algorithm when ε or ρ is set to 0 indicates that the pheromone updating rules play important roles in the AACS algorithm. The results of the mean *TWT* for $\varepsilon = 1.0$ or $\rho = 1.0$ are too poor to apply to the algorithm. Fig. S.5(a) shows that the solution quality is not very sensitive to ε and ρ when they are set from 0.1 to 0.9. Therefore, we set $\varepsilon = 0.1$ and $\rho = 0.9$, where the mean *TWT* value is minimal in Test Case 1 and Test Case 2.

E. Analysis of vRHC Parameters

In this section, the impact of parameters N_{RHC} and W_1^1 on the performance of AACS is investigated. We set parameter N_{RHC} from 1 to 6 with a step length of 1. We set an auxiliary parameter I to adjust the value of W_0 (i.e., $W_0 = I \times m \times PAT_{Vn}/\ n$), where I ranges from 1 to 4 with a step length of 1. Thirty independent runs on Test Case 2 mean *TWT* and mean CPU time are plotted in Fig. S.6 in the supplemental material. As can be observed, the test results vary greatly under different N_{RHC} and W_0, so we choose the setting that takes less time to find the best solution, where $N_{\mathrm{RHC}} = 4$ and $I = 1$. The appropriate allocation window width is set by the actual situation of the test case, $W_0 = m \times PAT_{Vn}/\ n$.

F. Effectiveness of AACS and Its Components on More Test Cases

In order to comprehensively evaluate the effectiveness of the AACS algorithm, more test cases are conducted for experiments and comparisons. In this section, Test Case 3 to Test Case 10 with $LUT_v = U(45, 240)$ are generated to test different algorithms. The numbers of vessels are shown in Table IV as 10, 20, 30, and 40. All test cases in Table IV are satisfied that the number of berths is 2.

Besides the compared FCFS, AEA, and ACO algorithms, different variants of AACS are also compared to investigate the effectiveness of different components in AACS. Especially, among the compared algorithms, the ACS is a special case of AACS, which does not have any strategy (i.e., without the AHI mechanism, the vRHC strategy, or the PSM mechanism). Moreover, the AACS-w/o-AHI is an AACS variant that does not include the AHI mechanism. Similarly, the PSM mechanism is not applied in the AACS-w/o-PSM variant. The minimum solution, the mean *TWT*, and the mean CPU time of each algorithm independently executed 30 times are expressed as Min, Mean, and CPU in Table IV. The values of the mean CPU time and the mean *TWT* are integer values rounded up. The results in Table IV not only verify the effectiveness of the AACS algorithm, but also show that all the AHI mechanism, vRHC strategy, and PSM mechanism contribute to the effectiveness and efficiencies of the AACS.

As shown in Table IV, FCFS still obtains quite poor results for any number of vessels. It is worth mentioning that in Test Case 7, the result of FCFS is better than ACO and ACS because most of the berthing time of the vessel in this random Test Case is exactly less than the interval of arrival times. Although there are more vessels in Test Case 7, the result is a relatively small value. Therefore, FCFS can obtain better results, while the ACO and ACS global search capabilities are too strong for faster convergence. However, the AEA algorithm and the related algorithm we proposed can both jump out of the local solution to find a better solution. When the number of vessels is 10, each algorithm can find the best solution except for FCFS. The mean value of AEA and AACS has also reached the optimal solution, which means that the optimal solution can be found each time it is executed. As the number of vessels increases, only the AACS algorithm can find

a better solution, but the average CPU time of the algorithm is longer. Given that it benefits from the vRHC strategy, the mean CPU time of AACS is better than those of AEA, ACO, and ACS.

G. Effectiveness of AACS on Medium- and Large-Scale Test Cases

We also further evaluate the performance of AACS in a set of medium- and large-scale test cases. The experiments are conducted on 10 test cases with the numbers of vessels from 50 to 120, and the number of berths being 2 or 4, as shown in Table S.II in the supplementary material. Moreover, the LUT_v of Test Case 11 to Test Case 20 are generated by $U(45, 240)$. The experimental results are compared in Table S.II.

As shown in Table S.II, AACS still can obtain the best results in medium- and large-scale test cases among the compared algorithms. In total, the proposed AACS achieves the best performance with respect to the minimal and mean *TWT* values in all the test cases. This indicates that our AACS is also effective in solving both medium- and large-scale problems. Moreover, considering the results with respect to the computational time, AACS also outperforms the compared algorithms in most cases. Although the AEA obtains better solutions in Test Case 14 and Test Case 16 with the least CPU time, in all the other cases, the CPU times of AACS are less than AEA and ACO. Besides, the results show that with more berths, the advantage of AACS in computational burden is more obvious. This may be due to that the number of sub-problems is fewer in vRHC strategy if more berths are used (i.e., more vessels are included in every sub-BAP and therefore the number of sub-BAP is smaller).

V. Conclusion

In this paper, to adapt to the discrete and real-time characteristics of BAP, we design adaptive heuristic information for ACS to propose the AACS by considering several given pieces of the BAP information. Furthermore, with respect to the computational burden of BAP, we have also proposed a variable domain RHC strategy to divide BAP into smaller problems. Moreover, a memory mechanism is utilized to improve convergence speed. To evaluate the effectiveness and efficiency of the proposed AACS, we generate test cases based on different BAP sizes. The test results show that our algorithm improves effectiveness and efficiency. Each mechanism of the algorithm is investigated separately. Experimental results show that all mechanisms (i.e., the AHI mechanism, the vRHC strategy, and the PSM mechanism) enhance the performance of the proposed AACS.

For future research, we plan to extend the algorithm to solve multi-objective BAP and multitasking BAP. Moreover, the BAP model with more characteristics such as uncertain circumstances is also worthy to be studied.

References

[1] Z.-G. Chen, Z.-H. Zhan, S. Kwong, and J. Zhang, "Evolutionary computation for intelligent transportation in smart cities: A survey," *IEEE Comput. Intell. Mag.*, vol. 17, no. 2, pp. 83–102, May 2022.

[2] (Jan. 2021). *Review of Maritime Transport*. [Online]. Available: https://unctad.org/topic/transport-and-trade-logistics/review-of-maritime-transport

[3] H.-P. Hsu, "A HPSO for solving dynamic and discrete berth allocation problem and dynamic quay crane assignment problem simultaneously," *Swarm Evol. Comput.*, vol. 27, pp. 156–168, May 2016.

[4] G. R. Mauri, G. M. Ribeiro, L. Antonio, N. Lorena, and G. Laporte, "An adaptive large neighborhood search for the discrete and continuous berth allocation problem," *Comput. Oper. Res.*, vol. 70, pp. 140–154, Jun. 2016.

[5] S. Kordić, T. Davidović, N. Kovač, and B. Dragović, "Combinatorial approach to exactly solving discrete and hybrid berth allocation problem," *Appl. Math. Model.*, vol. 40, nos. 21–22, pp. 8952–8973, Nov. 2016.

[6] M. A. Dulebenets, R. Moses, E. E. Ozguven, and A. Vanli, "Minimizing carbon dioxide emissions due to container handling at marine container terminals via hybrid evolutionary algorithms," *IEEE Access*, vol. 5, pp. 8131–8146, 2017.

[7] Y. Qian, D. Hu, Y. Chen, Y. Fang, and Y. Hu, "Adaptive neural network-based tracking control of underactuated offshore ship-to-ship crane systems subject to unknown wave motions disturbances," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Apr. 15, 2021, doi: 10.1109/TSMC.2021.3071546.

[8] R. T. Cahyono, E. J. Flonk, and B. Jayawardhana, "Discrete-event systems modeling and the model predictive allocation algorithm for integrated berth and quay crane allocation," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1321–1331, Mar. 2020.

[9] H. Li, J. Peng, X. Wang, and J. Wan, "Integrated resource assignment and scheduling optimization with limited critical equipment constraints at an automated container terminal," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7607–7618, Dec. 2021.

[10] Z. Zhang and C.-Y. Lee, "Multiobjective approaches for the ship stowage planning problem considering ship stability and container rehandles," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 10, pp. 1374–1389, Oct. 2016.

[11] T. W. Liao, "Integrated outbound vehicle routing and scheduling problem at a multi-door cross-dock terminal," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5599–5612, Sep. 2021.

[12] G. Maione, A. M. Mangini, and M. Ottomanelli, "A generalized stochastic Petri net approach for modeling activities of human operators in intermodal container terminals," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 4, pp. 1504–1516, Oct. 2016.

[13] C. Bierwirth and F. Meisel, "A survey of berth allocation and quay crane scheduling problems in container terminals," *Eur. J. Oper. Res.*, vol. 202, no. 3, pp. 615–627, May 2010.

[14] C. Bierwirth and F. Meisel, "A follow-up survey of berth allocation and quay crane scheduling problems in container terminals," *Eur. J. Oper. Res.*, vol. 244, no. 3, pp. 675–689, Aug. 2015.

[15] A. Imai, E. Nishimura, and S. Papadimitriou, "The dynamic berth allocation problem for a container port," *Transp. Res. B, Methodol.*, vol. 35, pp. 401–417, May 2001.

[16] J. F. Cordeau, G. Laporte, P. Legato, and L. Moccia, "Models and tabu search heuristics for the berth allocation problem," *Transp. Sci.*, vol. 39, no. 4, pp. 525–538, Nov. 2005.

[17] R. M. de Oliveira, G. R. Mauri, and L. A. Nogueira Lorena, "Clustering search for the berth allocation problem," *Expert Syst. Appl.*, vol. 39, no. 5, pp. 5499–5505, Apr. 2012.

[18] S.-W. Lin and C.-J. Ting, "Solving the dynamic berth allocation problem by simulated annealing," *Eng. Optim.*, vol. 46, no. 3, pp. 308–327, Mar. 2014.

[19] M. A. Dulebenets, "Application of evolutionary computation for berth scheduling at marine container terminals: Parameter tuning versus parameter control," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 25–37, Jan. 2018.

[20] M. M. Golias, G. K. Saharidis, M. Boile, S. Theofanis, and M. G. Ierapetritou, "The berth allocation problem: Optimizing vessel arrival time," *Maritime Econ. Logistics*, vol. 11, no. 4, pp. 358–377, Dec. 2009.

[21] C.-J. Ting, K.-C. Wu, and H. Chou, "Particle swarm optimization algorithm for the berth allocation problem," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1543–1550, Mar. 2014.

[22] Y. Cai, Y. Huo, and M. Yu, "Optimizing berth allocation by an artificial fish swarm algorithm," in *Proc. Int. Conf. Logistics Eng. Intell. Transp. Syst.*, Nov. 2010, pp. 26–28.

[23] C. J. Tong, H. C. Lau, and A. Lim, "Ant colony optimization for the ship berthing problem," *Lect. Notes Comput. Sci.*, vol. 1742, pp. 359–370, Jun. 1999.

[24] C. Y. Cheong and K. C. Tan, "A multi-objective multi-colony ant algorithm for solving the berth allocation problem," in *Advances of Computational Intelligence in Industrial Systems*. Berlin, Germany: Springer, 2008, pp. 333–350.
[25] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.
[26] X. Zhang, Z. H. Zhan, W. Fang, P. Qian, and J. Zhang, "Multi-population ant colony system with knowledge-based local searches for multiobjective supply chain configuration," *IEEE Trans. Evol. Comput.*, early access, Jul. 15, 2021, doi: 10.1109/TEVC.2021.3097339.
[27] Z.-H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung, and Y. Li, "Cloud computing resource scheduling and a survey of its evolutionary approaches," *ACM Comput. Surv.*, vol. 47, pp. 1–33, Jul. 2015.
[28] X.-F. Liu, Z.-H. Zhan, J. D. Deng, Y. Li, T. L. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 113–128, Feb. 2018.
[29] Z.-H. Zhan *et al.*, "An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 399–412, Jun. 2010.
[30] Z.-G. Chen *et al.*, "Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2912–2926, Aug. 2019.
[31] D. Liang, Z.-H. Zhan, Y. Zhang, and J. Zhang, "An efficient ant colony system approach for new energy vehicle dispatch problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4784–4797, Nov. 2020.
[32] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 35, no. 7, pp. 814–824, Jul. 1990.
[33] X. B. Hu, W. H. Chen, and E. D. Paolo, "Multiairport capacity management: Genetic algorithm with receding horizon," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 254–263, Jun. 2007.

Rong Wang (Student Member, IEEE) received the B.S. degree from Anhui Jianzhu University, Hefei, China, in 2013, and the M.S. degree from the Guilin University of Electronic Technology, Guilin, China, in 2017. She is currently pursuing the Ph.D. degree with the South China University of Technology, Guangzhou, China. Her research interests mainly include evolutionary computation, swarm intelligence, and their applications in real-world problems.

Fei Ji (Member, IEEE) received the B.S. degree in applied electronic technologies from Northwestern Polytechnical University, Xi'an, China, in 1992, and the M.S. degree in bioelectronics and the Ph.D. degree in circuits and systems from the South China University of Technology, Guangzhou, China, in 1995 and 1998, respectively.

She was a Visiting Scholar with the University of Waterloo, Canada, from June 2009 to June 2010. She was with the City University of Hong Kong as a Research Assistant from March 2001 to July 2002 and a Senior Research Associate from January 2005 to March 2005. She is currently a Professor with the School of Electronic and Information Engineering, South China University of Technology. Her research focuses on wireless communication systems and networking. She was the Registration Chair and the Technical Program Committee Member of the IEEE 2008 International Conference on Communication Systems.

Yi Jiang (Student Member, IEEE) received the B.S. degree in computer science and technology from the South China University of Technology, Guangzhou, China, in 2020, where he is currently pursuing the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering.

His research interests mainly include computational intelligence, machine learning, and their applications in real-world problems.

Sheng-Hao Wu (Student Member, IEEE) received the B.S. degree in computer science and technology from the South China University of Technology, Guangzhou, China, in 2019, where he is currently pursuing the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering.

His research interests mainly include computational intelligence, machine learning, and their applications in real-world problems; and in environments of distributed computing and big data.

Sam Kwong (Fellow, IEEE) received the Ph.D. degree from the University of Hagen, Germany, in 1996.

He is currently a Chair Professor with the Department of Computer Science, City University of Hong Kong. His research interests include pattern recognition, evolutionary computations, and video analytics.

Prof. Kwong was elevated to a fellow of IEEE for his contributions to optimization techniques for cybernetics and video coding in 2014. He is the President-Elect of the IEEE Systems, Man, and Cybernetics (SMC). He was also appointed as an IEEE Distinguished Lecturer of the IEEE SMC Society in March 2017. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

Jun Zhang (Fellow, IEEE) received the Ph.D. degree from the City University of Hong Kong in 2002.

He is currently a Korea Brain Pool Fellow Professor with Hanyang University, South Korea. His current research interests include computational intelligence, cloud computing, operations research, and power electronic circuits. He has published over more than 150 IEEE TRANSACTIONS papers in his research areas.

Dr. Zhang was a recipient of the Changjiang Chair Professor from the Ministry of Education, China, in 2013; the National Science Fund for Distinguished Young Scholars of China in 2011; and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON CYBERNETICS.

Zhi-Hui Zhan (Senior Member, IEEE) received the bachelor's and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2007 and 2013, respectively.

He is currently the Changjiang Scholar Young Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. His current research interests include evolutionary computation algorithms, swarm intelligence algorithms, and their applications in real-world problems; and in environments of cloud computing and big data.

Dr. Zhan was a recipient of the IEEE Computational Intelligence Society (CIS) Outstanding Early Career Award in 2021, the Outstanding Youth Science Foundation from the National Natural Science Foundation of China (NSFC) in 2018, and the Wu Wen-Jun Artificial Intelligence Excellent Youth from the Chinese Association for Artificial Intelligence in 2017. His doctoral dissertation was awarded the IEEE CIS Outstanding Ph.D. Dissertation and the China Computer Federation Outstanding Ph.D. Dissertation. He is one of the world's top 2% scientists for both career-long impact and year impact in artificial intelligence and one of the highly cited Chinese researchers in computer science. He is currently the Chair of the Membership Development Committee in the IEEE Guangzhou Section and the Vice-Chair of the IEEE CIS Guangzhou Chapter. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the *Neurocomputing*, and the *Memetic Computing*.