IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 January 2023, accepted 14 February 2023, date of publication 20 February 2023, date of current version 6 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3246518

RESEARCH ARTICLE

Hybridizing Whale Optimization Algorithm With Particle Swarm Optimization for Scheduling a Dual-Command Storage/Retrieval Machine

HSIEN-PIN HSU[1] **AND CHIA-NAN WANG**[2]

[1]Department of Supply Chain Management, National Kaohsiung University of Science and Technology, Nanzih, Kaohsiung 81157, Taiwan
[2]Department of Industrial Engineering and Management, National Kaohsiung University of Science and Technology, Sanmin, Kaohsiung 807618, Taiwan

Corresponding author: Chia-Nan Wang (cn.wang@nkust.edu.tw)

ABSTRACT Whale optimization algorithm (WOA) and particle swarm optimization (PSO) have been used individually usually. However, a separate use of them has a limitation. Hybridizing WOA with PSO is expected to evolve solutions better due to the cooperation between whales and seabirds. Developing such kind of model is the focus of this research. A framework has been further proposed to best utilize such hybridizations for developing simulation-based optimization approaches. The framework has the advantage of integrating metaheuristic, simulation, and optimization seamlessly. It can waive the rigorous and labor-intensive optimization procedure required for traditional simulation. In this research, simulation-based optimization approaches are used to deal with the dual-command block scheduling problem of a manufacturing firm's storage/retrieval (S/R) machine in an automated storage/retrieval system. The S/R machine is mainly used to store/retrieve stock-keeping units in an automated storage/retrieval system. Three simulation-based optimization approaches, Hybrid1 (WOA+PSO), Hybrid2 (WOA+PSO), and Hybrid3 (WOA+PSO), have been developed. To investigate their effectiveness, experiments have been conducted to compare them with their base models, WOA and PSO, as well as the genetic algorithm (GA) and PWOA. The PWOA is an abbreviation of a hybridization of PSO and WOA proposed in a previous study. The experimental results show that Hybrid3 (WOA+PSO) outperforms Hybrid2 (WOA+PSO), Hybrid1 (WOA+PSO), WOA, PSO, PWOA, and GA. The uses of techniques such as hybridization, Neighborhood heuristic, and adaptive movements of whales empower Hybrid3 (WOA+PSO) the most.

INDEX TERMS Whale optimization algorithm, particle swarm optimization, storage/retrieval machine, scheduling.

I. INTRODUCTION

Managing stock-keeping units (SKUs) is one essential function of an enterprise. Many enterprises have introduced automated storage/retrieval systems (AS/RSs) to do this. As a result, AS/RSs have been widely used in various areas, including warehouses (WHs), distribution centers (DCs), and manufacturing firms. An AS/RS owns the following advantages: lower labor, higher security, and lower space [1]. In addition to accommodating SKUs, an AS/RS is capable of transferring SKUs from one place to another [1].

There are many kinds of AS/RSs in the world. For classification, Roodbergen and Vis [2] proposed a scheme based on the three main attributes, storage/retrieval (S/R) machine, handling, and rack, of an AS/RS. In an AS/RS, an S/R machine can have one, two, or multiple shuttles. Also, it can be attributed as aisle-captive or shareable. An aisle-captive S/R machine is limited to one aisle, while a shareable one can move to another. In terms of handling, an AS/RS is attributed as the man-on-board (MOB), end-of-aisle (EOA), or unit-load (UL). For the kind of MOB handling, a man is loaded on board for picking up SKUs. For the kind of EOA handling, SKUs are picked at the end of an aisle. For the kind of UL handling, SKUs are handled in the unit of pallet/bin.

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

In terms of mobility, the racks of an AS/RS are of the stationary or moveable type. In addition, the structure of a rack can be single-deep, double-deep, or multiple-deep. In this research, the considered AS/RS has the following features: single shuttle, aisle-captive S/R machine, EOA, single-deep, and stationary racks.

An AS/RS is affected by its working environment. In a WH or DC, the AS/RS is used to collect, store, and distribute SKUs. In a manufacturing firm, the AS/RS also needs to support manufacturing operations on the shop floor. The manufacturing operations of a product are defined in a manufacturing procedure. The materials and subparts of a product are defined in a Bill of Materials (BOM) [3]. The AS/RS operations can affect the manufacturing operations. In this research, the manufacturing operations of a manufacturing firm have been considered. This manufacturing firm has five production floors and is located in Taiwan. The production lines are distributed among the five floors and used to assemble motherboards for computers. Such manufacturing firms are common in small countries.

An AS/RS consists of hardware and software. The standard hardware components include storage/retrieval (S/R) machines, racks, and aisles. The S/R machine is the primary device used to store/retrieve SKUs. The racks are partitioned into small storage units, termed cells or bins. Each cell/bin has a coordinate (x, y), where x is a column number and y is a tier number. In an aisle-captive AS/RS, an S/R machine is limited within an aisle and serves the two racks of the aisle. At the front end of each aisle, there is an I/O station for loading/unloading SKUs. A computer receives service requests from the shop floor and dispatches them to the S/R machines.

Some software components are used to control the hardware components of the AS/RS. These include storage rules, retrieval rules, scheduling algorithms, and storage position assignment policies. For example, FCFS (first-come-first-serve) is a simple storage rule, and FIFO (first-in-first-out) is a simple retrieval rule. They have often been used to determine the execution sequence of service requests for an AS/RS. However, simple heuristics are too simple to achieve a good performance. The storage position policy is another software component affecting the AS/RS performance. The command mode can be another impact factor. Single and dual command modes are commonly used for an S/R machine. The former completes one storage/retrieval request in one round trip; the latter can complete one storage request and one retrieval request in one round trip. The dual command mode can have a shorter total traveling time [4]. Thus, this command mode is focused on in this research.

Approaches such as analytical models, heuristics, metaheuristics, and simulations have been used to optimize AS/RSs. However, each of these approaches has its weaknesses. The analytical models tend to become computationally intractable when used to deal with a big instance due to NP-hard. Simple heuristics manages to find a general solution because of simplicity. The metaheuristics are limited by their capability when used alone. The traditional simulation uses a rigorous and labor-intensive procedure to approach optimality. An improvement is necessary. Recently, metaheuristics such as genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO), and artificial bee colony (ABC) have been increasingly used to solve AS/RS problems. However, these metaheuristics have their limitation when used alone. One improving direction is developing hybridizations and integrating with the simulation approach to become simulation-based optimization approaches. To facilitate the development of such methods, a framework is necessary.

The whale optimization algorithm (WOA) was proposed by Mirjalili and Lewis [5]. As a metaheuristic, WOA models the hunting behavior of humpback whales. Since its appearance, this algorithm has been used to deal with various problems, including power systems, IoT, wireless sensor networks, etc. [6]. The [6] had conducted a thorough review of the WOA applications and suggested that hybridization of WOA is a future research direction. However, the hybridization of WOA and PSO has never been used to deal with the dual-command block scheduling problem (DCBSP) of an S/R machine. The [7] is one recent study that hybridized PSO with WOA to deal with the workflow scheduling problems in a cloud-fog-mobile computing environment. The hybridization was abbreviated as PWOA. The objective was to minimize the total execution time (TET) and total execution cost (TEC) of dependent tasks. The PWOA was claimed to be able to improve the trapping problem. The simulation results showed that PWOA outperformed its base models, PSO and WOA. Combining the hybridizations of WOA and PSO with a simulation model is expected to result in powerful simulation-based optimization approaches.

This research first defines the DCBSP. Then, the DCBSP is formulated as a MILP. The objective is to minimize total operational time. However, simulation-based optimization approaches have been further developed as alternative approaches due to NP-hard. To develop simulation-based optimization approaches, a framework is proposed. The framework is advantageous as it can integrate metaheuristics, simulation, and optimization seamlessly. Three hybrid models of WOA and PSO, namely Hybrid1 (WOA+PSO), Hybrid2 (WOA+PSO), and Hybrid3 (WOA+PSO), have been developed and used as alternative sequencing methods in the framework. Together with a discrete-event simulation model, they are used to deal with the DCBSP of an S/R machine in a unit-load aisle-captive AS/RS. To investigate their effectiveness, they have been compared to WOA and PSO, GA, and PWOA through extensive experiments. The results showed that Hybrid3 (WOA+PSO) outperformed the others in terms of total operational time.

The rest of this paper is structured as follows. Section II reviews the main approaches for dealing with the AS/RS problems. Section III first defines the block scheduling of an

S/R machine and then formulates this problem as a MILP model. Section IV proposes a framework for developing simulation-based optimization approaches to deal with the DCBSP. Three hybridizations of WOA and PSO have been created. Section V performs experiments and analyzes the results. Section VI concludes and suggests future research directions.

II. LITERATURE REVIEW

A. ANALYTICAL MODELS

Analytical models have been used to optimize AS/RS. Bozer and White [8] proposed analytical models to analyze the operational time of a unit-load AS/RS. The single and dual command operations of the S/R machine were studied. The objective was to optimize the dimensions of the AS/RS. Park et al. [9] optimized the dwell-point policy for the S/R machines in an AS/RS. The AS/RS comprises square-in-time racks and uses a dedicated storage policy. The authors found that input points were good candidates for dwell points for the S/R machines. Koh et al. [10] studied the order-picking system in a mini-load AS/RS. The AS/RS consisted of multiple racks. Each aisle has a horse-shoe-style buffer installed at its front end. An order picker served among these aisles. In addition, a conveyor system moved through these buffer points for container transport. This AS/RS was modeled as a queueing system with a limited queue length. A supplementary variable technique was proposed to study this system and determine the buffer size. Lerher et al. [11] used an analytical model to estimate the cycle time of a multi-aisle AS/RS. In this research, the acceleration and deceleration of the S/R machine have been specially considered. Manzini et al. [12] optimized the design and management of a warehouse. Two cost-based mixed integer linear programming (MILP) models were proposed. The selections of part-to-picker systems and picker-to-part order picking systems were considered. A class-based storage assignment was proposed. It considered the turnover rate (popularity) of items over a life cycle was used. Foumani et al. [13] studied an AS/RS in which a Cartesian robot picks and palletizes items onto a mixed pallet. It retrieves orders in an optimal sequence and creates an optimal store-ready pallet of any order. The decisions to be made include finding the optimal sequence of orders and the optimal sequence of items inside each order. The objective was to minimize total travel time. A two-phase procedure was used. In the first phase, an avoidance strategy of movement sequence was developed for the robot (or automatic stacker crane). A Cross-Entropy (CE) method was proposed in the second phase to make these decisions. The comparison to CPLEX confirmed the efficiency of the CE.

However, optimizing a given set of service requests is an NP-hard problem [14], making exact approaches such as Integer linear programming (ILP) and mixed-integer linear programming (MILP) computationally intractable when dealing with a big instance. Alternative approaches, such as heuristics and metaheuristics, are more practical.

B. HEURISTICS

Heuristics have been widely used for AS/RSs in industries. The main merits of heuristics are their simplicity and ease of use. Hwang and Chang [15] proposed a rack-class-based storage assignment and a class selection procedure to minimize the number of S/R machines in an aisle and the number of aisles in an AS/RS. Sarker et al. [16] investigated the effectiveness of the nearest-neighbor heuristic and perimeter heuristic used to deal with the dual-shuttle S/R machine scheduling problem and layout problem in a warehouse. Larson et al. [17] optimized the aisle layout, storage zone dimensions, assignment of material to a storage medium, and floor space allocation for an AS/RS. A class-based storage method was proposed. This method can increase floor space utilization and decrease operational time. Mahajan et al. [18] used a nearest-neighbor heuristic to deal with the dual-command S/R machine scheduling problem in an AS/RS. Yang et al. [19] examined the joint optimization problem of storage location assignment and storage/retrieval scheduling in a multi-shuttle AS/RS. They found that a shared storage policy is beneficial. The authors further proposed a variable neighborhood search (VNS). Their experiments confirmed that the VNS was effective. Wauters et al. [20] proposed a decomposition heuristic to deal with the location assignment problem and sequencing problem for a dual-shuttle S/R machine in an AS/RS. Li and Chen [21] used Hungarian-based heuristics to sequence dual commands and assign storage locations for SKUs for an S/R machine in a flow-rack AS/RS.

However, simple heuristics tend to find a general solution due to their simplicity.

C. METAHEURISTICS

The use of metaheuristics to deal with the S/R machine scheduling problem in an AS/RS is a current trend. This is especially true for population-based metaheuristics, such as GA, ACO, and PSO. Available benefits of this kind of metaheuristics include population advantage and swarm intelligence (SI). Yang et al. [22] proposed a Hybrid Genetic Algorithm (HGA) to deal with the multi-shuttle S/R machine scheduling problem. The objective was to minimize the travel time. Wu et al. [23] focused on reducing the retrieval times for the S/R machine in a double-deep AS/RS. The AS/RS was in a Flexible Manufacturing System. The effectiveness of genetic algorithm (GA), immune GA (IGA), and particle swarm optimization (PSO) were investigated in that study. However, that study did not consider dual commands. Popovic et al. [24] used GA to deal with the S/R machine scheduling problem in an AS/RS. The S/R machine was equipped with triple shuttles. Brezovnik et al. [25] used ant colony optimization (ACO) to optimize the operations of an AS/RS. That study considered various factors, including inquiry (FOI), product height (PH), storage space usage (SSU), and path to dispatch (PD). Nia et al. [26] also applied

ACO to deal with the dual-command S/R scheduling problem in a unit-load multi-rack AS/RS. The objective was to minimize the total cost of GHG efficiency. Particularly, that study took the environmental issue into consideration. It was found that the ACO outperformed the GA. Chen et al. [27] hybridized ACO with GA to solve the routing problem of an S/R machine in a multi-block warehouse. Twelve different layouts of the warehouse were used to investigate the effectiveness of the hybrid model. It was found that the hybrid model outperformed the standard ACO and GA. Cunkas and Ozer [28] optimized the assignment of storage positions for SKUs in a unit-load AS/RS. The S/R machines of the AS/RS are equipped with dual shuttles and use the quadruple command cycle. A PSO approach was proposed. It outperformed the Binary Coded GA (BGA) and Real Coded GA (RGA). Tostani et al. [29] proposed a modified Cooperative Coevolutionary Algorithm (MCoBRA) to deal with the dual-shuttle S/R scheduling problem in an AS/RS. A bi-level optimization model was used. The upper level allocates SKU storage locations based on a class-based storage policy. The lower level solves the S/R machine scheduling problem. The objective was to balance operational costs and energy consumption. Hojaghani et al. [30] studied the online order batching problem of an AS/RS. A mixed-integer nonlinear programming (MINLP) model was formulated for this problem. In addition, ACO and artificial bee colony (ABC) were used as alternatives. The objective was to improve the response and idle times. The ACO was found to outperform the ABC.

The above literature review shows that applying metaheuristics to solve the S/R machine is a current trend. Metaheuristics such as GA, ACO, PSO, and ABC have been used to deal with the S/R machine scheduling problem. However, hybrid models of WOA and PSO have never appeared for this purpose. Though the WOA is advantageous in terms of simple structure, fewer operators, fast convergence speed, and balanced exploration and exploitation [5], it is limited when used alone. In addition, this algorithm cannot be applied to solve a discrete problem directly. Another weakness found for the WOA is the lack of using adaptive movements for whales to better approach the prey. Improving this algorithm is necessary.

D. SIMULATION STUDIES

The simulation approach is suitable for studying dynamic systems [1], such as the AS/RS. Many researchers used simulations to optimize AS/RSs. Ashayeri et al. [31] used simulation to optimize the design of a unit-load warehouse in an oil company. Han et al. [14] and Azzi et al. [32] used Monte Carlo simulation to optimize the operations of an AS/RS. Han et al. [14] focused on improving the throughput of a unit-load AS/RS. The FCFS was used as a sequencing rule, while the nearest-neighbor heuristic was used as an alternative. Azzi et al. [32] used simulation to improve the travel time of a multi-shuttle S/R machine in a unit-load AS/RS. Randhawa and Shroff [33] examined the effectiveness of different sequencing rules on six different layout configurations of a unit-load AS/RS. The I/O stations, SKU distribution, and rack distribution are variables to be optimized. Lee et al. [34] used a simulation software, ARENA, to optimize the deployment number of rail-guided vehicles (RGVs) in an AS/RS. Hachemi et al. [35] also used simulation to solve the dual-command S/R machine scheduling problem in a unit-load multi-rack AS/RS. The objective was to minimize the travel time of the S/R machine. The FCFS was used to handle storage requests. Yang et al. [19] solve the joint optimization problem of the S/R machine scheduling and storage location assignment in a unit-load multi-shuttle AS/RS. An effective variable neighborhood search (VNS) algorithm was proposed. The above literature review found that without a close connection between the simulation and optimization, traditional simulations use a rigorous and labor-intensive optimization procedure. An improvement is necessary.

III. PROBLEM DEFINITION AND MATHEMATICAL MODEL FORMULATION

A. DEFINITION OF THE DUAL-COMMAND BLOCK SEQUENCING PROBLEM OF AN S/R MACHINE

Two kinds of approaches are available to deal with a sequence problem: block sequencing and dynamic sequencing [14]. The dual-command block sequencing problem (DCBSP) of an S/R machine is focused and defined as follows.

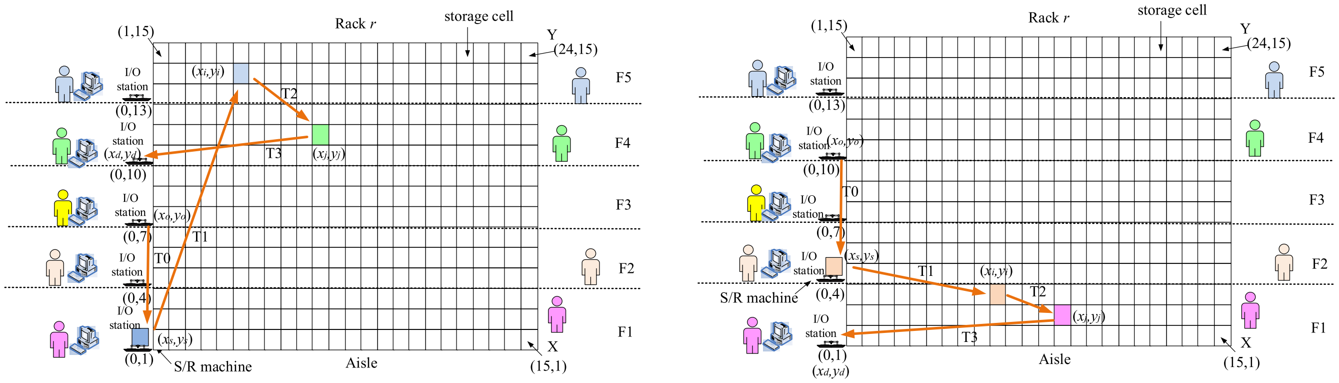
Definition 1 (Dual-Command Block Sequencing Problem (DCBSP) of an S/R Machine): This is a problem of forming and scheduling dual commands for an S/R machine in a unit-load aisle-captive AS/RS. To solve this problem, a block sequencing approach is necessary. The approach first selects the most urgent number of storage and retrieval requests from the shop floor. Then, it transforms these requests into dual commands. Subsequently, these dual commands are sequenced and to be executed by the S/R machine. Once a block of service requests is completed, the next block is processed continuously. In this research, the hybridizations of WOA and PSO are used as block sequencing approaches. The objective is to minimize total operational time. The "unit-load" feature indicates that SKUs are handled in the units of a pallet in the AS/RS, and the "aisle-captive" stipulates that the S/R machine serves two racks of one specific aisle only in this AS/RS. The S/R machine moves in the AR/RS horizontally and vertically simultaneously. The moving distance is measured by the Chebychev metric. The racks are stationary and single-deep. According to the classification scheme proposed by Boysen and Stephan [36], the focused problem is denoted as $[F|O^2|\text{Open }|\text{Min}\sum_{i,j\in N}\text{OT}(i,j)]$. It owns the following features:

F: the I/O station is located at the front end of an aisle.

Open: any open storage cell is a potential location for a storage request.

O^2: Dual-command mode (a pair of storage and retrieval commands).

$\sum_{i,j\in N}\text{OT}(i,j)$: the total operational time for an S/R machine to complete the number of N dual-commands. Each dual



1 (a). The execution of the first dual command

1 (b). The execution of the second dual command

FIGURE 1. **The example of executing a two dual-command in a unit-load AS/RS consecutively.**

command is denoted as (i, j), where i indicates the i-th storage request while j indicates the j-th retrieval request.

Min $\sum_{i,j\in N}$ OT(i, j): the objective is to minimize the total operational time required for an S/R machine to complete the number of N dual-commands.

B. ANALYSIS OF THE OPERATIONAL TIME FOR DUAL-COMMAND

Fig. 1 shows the consecutive executions of two dual commands of the same S/R machine. As shown in Fig. 1(a), the S/R machine first moves from the dwell point (x_o, y_o) to the I/O station at the coordinate (x_s, y_s) on the 1st floor. This traveling time is denoted as T0. After loading SKU i, the S/R machine moves to the assigned storage position (x_i, y_i). This traveling time is denoted as T1. Then, the shuttle of this S/R machine puts the SKU i into its storage position. This shuttle time is denoted as Ts. After this, the S/R machine continues to retrieve SKU j at the storage location (x_j, y_j). This traveling time is denoted as T2. Subsequently, the shuttle retrieves SKU j. This shuttle time is denoted as Tr. Finally, the S/R machine moves SKU j to its destination I/O station with the coordinate (x_d,y_d). This traveling time is denoted as T3. As shown in Fig. 1(b), the S/R machine continues to execute the following dual command. First, it stores an SKU from the 2nd floor to the 1st floor. Then, it retrieves another SKU and moves it to the I/O station on the 1st floor.

Based on the above scenarios, Equation (1) defines the operational time, OT(i, j), for the S/R machine to complete a dual-command (i, j).

$$\text{OT }(i, j) = \text{travel time} + \text{shuttle time} \tag{1}$$

It includes travel time and shuttle time, which are dependent on the dimensions of the racks and the characteristics of the S/R machine [14].

Based on the Chebychev metric, Equation (2) defines the formula for calculating the travel time.

$$\begin{aligned}\text{Travel time} &= T0 + T1 + T2 + T3\\ &= \frac{H\cdot(y_{s-}y_0)}{v_v}\\ &\quad + \text{Max}\left\{\frac{W\cdot(x_{i-}x_0)}{v_h}, \frac{H\cdot(y_{i-}y_0)}{v_v}\right\}\\ &\quad + \text{Max}\left\{\frac{W\cdot|x_i - x_j|}{v_h}, \frac{H\cdot|y_i - y_j|}{v_v}\right\}\\ &\quad + \text{Max}\left\{\frac{W\cdot(x_j - x_d)}{v_h}, \frac{H\cdot(y_j - y)}{v_v}\right\}\end{aligned} \tag{2}$$

Equation (3) defines the formula for calculating the shuttle time.

$$\text{shuttle time} = t_s + t_r = \frac{2D}{v_d} + \frac{2D}{v_d} \tag{3}$$

where

v_h: the moving speed of the S/R machine in the horizontal (X) direction (M/s).

v_v: the moving speed of the S/R machine in the vertical (Y) direction.

v_d: the moving speed of the shuttle on the S/R machine in depth (Z) direction.

W : the width of a storage cell (M).

H : the height of a storage cell (M).

D : the depth of a storage cell (M).

C. FORMULATION OF A MIXED INTEGER LINEAR PROGRAMMING (MILP) MODEL

In this section, the DCBSP is formulated as a MILP model. To do this, some assumptions are first made, and then the notations of indices, sets, parameters, and decision variables are defined.

Assumptions

- Each storage/retrieval request relates to an SKU with a storage/retrieval position (x,y), where the

x and y represent a column number and a tier number, respectively.

- The S/R machine moves in the horizontal and vertical directions simultaneously.
- No interruption during the transportation, storage, and retrieval of an SKU.
- The numbers of storage and retrieval requests are the same.
- Each aisle has an I/O station at its front end.
- Each dual command involves a source I/O station for loading an SKU and a destination I/O station for unloading an SKU.
- The I/O station on the 1st floor is initialized as the dwell position of an S/R machine. Afterward, the destination I/O station in the last dual command is set as the dwell position of the S/R machine.

Notations

Indices

i, i', j'	a storage request ID; $i, i', j' \in$ SR;
j	a retrieval request ID; $j \in$ RR;
r	a rack number; $r \in R$;
x	a column number on a rack; $x \in$ X;
y	a tier number on a rack; $y \in$ Y;

Sets

SR	a set of storage requests in a block; SR $= \{1, \ldots, N\}$;
RR	a set of retrieval requests in a block; RR $= \{1, \ldots, N\}$;
R	a set of rack; R $= \{1, \ldots, \|\text{R}\|\}$;
X	a set of columns; X $= \{1, \ldots, \|\text{X}\|\}$;
Y	a set of tiers on a rack; Y $= \{1, \ldots, \|\text{Y}\|\}$;

Parameters (Input data)

$\|\text{R}\|$	the number of racks served by a crane;
$\|\text{X}\|$	the total of columns of a rack;
$\|\text{Y}\|$	the total number of tiers of a rack;
N	the total number of storage/retrieval service requests
(x_i, y_i)	the storage position of an SKU i in a rack;
(x_j, y_j)	the storage position of an SKU i' in a rack;
v_h	the moving speed of a crane along the X horizontal (column) direction;
v_v	the moving speed of a crane along the Y vertical (tier) direction;
v_d	the moving speed of a shuttle along the depth direction;
t_s	the shuttle time for storing an SKU into the AS/RS;
t_r	the shuttle time for retrieving an SKU from the AS/RS;

Decision variables

X_{ij}	$= 1$, if the retrieval request j is assigned to the storage request i; $= 0$, otherwise;
$Y_{i'j'}$	$= 1$, if the storage request i' is performed before the storage request j'; $= 0$, otherwise;

A sequence of dual commands of storage and retrieval requests is to be created. The MILP model for the crane scheduling problem is formulated as follows.

$$\text{Min } Z = \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{N} OT(i,j) \cdot X_{ij} \tag{4}$$

$$s.t.\ OT(i,j) \geq 0 \forall i,j \in \text{SR} \tag{5}$$

$$\sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{N} X_{ij} = N \tag{6}$$

$$\sum\nolimits_{i'=1}^{N} Y_{i'j'} = 1 \quad \forall j' \in \text{SR} \tag{7}$$

$$\sum\nolimits_{j'=1}^{N} Y_{i'j'} = 1 \quad \forall i' \in \text{SR} \tag{8}$$

$$Y_{i'j'} \leq (N-1) Y_{i'j'} \forall i', j' \in \text{SR} \tag{9}$$

$$\sum\nolimits_{j'=1}^{N} Y_{i'j'} = \sum\nolimits_{j'=1}^{N} Y_{i'j'} + 1 \forall i', j' \in \text{SR} \tag{10}$$

$$X_{ij,} Y_{i'j'} \in \{0, 1\} \quad \forall i, i', j' \in \text{SR} \quad \forall \text{j} \in \text{RR} \tag{11}$$

Equation (4) is the objective function to minimize the total operational time for completing all service requests. This model also includes Equations (1)–(3). Constraint (5) requires the value of the variable $OT(i,j)$ to be zero or positive. Equation (6) is associated with the variable X_{ij}, which models an assignment problem. It stipulates that one storage request is only assigned with one retrieval request. Equations (7)–(9) relate to variable $Y_{i'j'}$, which models a sequencing problem from a network view. Equation (7) states that each storage request j' has only predecessor i'. Equation (8) stipulates that each storage request i' has only one successor j'. From a network view, constraint (9) specifies that there are at most $N-1$ storage requests to assign when leaving the node i'. Equation (10) is a conservation constraint for node i'. Constraint (11) defines that X_{ij} and $Y_{i'j'}$ are binary variables.

IV. SIMULATION-BASED OPTIMIZATION APPROACHES

A. A SIMULATION-BASED OPTIMIZATION FRAMEWORK

The DCBSP has a solution space of $N! \times N!$. Applying the MILP to solving the DCBSP (with a big N) to optimality will be NP-hard. Thus, simulation-based optimization approaches are proposed as alternatives. For their development, a framework (Fig. 2) is proposed. The framework's main steps are detailed as follows.

1) **Set parameter values**: this step initializes the parameter values, such as the number of S/R machines and racks, the number of columns and the number of tiers in a rack, the parameter values of metaheuristics, the number of storage/retrieval requests, the number of iterations.
2) **Generate S/R requests**: this step automatically generates storage and retrieval requests using a computer. This step's output is random storage and retrieval requests.

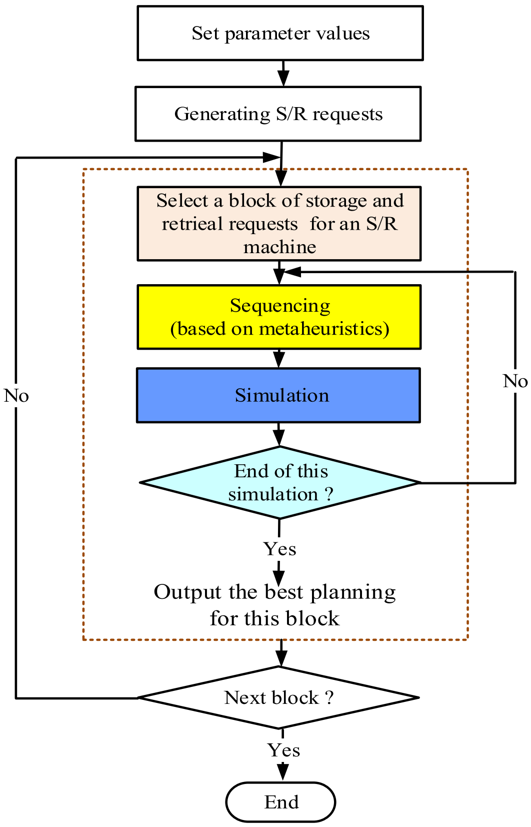

Set parameter values
Generating S/R requests
Select a block of storage and retrieal requests for an S/R machine
Sequencing (based on metaheuristics)
Simulation
End of this simulation ?
No
Yes
Output the best planning for this block
No
Next block ?
Yes
End

FIGURE 2. The framework for integrating metaheuristic, simulation, and optimization seamlessly.

3) **Select a block of storage and retrieval requests for a specific S/R machine:** this step filters out the storage and retrieval service requests for a specific S/R machine. The rack numbers of the SKUs are used. For example, the SKUs on racks No.1 and No.2 are dispatched to S/R machine No. 1. This step's output is the storage and retrieval requests dispatched to each S/R machine.
4) **Sequencing**: with the previous step's output, this step focuses on forming and sequencing dual commands. This step's output is a sequence of dual commands.
5) **Simulation:** with the previous step's output, this step simulates the execution of the sequences of dual commands for the S/R machine. Meanwhile, each sequence of dual commands is evaluated, and the best one is identified.
6) **End of this simulation check**: this step checks whether the termination condition is met. If "yes," the simulation is terminated and goes to Step 7); otherwise, it goes back to Step 4).
7) **Output**: this step outputs the best sequence of dual commands (solution) for each block of service requests.
8) **Next block check:** this check determines whether to continue the next block. A total number of blocks can be used as a termination condition.

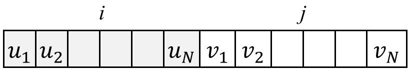


FIGURE 3. The position scheme used by whales.

B. POSITION SCHEME

Fig. 3 shows the position scheme of whales/seabirds in the solution space. There are two segments. Each segment is N-dimensional, where N indicates the number of storage/retrieval requests.

The left segment is used to represent a sequence of storage requests. The u_i is an order assigned for the storage request i. The right segment is used to represent a sequence of retrieval requests. The v_j is an order of the retrieval request j. The series of N dual commands is formed and has the paired representation as $(u_1, v_1), (u_2, v_2), \ldots, (u_N, v_N)$.

To form a feasible solution, a rank order value (ROV) technique is used to transform the position vector of a whale/seabird from a continuous domain (real values) to a discrete domain (integers). For example, the real set [0.1,0.4,0.3] is transformed into the ranking set [1,3,2], representing an operational sequence.

Algorithm 1 . The Logic Flow of the Standard WOA

```
Define the objective function F(X) of the problem
Set values for parameter $Pw$, $N$, $T$, $a$, $A$, $C$, $l$, $b$, and $p$
Generate a block of storage and retrieval requests
Initialize the positions $Xi$ of whales
Calculate the F($Xj$) for each whale $j$; Find the $\overrightarrow{X_{w^*}}(t)$
Set $t = 1$ // $t$ is iteration counter
REPEAT1 until ($t = T$)
Set $i = 1$ // $i$ is the population counter
REPEAT2 until ($i = Pw$)
    Update $a$, $|\vec{A}|$, $\vec{C}$, $l$, and $p$
    IF ($p$< 0.5)  // shrink encircling movement
            IF (|A| < 1) // exploitation
                Update the $Xi$ with Eq. (12) and (14)
            ELSE //exploration
                Select a random $X_R$ from the whale group
                Update the $Xj$ with Eq. (17) and (18)
            ENDIF
    ELSE // spirial track movement
            Update the $Xi$ with Eq. (12) and (16)
    END IF
END REPEAT2
Check if any whale out of search space and amend it
Evaluate F($X_i$) for each whale $i$
Update the $\overrightarrow{X_{w^*}}(t)$ if better
END REPEAT1
Output $\overrightarrow{X_{w^*}}$
```

C. HYBRID MODELS OF WOA AND PSO

This section first introduces the standard WOA and PSO and then develops three hybridizations, namely Hybrid1 (WOA+PSO), Hybrid2 (WOA+PSO), and Hybrid3 (WOA+PSO).

1) WOA

Fig. 4 shows the whale's spiral-track movement when encircling and attacking the prey (fish school) [25]. The $\vec{D}_i$ is the

Algorithm 2 . The Logic Flow of the Standard PSO ( )

```
Define the objective function F(X) of the problem
Set values for parameter $Ps$, $N$, $T$, $\overline{V}$, $\underline{V}$, and $W$
Generate a block of storage and retrieval requests
Initialize the positions $Xj$ of particles
Set $t = 1$ // $t$ is iteration counter
REPEAT1 until ($t = T$)
    Set $j = 1$ // $i$ is the population counter
    REPEAT2 until ($j$=$Ps$)
        Update Equations (19) and (20).
        Calculate the F($Xj$) for each particle (seabird) $j$.
        Compare and store the better personel best seabird $P_j$.
        Compare and store the global best seabird $\overrightarrow{X_{s^*}}\,(t)$.
    END REPEAT2
END REPEAT1
Output $\vec{X}_{s^*}y$
```

Algorithm 3 . The Logic Flow of the Hybrid1 (WOA+PSO)

```
Define the objective function $F(X)$ of the problem
Set values for parameter $PW$, $PS$, $N$, $T$, $a$, $A$, $C$, $l$, $b$, and $p$
Generate a block of storage and retrieval requests
Set the parameters $c_1$, $c_2$ and inertia weight $w$ // for PSO
Initialize the positions $Xi$ of whales
Initialize the positions $Xj$ of seabirds
Evaluate the F($Xi$) for each whale $i$; Find the $\vec{X}_{w^*}\,(t)$
Evaluate the F($Xj$) for each seabird $j$;
Set $t = 1$ // $t$ is iteration counter
REPEAT1 until ($t = T$)
Call PSO( ) and find the best $\vec{X}_{s^*}\,(t)$;   // call PSO
If $\vec{X}_{s^*}\,(t)$ is better then update the $\vec{X}_{w^*}\,(t)$ with the $\vec{X}_{s^*}\,(t)$
Set $i = 1$ // $i$ is the population counter
REPEAT2 until ($i$ =$Pw$)
    Update parameters $a$, $\left|\vec{A}\right|$, $\vec{C}$, $l$, and $p$
    IF ($p$< 0.5) // shrink encircling movement
        IF (|A| < 1) // Exploitation
            Update the $Xi$ with Eq. (12) and (14)
        ELSE   // Exploration
            Select a random $X_R$ from the whale group
            Update the $Xi$ with Eq. (17) and (18)
        ENE IF
    ELSE // spirial track movement
        Update the $Xi$ with Eq. (12) and Eq. (16)
    END IF
    $i$=$i$+1
END REPEAT2
Check and amend the whales out of the search space
Evaluate F($X_i$) for each whale $i$;
Update the global best $\vec{X}_{w^*}(t)$
$t$=$t$+$1$
END REPEAT1
Output $\vec{X}_{w^*}$
```

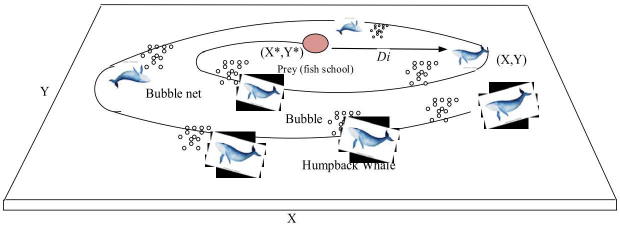


FIGURE 4. The spiral movements of humpback whales.

Algorithm 4 . The Logic Flow of the Hybrid2 (WOA+PSO)

```
Define the objective function $F(X)$ of the problem
Set values for parameter $PW$,$PS$, $N$, $T$, $a$, $A$, $C$, $l$, $b$, and $p$
Generate a block of storage and retrieval requests
Generate distance table // for nearest-neighbor
Set the parameters $c_1$, $c_2$ and inertia weight $w$ // for PSO
Initialize the positions $Xi$ of whales
Initialize the positions $Xj$ of seabirds
Evaluate the F($Xi$) for each whale $i$; Find the $\overrightarrow{X_{w^*}}\,(t)$
Evaluate the F($Xj$) for each seabird $j$;
Set $t = 1$ // $t$ is iteration counter
REPEAT1 until ($t = T$)
Call PSO( ) and find the best $\overrightarrow{X_{s^*}}\,(t)$;   // call PSO
If $\overrightarrow{X_{s^*}}\,(t)$ is better then update the $\overrightarrow{X_{w^*}}\,(t)$ with the $\overrightarrow{X_{s^*}}\,(t)$
Set $i = 1$ // $i$ is the population counter
    REPEAT2 until ($i$ =$Pw$)
    Update parameters $a$, $\left|\vec{A}\right|$, $\vec{C}$, $l$, and $p$
    IF ($p$ < 0.5) // shrink encircling movement
        IF (|A| < 1) // Exploitation
            Update the $Xi$ with Eq. (12) and (14)
        ELSE   // Exploration
            Select a random $X_R$ from the whale group
            Update the $Xi$ with Eq. (17) and (18)
        ENE IF
    ELSE // spirial track movement
        Update the $Xi$ with Eq. (12) and Eq. (16)
    END IF
    $i$=$i$+1
    END REPEAT2
    Check and amend the whales out of the search space
    Evaluate F($X_i$) for each whale $i$;
    Update the global best $\overrightarrow{X_{w^*}}\,(t)$
    $t$=$t$+$1$
END REPEAT1
Output $\overrightarrow{X_{w^*}}$
```

distance between the whale i and the prey. To hunt the prey, the efforts for each whale i is to shorten the $\vec{D}_i$. A whale's position corresponds to a solution in the solution space. Changing a whale's position simulates a search in the solution space.

Without knowing the exact position of the prey, the position of the global best whale w^* is used to represent the prey's position. Thus, whales in this group refer to the w^*.

Given $\vec{X}_i(t)$ and $\overrightarrow{X_{w^*}}\,(t)$ as the positions of whales i and w^*, respectively. Then, Equation (12) is used to determine the $\vec{D}_i$ for each whale.

$$\vec{D}_i = \left|\vec{C}\cdot\overrightarrow{X_{w^*}}\,(t) - \vec{X}_i(t)\right| \tag{12}$$

where $\vec{C}$ is a coefficient vector defined as follows.

$$\vec{C} = 2\cdot\vec{r} \tag{13}$$

where the $\vec{r}$ is a random vector in [0,1]. Given the distance $\vec{D}_i$, the next position of the whale i is determined by Equation (14).

$$\vec{X}_i(t+1) = \overrightarrow{X_{w^*}}\,(t) - \vec{A}\cdot\vec{D}_i \tag{14}$$

The $\vec{A}$ is a coefficient vector with values within [−1,1] and is defined by Equation (15).

$$\vec{A} = 2\vec{a}\cdot\vec{r} - \vec{a} \tag{15}$$

where the $\vec{a}$ is a linear value decreasing from 2 to 0 iteratively, the $|\vec{A}|$ is used to determine the use of exploration ($|\vec{A}| \geq 1$) or exploitation ($|\vec{A}| < 1$) for a whale.

In addition to the shrink-encircling movement, the whales in the WOA also employ the spiral track movement. Equation (16) defines the use of either a shrink-encircling movement or a spiral-track movement, which is controlled by a parameter p ($p \in [0,1]$).

$$\vec{X}_i(t+1) = \begin{cases} \overrightarrow{X_{w^*}}(t) - \vec{A} \cdot \vec{D}_i, & \text{if } p < 0.5 \\ \vec{D}_i \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X_{w^*}}(t), & \text{if } p \geq 0.5 \end{cases} \tag{16}$$

In Equation (16), if $p < 0.5$, then the shrink-encircling movement is used; Otherwise, the spiral-track movement is used, where the b is a constant defines the shape of logarithmic spiral and the l is a random number in [0,1].

Equations (17) and (18) are used for a whale i to perform an exploration search.

$$\vec{D}_i = \left|\vec{C} \cdot \vec{X}_R(t) - \vec{X}_i(t)\right| \tag{17}$$

$$\vec{X}_i(t+1) = \vec{X}_R(t) - |\vec{A}| \cdot \vec{D}_i \tag{18}$$

The $\vec{X}_R(t)$ is a position of a whale randomly selected from the swarm.

Algorithm 1 shows the main logic flow of WOA. Line 2 sets the parameter values for WOA, where Pw is the population of whales, N is the total number of storage/retrieval requests, and T is the total number of iterations. Line 3 generates a block of storage and retrieval requests for an experiment. Line 4 initializes the whales' positions. Line 5 evaluates the F(X_i) for each whale i and finds the $\overrightarrow{X_{w^*}}(t)$. Line 6 initializes the iteration counter $t = 1$. Line 8 initializes the whale population counter $i = 1$.

2) PSO

Proposed by Kennedy and Eberhart [26], the PSO was inspired by the social behavior of bird flock or fish school. This research treats a particle as a seabird with its flying velocity defined in Equation (18).

$$\vec{V}_j(t+1) = \omega\vec{V}_j(t) + c1R1(\vec{P}_j(t) - \vec{X}_j(t)) + c2R2(\overrightarrow{X_{s^*}}(t)) - \vec{X}_j(t)) \tag{19}$$

where

- ω: an inertia weight varying linearly between [0,1];
- $c1$: an acceleration coefficient commonly set with the value 2.0;
- $c2$: an acceleration coefficient commonly set with the value 2.0;
- $R1$: a random number within the interval [0,1];
- $R2$: a random number within the interval [0,1];
- $\vec{X}_j(t)$: the position of seabird i at the time t;
- $\vec{V}_j(t)$: the velocity of seabird j at the current time t, which is within the range $[\overline{V}, \underline{V}]$, where the $\overline{V}$ and the $\underline{V}$ indicate the allowed maximum and minimum velocities, respectively;
- $\vec{P}_j(t)$: the personal best position of the seabird j;
- $\overrightarrow{X_{s^*}}(t)$: the position of the global best seabird s^*;

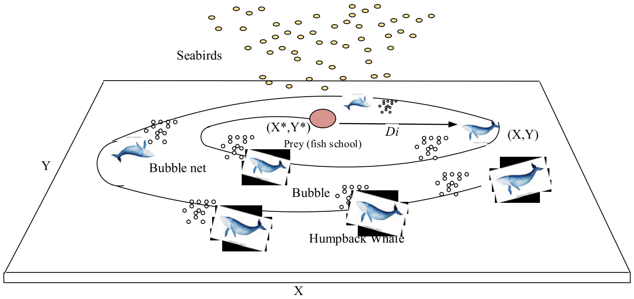


FIGURE 5. The collaborative hunting of humpback whales and seabirds.

The next position of the seabird j is defined by Equation (20).

$$\vec{X}_j(t+1) = \vec{X}_j(t) + \vec{V}_j(t+1) \tag{20}$$

Algorithm 2 shows the main logic flow of PSO. Line 2 sets the parameter values for PSO, where Ps is the population number of particles/seabirds, N is the total number of storage/retrieval requests, T is the total number of iterations. Line 3 generates a block of storage and retrieval requests for an experiment. Line 4 initializes the positions of seabirds. Line 5 sets the iteration counter $t = 1$. Line 6 evaluates the F(Xj) for each seabird j and finds the $\overrightarrow{X_{s^*}}(t)$. Line 7 initializes the population counter $j = 1$.

3) HYBRID MODELS OF WOA AND PSO

Fig. 5 depicts a hunting model including whales and seabirds. This idea originates from nature. It is observed that while humpback whales are attacking the fish school in the water, seabirds are simultaneously attacking it from the air. The seabirds are more capable of spotting the fish school from the air. Together humpback whales and seabirds hunting simultaneously is natural. This is also advantageous due to more populations joining the hunting. In this research, the WOA and PSO are used to model the hunting behavior of humpback whales and seabirds, respectively.

Three versions of hybrid models, namely Hybrid1 (WOA+PSO), Hybrid2 (WOA+PSO), and Hybrid3 (WOA+PSO), were developed in this research. They are detailed as follows.

a: Hybrid1 (WOA+PSO)

Algorithm 3 shows the main logic flow of Hybrid1 (WOA+PSO). Line 2 sets the parameter values for WOA. Line 3 generates the block of storage and retrieval request. Line 4 sets parameter values for PSO. Lines 5 and 6 initialize the positions of whales and seabirds, respectively. Lines 7 and 8 evaluate F(Xi) and F(Xj). Line 9 sets the iteration counter $t = 1$. Line 11 calls PSO and finds the global best seabird s^*. Line 12 updates $\vec{w}^*$ to $\vec{s}^*$ if the $\vec{s}^*$ is better. This enables the $\vec{w}^*$ to be guided by the $\vec{s}^*$ while other whales remain guided by the global best $\vec{w}^*$. Hybrid1 (WOA+PSO) uses the ROVs to transform position vectors from real numbers into ranking numbers of a sequence.

b: Hybrid2 (WOA+PSO)

Algorithm 4 shows the logic of Hybrid2 (WOA+PSO). It is similar to **Algorithm 3**, except that Lines 20, 24, and 27 use a Nearest-Neighborhood (NN) heuristic to match a retrieval request with a storage request. The NN heuristic uses a distance table which stores the distance data of SKUs. This table is generated in Line 4, which uses the SKU storage position data generated in Line 3. The ROV technique is also used.

c: Hybrid3 (WOA+PSO)

Hybrid3 (WOA+PSO) aims to improve Hybrid2 (WOA+PSO). This is a further improvement by allowing whales to use adaptive movements. Each of the movements refers to the $\overrightarrow{w^*}$. The following steps are used.

1) **Measuring the Hamming distance:** the first step is measuring the Hamming distance between the whale X and the w^* using Equation (21).

$$\mathrm{HD}(\overrightarrow{w^*}, \vec{X}) = \sum\nolimits_{i=1}^{D} \mathrm{XOR}(\overrightarrow{w^*_i}, \vec{X}_i) \tag{21}$$

This formula counts the total number of different elements between the two vectors $\overrightarrow{w^*}$ and $\vec{X}$ of the w^* and whale X, respectively. The i indicates the i-th element, while the D indicates the dimension of the position vector. The XOR is an operator which works as follows:

$$\mathrm{XOR}(\overrightarrow{w^*_i}, \vec{X}_i) = \begin{cases} 1, & \text{if } \overrightarrow{w^*_i} \neq \vec{X}_i \\ 0, & \text{if } \overrightarrow{w^*_i} = \vec{X}_i \end{cases} \tag{22}$$

2) **Decide a change rate (CR):** for each element i in the $\vec{X}$(t), a change rate (CR) is used to determine the change of the i-th element in $\vec{X}_i$(t+1) to become the i-th element in the $\overrightarrow{w^*}(t)$. This simulates approaching the w^*. The CR is defined in Equation (23) to make a whale move adaptively.

$$\mathrm{CR} = \mathrm{HD}(\vec{w}^*, \vec{X})/D \tag{23}$$

3) **Determine the next position of a whale:** this step moves whale X to its next position. The i-th element of the next position vector is defined by Equation (24).

$$\vec{X}_i(t+1) = \begin{cases} \overrightarrow{w^*_i}(t), & \text{if } \vec{X}_i(t) \neq \overrightarrow{w^*_i}(t) \\ & \text{and } R_i < CR \\ \vec{X}_i(t), & \text{if } \vec{X}_i(t) \neq \overrightarrow{w^*_i}(t) \\ & \text{and } R_i \geq 0 \\ \vec{X}_i(t), & \text{if } \vec{X}_i(t) = \overrightarrow{w^*_i}(t) \end{cases} \tag{24}$$

TABLE 1. The features of different approaches.

Approach	ROV	NN	Adaptive movement
GA	v		
WOA	v	-	-
PSO	v	-	-
PWOA	v		
Hybrid1 (WOA+PSO)	v	-	-
Hybrid2 (WOA+PSO)	v	v	-
Hybrid3 (WOA+PSO)	v	v	v

-: not used; NN: Nearest Neighborhood.

TABLE 2. The parameter setting of different approaches.

Parameter	Approach						
	GA	PWOA	WOA	PSO	Hybrid1 (WOA+ PSO)	Hybrid2 (WOA+ PSO)	Hybrid3 (WOA+ PSO)
N	20,40, 80,160	20,40, 80,160	20,40, 80,160	20,40, 80,160	20,40, 80,160	20,40, 80,160	20,40, 80,160
Pw	60	60	60	-	60	60	60
Ps	-	60	-	60	60	60	60
I	500	500	500	500	500	500	500
b	-	1	0.5/1	-	0.5/1	0.5/1	0.5/1
ω	-	-	-	0.5	0.5	0.5	0.5
$\overline{V}$	-	-	-	2	-	-	-
$\underline{V}$	-	-	-	2	-	-	-
Rm	0.3	-	-	-	-	-	-
Rc	0.4	-	-	-	-	-	-

The R_i is a random number in [0,1], which helps determine the i-th element in $\vec{X}_i(t+1)$. Equation (23) enables a higher probability for a farther whale to approach the w^* quickly but slowly for a near whale. If the $\overrightarrow{w^*_i}(t)$ replaces the $\vec{X}_i(t)$ in the $\vec{X}_i(t+1)$, the $\vec{X}_i(t)$ then takes the place of the original $\overrightarrow{w^*_i}(t)$. This ensures the feasibility of the next position.

For example, given $\vec{X}_i(t) = [5,4,3,2,1]$ and $\overrightarrow{w^*_i}(t) = [3,4,2,5,1]$, we derive $\mathrm{HD}(\overrightarrow{w^*}, \vec{X}) = 2$, $\mathrm{CR} = 2/5 = 0.4$. Given the random number $R_1 = 0.5$, then $\vec{X}_1(t+1)$ becomes 3, and the 5 in $\vec{X}_i(t)$ takes the position of 3 in $\vec{X}_i(t)$ and appears in $\vec{X}_i(t+1) = [3,4,5,2,1]$. The next position remains feasible due to being a combination of a sequence with no repetition of numbers.

The main logic flow of Hybrid3 (WOA+PSO) is similar to **Algorithm 4**, except for that Lines 18, 22, and 25 use Equations (21)–(24) to move a whale adaptively.

Table 1 shows the techniques employed for different approaches.

D. SIMULATION MODEL

A timed Predicate/Transition Net (Fig. 6) model is developed. It provides discrete-event simulation to evaluate solutions to the DCBSP. The main components of this net include Predicate, Transition, and Directed Arc [37], [38], which are defined as follows.

- Predicate = {Storage_request, Retrieval_request, Dual_Command_task, S/R machine, Execution, Closed};

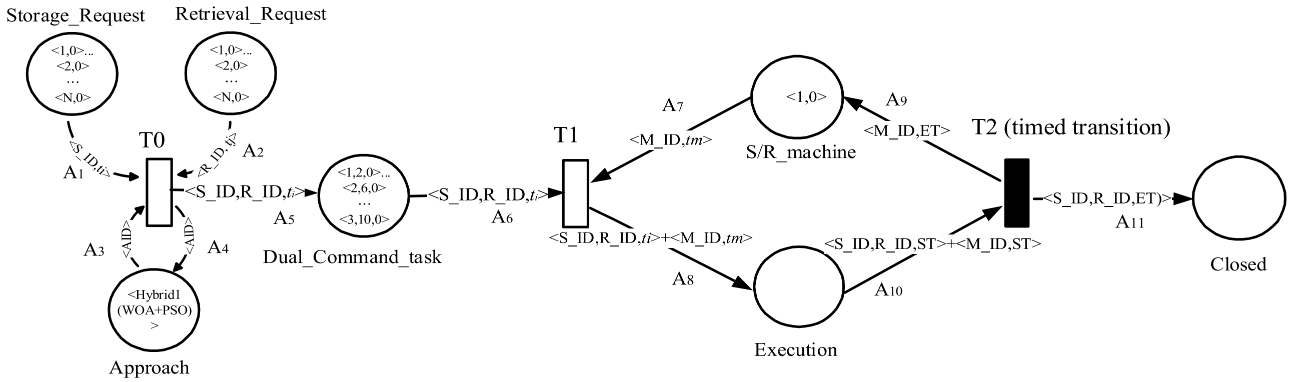


FIGURE 6. A timed predicate/transition net model for discrete-event simulation.

- Transition = {T0,T1,T2};
- Directed Arc = {A1, . . . , A11};

Together with a circle symbol, a predicate asserts a fact. For example, once a token < *i*, *ai*> appears in the circle symbol of Predicate **Storage_request**, it asserts the existence of a storage request. Once a token <*i,j*, *ai*> appears in its circle symbol of Predicate **Dual_commnd_task**, it asserts the existence of a dual command. The token's attributes show the information of storage_request_id, retrieval_request_id, and the available time of the dual command. Once a token <M_id, a_m > appears in the circle symbol of Predicate **S/R_machine**, it asserts the existence of an S/R machine. The token's attributes show the information of machine ID and available time of this machine. Once a token appears in the circle symbol of the predicate **Closed**, it asserts the existence of a completed task. T0, T1, and T2 are Transitions with a bar symbol. When tokens appear in the predicates **Storage_request**, **Retrieval_request**, and **Approach** simultaneously, it enables Transition T0 to fire. After this firing, a dual_command_task token is formed and to be transitioned to the predicate **Dual_Command_task**. When the predicates **Dual_Command_task** and **S/R_machine** are true simultaneously, it enables Transition T1 to fire. After this firing, it forms a token <S_ID,R_ID, *ai*> + <M_ID, a_m > which will flow to Predicate **Execution**. Meanwhile, it enables Transition T2 to fire. Note that T2 is a timed transition that takes time to complete. After firing T2, the <M_ID,CT> token flows back to Predicate **S/R_machine** while the token <*i,j*, CT) goes to Predicate **Closed**. When all tokens stay with Predicate **Closed**, it terminates the simulation.

Given a_i and a_m as the available times of the storage request *i* and the S/R machine *m*. Equation (25) defines the start event-time to handle this dual command.

$$\mathrm{BT_i}= \mathrm{Max}\,\{a_i, a_m\} \tag{25}$$

Given the $OT(i, j)$ as the operational time to complete the *i*-th dual command, Equation (26) defines the end-event time of this dual command.

$$\begin{aligned}\mathrm{CT_i} &= \mathrm{BT_i}+\mathrm{OT}(i, j)\\ &= \mathrm{Max}\,\{a_i, a_m\} + \mathrm{OT}(i, j)\end{aligned} \tag{26}$$

V. NUMERICAL EXPERIMENTS

In this section, the effectiveness of the three hybridizations, Hybrid1 (WOA+PSO), Hybrid2 (WOA+PSO), and Hybrid3 (WOA+PSO,) are investigated through experiments. First, a small-sized experiment is used to illustrate the obtained results. Then, extensive experiments were conducted to compare with standard WOA, PSO, GA, and the PWOA proposed in [7]. These experiments are performed in a computer with Intel PENTIUM CPU 2117U (64bits and 1.8GHz) and 4GB DRAMs. Java is used as the programming language.

A. PARAMETER VALUES SET FOR THE AS/RS

For the S/R machine, the horizontal moving speed is set $v_h = 5$ (M/s), the vertical moving speed is set $v_v = 1$ (M/s), and the shuttle moving speed is set $v_d = 5$ (M/s). For a rack, the total number of columns is X = 40; the number of tiers is Y = 30. Each rack has a total number of 40 × 30 storage cells. The dimensions of a storage cell are $W = 1.5$ M, $H = 1.75$ M, and $D = 1.5$ M. In addition, the I/O stations on the 1st, 2nd, 3rd, 4th, and 5th floors are (0,1), (0,4), (0,7), (0,10), and (0,13), respectively. The settings of these parameter values are based on the physical data of the real AS/RS studied in this research.

B. PARAMETER VALUES SETTING FOR VARIOUS APPROACHES

Table 2 shows the parameter values set for different approaches. The following variables are used: *N* indicates the total number of storage/retrieval requests; *Pw* indicates the total number of whale populations; *Ps* indicates the total number of seabird populations; *I* indicates the total number of iterations; *b* is a constant defining the shape of the logarithmic spiral; ω indicates an inertia weight of particle; $\overline{V}$ indicates an upper limit of velocity; $\underline{V}$ indicates a lower limit of velocity; Rm indicates a mutation rate of the GA; Rc indicates a crossover rate of the GA; -: not used.

For fair comparison, the *Pw* and *Ps* are set to 60, an acceptable group size of seabirds and whales. The $I = 500$ iterations are used, the same number used in [7]. The $\overline{V} = \underline{V} = 2$ are common values set as the upper and lower speed limits

TABLE 3. Storage requests.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x	12	18	34	6	20	27	36	25	31	5	27	10	26	37	3	36	13	19	27	32
y	8	6	2	7	7	10	7	6	7	11	3	4	14	12	10	3	3	12	14	12
TP	2	2	3	1	1	2	1	1	1	2	2	2	2	1	2	3	3	2	1	2
OID	3	4	5	3	3	2	3	4	3	2	5	4	1	2	2	5	5	2	1	2
LF	3	2	1	3	3	4	3	2	3	4	1	2	5	4	4	1	1	4	5	4

TABLE 4. Retrieval requests.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x	30	33	12	36	23	23	9	13	39	28	35	3	36	17	24	15	31	25	15	38
y	8	12	8	3	1	2	2	1	11	15	9	6	15	13	1	1	7	13	12	15
TP	2	2	1	2	3	3	2	3	1	2	2	2	1	2	3	3	2	2	2	2
OID	3	2	3	5	5	5	5	5	2	1	3	4	1	1	5	5	3	1	2	1
ULF	3	4	3	1	1	1	1	1	4	5	3	2	5	5	1	1	3	5	4	5

for a particle/seabird, the same values used in [7]. The $b = 0.5$ and $b = 1$ are used. The $b = 0.5$ is used for Hybrid1 (WOA+PSO), Hybrid2 (WOA+PSO), and Hybrid3 (WOA+PSO) in comparison to WOA and PSO. This setting aims to avoid an excessive distortion of a spiral shape. However, this parameter value is changed to $b = 1$ compared to PWOA, which used $b = 1$ in [7]. Rm = 0.3 and Rc = 0.4 are set for the GA due to a good performance in the test experiments. The ω is within the range [0,1]. Thus, $\omega = 0.5$ is used in this research to avoid bias. The use of $\omega = 0$ will lead to no existence of an inertia momentum; the use of $\omega = 1$ will lead to an excessive inertia momentum. Due to stable performance in the test experiments, the above parameter settings were used in this research.

C. GENERATION OF EXPERIMENTAL INSTANCES

Based on Table 2, random storage and retrieval requests were generated by a computer automatically. It simulates random service requests arising from the shop floor. In each experiment, the numbers of N storage and retrieval requests are generated with different types of SKUs and storage positions. A problem size is defined as $N \times N$.

D. AN EXAMPLE OF A SMALL INSTANCE

This section demonstrates the best solutions found by different PSO, WOA, Hybrid1 (WOA+PSO), Hybrid2 (WOA+PSO), and Hybrid3 (WOA+PSO) for a small instance 20 × 20.

Tables 3 and 4 show the storage and retrieval requests generated by the computer automatically. The i indicates a storage request ID; the j indicates a retrieval request ID; the x indicates a column number; the y indicates a tier number; the TP indicates a material type; the OID indicates an operation ID; the LF indicates a floor number whose I/O station is used for loading a storage SKU; the ULF indicates a floor whose I/O station is used for unloading a retrieved SKU. The distance table of each pair of i and j is shown in Appendix A.

For this experimental instance, Tables 5-9 show the best solution found by WOA, PSO, Hybrid1 (WOA+PSO), Hybrid2 (WOA+PSO), and Hybrid3 (WOA+PSO), respectively. The terms are defined as follows: i: storage r request ID; j retrieval request ID; T0: travel time to the I/O station; T1: travel time to the storage location; ts: storage shuttle time; T2: travel time from the storage location to the retrieval location; Tr: retrieval shuttle time; T3: travel time from the retrieval location to the I/O station; OT(i,j): operational time of the dual commands; AOT: accumulated total operational time. The results are summarized in Table 10.

Hybrid3 (WOA+POS) finds the best solution with $Z = 531.3$ (s) at the cost of 12.2 (s) CPU time. The PSO finds the worst solution with $Z = 624.8$ (s) at the cost of 4.07 (s) of CPU time.

E. EXTENSIVE EXPERIMENTS OF DIFFERENT PROBLEM SIZES

1) COMPARING TO THE BASE MODELS WOA AND PSO

Extensive experiments of different problem sizes (20 × 20, 40 × 40, 80 × 80, 160 × 160) have been conducted to examine the effectiveness of different approaches. Table 11 shows the experimental results obtained from PSO, WOA, Hybrid1 (WOA+PSO), Hybrid2 (WOA+PSO), and Hybrid3 (WOA+PSO) under different problem sizes.

The results are summarized as follows:

1) At the problem size 20 × 20, Hybrid3 (WOA+PSO) outperforms Hybrid2 (WOA+PSO), and Hybrid1 (WOA+PSO), WOA, and PSO by average edges of 4.8%, 13.0%, 18.8%, and 20.6%, respectively.
2) At the problem size 40 × 40, Hybrid3 (WOA+PSO) outperforms Hybrid2 (WOA+PSO), Hybrid1 (WOA+PSO), WOA, and PSO by average edges of 1.7%, 15.7%, 23.7%, and 25.2%, respectively.
3) At the problem size 80 × 80, Hybrid3 (WOA+PSO) outperforms Hybrid2 (WOA+PSO), Hybrid1 (WOA+PSO), WOA, and PSO by average edges of 3.6%, 21.3%, 26.3%, and 26.2%, respectively.
4) At the problem size 160 × 160, Hybrid3 (WOA+PSO) outperforms Hybrid2 (WOA+PSO), Hybrid1 (WOA+PSO), WOA, and PSO by average edges of 0.8%, 26.6%, 29.5%, and 27.9%, respectively.
5) Hybrid3 (WOA+PSO) outperforms the other approaches.

2) COMPARING TO THE PWOA AND GA

In addition to WOA and PSO, the three hybridizations have been further compared to GA and PWOA in this subsection. The GA is a traditional algorithm which uses Rotulet wheel selection, crossover, and one-point mutation operations for genes. The PWOA was proposed by Bansal and Aggarwal [7]. The algorithm of the PWOA is shown in **Algorithm 5** (Appendix B). Given that max_iteration is the total number of iterations, in the PWOA, the PSO first uses max_iteration/2 to find the global best particle p^*. Then, the position of the p^* is set for the global best whale (w^*), and the WOA starts to find the global best solution in another

TABLE 5. The best solutions found by WOA (20 × 20).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
(i,j)	(11,19)	(15,6)	(10,3)	(12,17)	(6,8)	(17,5)	(3,12)	(14,1)	(1,4)	(5,18)	(2,7)	(20,2)	(13,9)	(4,20)	(8,16)	(16,15)	(18,13)	(19,10)	(9,11)	(7,14)
T0	0	0	15.8	5.2	5.2	0	0	10.5	0	10.5	15.8	15.8	5.2	5.2	15.8	0	15.8	0	10.5	0
T1	8.1	0.9	1.8	3	8.1	3.9	10.2	11.1	3.6	6	5.4	9.6	7.8	1.8	7.5	10.8	5.7	8.1	9.3	10.8
Ts	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
T2	15.8	14	5.2	6.3	15.8	3.5	9.3	7	8.8	10.5	7	0.3	5.2	14	8.8	3.6	5.2	1.8	3.5	10.5
Tr	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
T3	4.5	6.9	3.6	9.3	3.9	6.9	1.8	9	10.8	7.5	3.5	9.9	11.7	11.4	4.5	7.2	10.8	8.4	10.5	5.1
OT(i,j)	29.6	23	27.6	25	34.2	15.5	22.5	38.8	24.4	35.7	32.9	36.8	31.2	33.6	37.7	22.8	38.7	19.4	35	27.6
AOT	29.6	52.5	80.1	105.1	139.3	154.8	177.3	216.1	240.5	276.2	309	345.8	377	410.6	448.3	471.1	509.8	529.2	564.2	591.9

TABLE 6. The best solutions found by PSO (20 × 20).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
(i,j)	(10,5)	(12,13)	(7,4)	(5,17)	(14,2)	(15,10)	(8,11)	(13,12)	(18,19)	(6,3)	(9,7)	(17,6)	(11,20)	(19,1)	(2,18)	(4,9)	(3,16)	(1,14)	(20,15)	(16,8)
T0	15.8	5.2	10.5	10.5	5.2	0	15.8	10.5	10.5	0	0	0	0	0	5.2	10.5	15.8	10.5	5.2	0
T1	1.8	3	10.8	6	11.1	0.9	7.5	7.8	5.7	8.1	9.3	3.9	8.1	8.1	5.4	1.8	10.2	3.6	9.6	10.8
Ts	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
T2	17.5	19.2	7	3.3	1.2	8.8	5.2	14	1.2	4.5	8.8	3	21	10.5	12.2	9.9	5.7	8.8	19.2	6.9
Tr	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
T3	6.9	10.8	10.8	9.3	9.9	8.4	10.5	3.5	4.5	3.6	2.7	6.9	11.4	9	7.5	11.7	4.5	5.1	7.2	3.9
OT(i,j)	43.1	39.5	40.3	30.3	28.7	19.2	40.2	37	23.1	17.4	21.9	15	41.7	28.8	31.6	35.1	37.4	29.1	42.5	22.8
AOT	43.1	82.6	122.9	153.2	181.9	201.1	241.3	278.3	301.4	318.8	340.8	355.8	397.4	426.2	457.9	493	530.3	559.5	602	624.8

TABLE 7. The best solutions found by hybrid1 (WOA+PSO) (20 × 20).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
(i,j)	(3,11)	(8,8)	(9,10)	(13,5)	(11,7)	(16,16)	(17,17)	(6,18)	(19,12)	(2,4)	(10,2)	(20,3)	(1,20)	(14,6)	(15,19)	(7,15)	(12,9)	(18,14)	(5,1)	(4,13)
T0	0	5.2	10.5	0	0	0	0	5.2	0	0	15.8	0	0	5.2	15.8	5.2	5.2	0	10.5	0
T1	10.2	7.5	9.3	7.8	8.1	10.8	3.9	8.1	8.1	5.4	1.8	9.6	3.6	11.1	0.9	10.8	3	5.7	6	1.8
Ts	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
T2	12.2	8.8	14	22.8	5.4	6.3	7	5.2	14	5.4	8.4	7	12.2	17.5	3.6	10.5	12.2	1.8	3	14
Tr	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
T3	10.5	3.9	8.4	6.9	3.5	4.5	9.3	7.5	3.5	10.8	9.9	3.6	11.4	6.9	4.5	7.2	11.7	5.1	9	10.8
OT(i,j)	34.1	26.6	43.4	38.6	18.2	22.8	21.4	27.3	26.8	22.8	37	21.4	28.5	42	26	35	33.4	13.8	29.7	27.8
AOT	34.1	60.8	104.2	142.8	161	183.8	205.2	232.5	259.3	282.1	319.1	340.5	368.9	410.9	436.8	471.8	505.2	518.9	548.6	576.4

TABLE 8. The best solutions found by the hybrid2 (WOA+PSO) (20 × 20).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
(i,j)	(4,12)	(18,14)	(19,10)	(10,3)	(9,17)	(7,11)	(13,18)	(6,1)	(1,19)	(12,7)	(20,2)	(17,8)	(5,6)	(8,15)	(3,4)	(2,16)	(16,9)	(15,5)	(11,13)	(14,20)
T0	10.5	10.5	0	5.2	0	0	10.5	5.2	0	10.5	15.8	15.8	10.5	5.2	0	5.2	0	0	0	5.2
T1	1.8	5.7	8.1	1.8	9.3	10.8	7.8	8.1	3.6	3	9.6	3.9	6	7.5	10.2	5.4	10.8	0.9	8.1	11.1
Ts	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
T2	1.8	1.8	1.8	5.2	0	3.5	1.8	3.5	7	3.5	0.3	3.5	8.8	8.8	1.8	8.8	14	15.8	21	5.2
Tr	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
T3	3.5	5.1	8.4	3.6	9.3	10.5	7.5	9	4.5	3.5	9.9	3.9	6.9	7.2	10.8	4.5	11.7	6.9	10.8	11.4
OT(i,j)	18.8	24.2	19.4	17.1	19.8	26	28.8	27.1	16.3	21.7	36.8	28.2	33.4	29.9	23.9	25.1	37.7	24.8	41.1	34.2
AOT	18.8	43	62.5	79.5	99.3	125.3	154.1	181.1	197.4	219.1	255.9	284.1	317.5	347.4	371.3	396.4	434.1	458.9	500	534.2

TABLE 9. The best solutions found by the hybrid3 (WOA+PSO) (20 × 20).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
(i,j)	(15,12)	(17,8)	(12,7)	(3,4)	(4,3)	(5,6)	(18,14)	(14,9)	(19,10)	(7,11)	(9,17)	(1,19)	(13,18)	(16,1)	(10,16)	(11,15)	(8,5)	(2,2)	(6,13)	(20,20)
T0	15.8	5.2	5.2	0	10.5	0	15.8	5.2	5.2	10.5	0	0	5.2	21	5.2	0	5.2	5.2	0	5.2
T1	0.9	3.9	3	10.2	1.8	6	5.7	11.1	8.1	10.8	9.3	3.6	7.8	10.8	1.8	8.1	7.5	5.4	8.1	9.6
Ts	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
T2	7	3.5	3.5	1.8	1.8	8.8	1.8	1.8	1.8	3.5	0	7	1.8	8.8	17.5	3.5	8.8	10.5	8.8	5.2
Tr	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
T3	0.9	3.9	3.5	10.8	3.6	6.9	5.1	11.7	8.4	10.5	9.3	4.5	7.5	9	4.5	7.2	6.9	9.9	10.8	11.4
OT(i,j)	25.8	17.8	16.4	23.9	18.9	22.9	29.5	31	24.7	36.5	19.8	16.3	23.5	50.8	30.2	20	29.6	32.2	28.9	32.7
AOT	25.8	43.5	60	83.9	102.8	125.7	155.2	186.2	210.9	247.4	267.2	283.5	307	357.7	387.9	407.9	437.5	469.8	498.6	531.3

max_iteration. The logic of the PWOA is simple and easy to implement. However, the ROV technique is still necessary to make solutions feasible for the DCBSP. Table 12 shows the experimental results obtained from GA, PWOA, Hybrid1 (WOA+PSO), Hybrid2 (WOA+PSO), and Hybrid3 (WOA+PSO) under different problem sizes.

TABLE 10. The comparison of different approaches.

	PSO	WOA	Hybrid1 (WOA+PSO)	Hybrid2 (WOA+PSO)	Hybrid3 (WOA+PSO)
CPU	4.07	3.5	10.3	11.4	12.2
Z	624.8	591.9	576.4	534.2	531.3

The results are summarized as follows.

1) At the problem size 20 × 20, Hybrid3 (WOA+PSO) outperforms Hybrid2 (WOA+PSO), Hybrid1 (WOA+PSO), PWOA, and GA by average edges of 3.7%, 9.8%, 21.1%, and 30.8%, respectively.
2) At the problem size 40 × 40, Hybrid3 (WOA+PSO) outperforms Hybrid2 (WOA+PSO), Hybrid1 (WOA+PSO), PWOA, and GA by average edges of 3.5%, 20.6%, 24.9%, and 30.2%, respectively.
3) At the problem size 80 × 80, Hybrid3 (WOA+PSO) outperforms Hybrid2 (WOA+PSO), Hybrid1 (WOA+PSO), PWOA, and GA by average edges of 1.9%, 22.8%, 26.5%, and 33.6%, respectively.
4) At the problem size 160 × 160, Hybrid3 (WOA+PSO) outperforms Hybrid2 (WOA+PSO), Hybrid1 (WOA+PSO), PWOA, and GA by average edges of 0.5%, 26.3%, 28.8%, and 331.6%, respectively.
5) Hybrid3 (WOA+PSO) outperforms the other approaches.

F. A SENSITIVITY ANALYSIS OF POPULATION ADVANTAGE IN THE HYBRID APPROACHES

The sensitivity analysis of population advantage in the hybrid approaches is a study to investigate the impact of populations of seabird/whale on the experimental results. The need for this analysis is due to the following reason. In previous experiments, the populations of seabirds and whales in the PSO and WOA are set to 60, respectively. The same population number of seabirds and whales in Hybrid1, Hybrid2, and Hybrid3 leads to a total population of 120. One may argue that this makes unfair comparisons between the hybrid models and their base models (i.e., WOA and PSO). To address this concern, the populations of seabirds and whales in the hybrid approaches are reduced to 30, resulting in a total number of 60 populations, identical to those in the base models. With this setting, these models have been compared again.

Fig. 7(a), 7(b), 7(c), and 7(d) show the Z values obtained from different approaches for the problem sizes of 20 × 20, 40 × 40, 80 × 80, and 160 × 160, respectively. These figures show that these hybrid approaches remain to outperform the standard WOA and PSO, and Hybrid3 (WOA+PSO) remains to be the best one.

G. ANALYSIS AND DISCUSSIONS

1) Table 11 shows that Hybrid3 (WOA+PSO) is better than Hybrid2 (WOA+PSO), Hybrid1 (WOA+PSO), WOA, and PSO. Table 12 shows that Hybrid3 (WOA+PSO) is better than Hybrid2 (WOA+PSO), Hybrid1 (WOA+PSO), PWOA, and GA. In summary, Hybrid3 (WOA+PSO) outperforms the other approaches.
2) Table 1 shows the features of all approaches used in this research. The ROV technique is essential for making solutions to be feasible. Thus, this technique is used by all approaches. The experimental results showed that Hybrid1 (WOA+PSO) outperformed the standard WOA and PSO. Hybrid2 (WOA+PSO) is better than Hybrid1 (WOA+PSO) due to the further use of NN. Hybrid3 (WOA+PSO) is better than Hybird2 (WOA+PSO) due to the further use of adaptive movements for whales. Thus, factors including hybridization, NN, and adaptive movements are considered to be able to empower Hybrid3 (WOA+PSO) and make it to be the best one.
3) In the standard WOA, the whales are only guided by the global best whale w^*. In the standard PSO, the seabirds are only guided by the global best seabird s^*. As a result, their individual uses have a limitation. In our developed hybrid models, the w^* also refers to the s^* to improve its position. As a result, the whales are influenced by both elites from the seabirds and whales. This is why the preliminary Hybrid1(WOA+PSO) can outperform its base model, WOA and PSO). In addition, it can outperform GA and PWOA.
4) The PWOA is a hybridization of PSO and WOA proposed by Bansal and Aggarwa1 [7]. In the PWOA, the PSO is first used to find the p^* (the global best particle) by using 1/2 maximum iteration. After this, the position of the p^* is set for the w^* (the global best whale), and the WOA is subsequently used to find the global best solution by using another maximum iteration. However, in the PWOA, the PSO and WOA are found to have no interaction in the later maximum iteration. This differs from the three hybrid models proposed in this research, in which the PSO interacts with the WOA in each iteration. As a result, the three hybrid models have a high degree and immediate interaction between the PSO and WOA. This is why the three hybrid models can outperform the PWOA.
5) In Tables 11 and 12, some of the experimental results showed that Hybrir3 (WOA+PSO) is inferior to Hybrid2 (WOA+PSO). One of the possible reasons is the uncertainties encountered by experimental instances. For example, if the initial positions of the whales and seabirds are not well generated to better distribute in the solution space. Then, the seabirds and whales may not be able to better approach optimality. Thus, if the Hybrid2 (WOA+PSO) has better initial positions for whales and seabirds, it may still have a chance to outperform the Hybrid3 (WOA+PSO). Thus, if uncertainty exists, there is no guarantee that the best method can always outperform the others in all instances. This is why we need many experiments to have a better investigation. Only an exact approach

TABLE 11. The results of comparisons of the three hybridizations with PSO and WOA with different problem sizes.

	PSO			WOA			Hybrid1 (WOA+PSO)			Hybrid2 (WOA+PSO)			Hybrid3 (WOA+PSO)	
20x20	Z	T	G	Z	T	G	Z	T	G	Z	T	G	Z	T
1	628.8	4.1	25.5	598.3	3.5	19.4	564.1	10.2	12.5	546.4	11.4	9.0	501.2	12.4
2	562.2	3.9	15.2	606.6	3.9	24.3	546.1	11.1	11.9	504.9	11.4	3.5	488.0	11.9
3	584.6	4.1	20.5	559.6	3.5	15.3	508.3	10.4	4.7	510.7	11.2	5.2	485.3	12.5
4	564.8	4.1	21.3	558.9	3.4	20.0	533.5	10.4	14.5	520.9	11.4	11.8	465.8	11.3
5	558.1	4.2	6.8	592.0	3.5	13.3	577.3	10.0	10.5	537.3	11.4	2.8	522.6	13.0
6	604.3	4.1	26.1	575.2	3.5	20.0	506.0	9.8	5.6	513.6	11.0	7.2	479.2	12.0
7	525.4	4.2	23.0	547.7	3.2	28.3	538.9	10.7	26.2	434.2	11.4	1.7	427.0	12.3
8	578.0	4.1	13.6	557.3	3.5	9.5	586.8	10.4	15.3	522.0	11.7	2.6	508.9	12.8
9	613.5	4.0	22.4	574.4	3.5	14.6	574.4	9.9	14.6	500.1	11.4	-0.2	501.2	11.9
10	675.3	3.9	31.3	635.1	3.5	23.4	585.7	10.0	13.8	535.5	11.3	4.1	514.5	12.0
Avg.	589.5	4.1	20.6	580.5	3.5	18.8	552.1	10.3	13.0	512.6	11.4	4.8	489.4	12.2
40x40														
1	1166.4	8.1	16.3	1212.0	8.1	20.8	1158.1	25.6	15.5	986.2	28.2	-1.7	1003.0	29.0
2	1199.2	8.0	18.9	1210.5	7.8	20.0	1116.2	25.4	10.7	1053.9	29.5	4.5	1008.6	29.8
3	1277.3	8.1	30.5	1256.9	8.4	28.4	1096.7	25.1	12.1	998.2	29.3	2.0	978.7	30.0
4	1221.1	8.4	22.2	1180.2	7.8	18.1	1151.8	25.8	15.3	981.6	29.4	-1.8	999.1	30.1
5	1154.7	8.1	31.2	1141.6	7.8	29.7	1171.1	25.4	33.1	945.2	29.2	7.4	879.9	30.6
6	1248.0	8.1	22.3	1215.6	7.8	19.1	1112.0	25.3	8.9	1027.9	29.5	0.7	1020.7	30.4
7	1279.8	7.8	22.4	1271.0	7.9	21.5	1206.8	25.0	15.4	1043.6	29.4	-0.2	1045.9	31.1
8	1234.8	8.1	29.2	1229.9	7.8	28.7	1203.6	26.6	25.9	985.5	29.5	3.1	955.9	30.3
9	1258.3	7.9	30.1	1222.0	8.8	26.3	1107.1	25.0	14.4	988.6	29.8	2.2	967.5	30.3
10	1270.5	8.0	28.8	1230.0	7.8	24.7	1046.2	25.3	6.1	990.4	26.6	0.4	986.4	29.2
Avg.	1231.0	8.1	25.2	1217.0	8.0	23.7	1137.0	25.5	15.7	1000.1	29.0	1.7	984.6	30.1
80x80														
1	2369.6	13.3	24.5	2440.0	11.7	28.2	2291.1	58.4	20.4	1939.2	69.5	1.9	1902.7	71.2
2	2461.9	13.0	25.2	2381.2	12.5	21.1	2237.1	58.3	13.8	1967.8	70.5	0.1	1966.0	72.3
3	2464.0	14.1	25.9	2336.4	10.8	19.4	2411.7	59.3	23.2	1914.2	73.0	-2.2	1957.2	76.0
4	2395.5	13.3	24.9	2409.5	11.7	25.7	2381.1	58.1	24.2	1985.3	73.1	3.5	1917.3	75.3
5	2453.8	12.9	25.8	2502.3	11.1	28.3	2340.4	58.2	20.0	2017.0	71.8	3.4	1950.9	73.8
6	2436.4	13.3	27.9	2462.0	11.7	29.2	2375.1	58.2	24.7	1947.5	70.1	2.2	1905.1	73.5
7	2491.6	14.9	27.8	2510.9	11.7	28.8	2348.2	58.9	20.4	1985.7	72.1	1.8	1950.0	71.0
8	2516.1	13.3	30.3	2458.2	11.0	27.3	2439.1	58.3	26.3	2363.4	70.2	22.4	1931.1	72.7
9	2445.8	14.2	25.0	2439.4	11.7	24.7	2094.3	55.9	7.0	1975.9	69.8	1.0	1956.7	73.2
10	2351.2	13.3	25.1	2454.7	11.7	30.6	2499.0	56.8	32.9	1905.7	69.1	1.4	1879.7	72.4
Avg.	2438.6	13.3	26.2	2439.5	11.6	26.3	2341.7	58.0	21.3	2000.2	70.9	3.6	1931.7	73.1
160x160														
1	4805.3	17.2	29.2	4900.1	25.3	31.7	4923.5	142.0	32.3	3761.2	169.1	1.1	3720.6	175.0
2	4868.8	16.8	26.5	4911.8	25.3	27.6	4840.2	135.5	25.7	3879.1	174.2	0.8	3849.8	178.5
3	5090.0	17.2	29.7	5163.1	25.1	31.6	4765.5	134.7	21.4	3920.3	175.6	-0.1	3924.2	176.2
4	4798.4	17.2	30.4	4687.0	25.3	27.3	5072.8	133.9	37.8	3687.4	177.2	0.2	3680.5	174.3
5	4892.4	17.5	31.4	4906.9	25.3	31.8	4755.7	133.6	27.8	3750.9	177.8	0.8	3721.9	183.7
6	5024.9	17.2	27.2	5031.9	24.8	27.4	4769.6	132.1	20.7	3968.9	168.8	0.5	3950.9	163.7
7	4911.3	17.2	25.2	5131.5	25.3	30.9	4730.7	134.1	20.6	3972.6	173.6	1.3	3921.4	179.0
8	4916.2	16.2	27.0	5000.1	25.0	29.1	4867.0	134.7	25.7	3974.9	174.6	2.7	3872.1	182.6
9	4995.4	17.2	30.4	5031.2	25.3	31.3	4883.1	132.1	27.4	3850.9	173.3	0.5	3831.6	168.2
10	4809.4	17.1	22.6	4948.2	24.4	26.1	4967.1	133.7	26.6	3944.8	173.3	0.5	3923.8	166.0
Avg.	4911.2	17.1	27.9	4971.2	25.1	29.5	4857.5	134.6	26.6	3871.1	173.8	0.8	3839.7	174.7

Z: the objective function value; T: time (s); G: gap to the Hybrid3 (WOA+PSO)

can make this guarantee as it aims to find the optimum. Metaheuristics, including the hybridizations developed in this research, are not the kind of exact approach. Nevertheless, on average, Hybrid3 (WOA+PSO) can outperform Hybrid 2 (WOA+PSO).

6) From Algorithms 1, we know the WOA mainly contains two REPEAT loops. The first loop contains the number of I iterations, and the second loop contains the number of Pw iterations. The second loop also needs to handle the N elements of a solution. Thus, the computational complexity of WOA is O($I \times Pw \times N$). Similarly, the PSO contains two REPEAT loops, and the second loop also has to handle N elements of a solution. Thus, the computational complexity of PSO is O($I \times Ps \times N$). The three hybrid models also contain two REPEAT loops. The first loop contains the number of I iterations, and the second loop contains the number of Pw iterations. The second loop has to handle the N elements of a solution. In addition, the PSO( ) is called in the first loop. Thus, the computational complexity of these hybridizations is O(($I \times Ps \times N) \times Pw \times N$). From Algorithm 5, we know PWOA first runs a half maximum iteration of PSO, followed by another maximum iteration of WOA. Thus, the computational complexity of PWOA is O(($I/2 \times Ps \times N$)+($I \times Pw \times N$)). The

TABLE 12. The results of comparisons of the three hybridizations with GA and PWOA with different problem sizes.

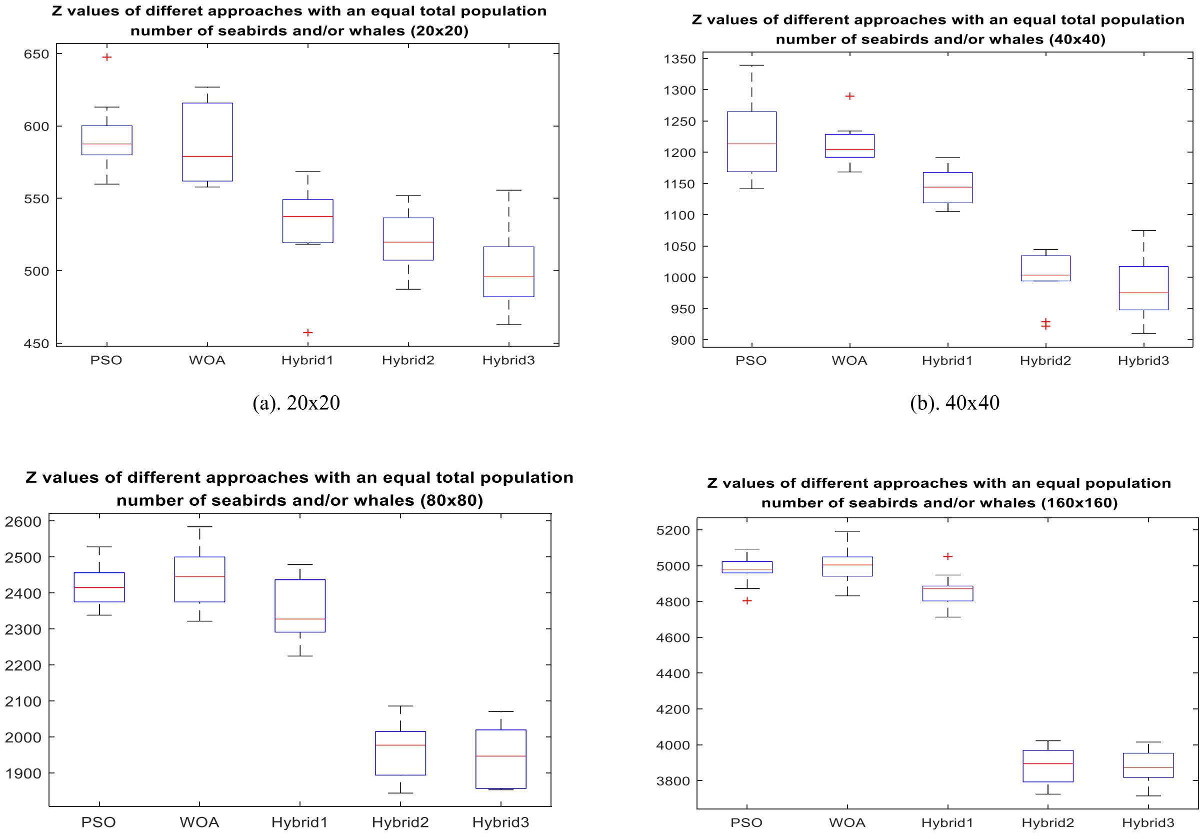
	GA			PWOA			Hybrid1 (WOA+PSO)			Hybrid2 (WOA+PSO)			Hybrid3 (WOA+PSO)	
20x20	Z	T	G	Z	T	G	Z	T	G	Z	T	G	Z	T
1	644.1	3.8	34.9	597.7	11.4	25.2	533.7	10.2	11.8	482.8	11.5	1.1	477.4	13.0
2	625.2	4.0	25.6	552.5	11.2	11.0	538.4	11.9	11.8	498.7	12.3	0.2	497.6	12.3
3	697.1	4.2	38.9	643.7	11.5	28.3	563.8	11.3	12.5	554.5	12.3	10.5	501.7	13.1
4	550.4	4.3	16.6	566.5	11.4	20.0	504.6	11.2	6.9	483.7	12.4	2.4	472.1	13.2
5	657.9	4.0	26.3	596.1	11.3	14.5	567.9	11.4	9.1	525.0	12.3	0.8	520.7	13.1
6	647.1	3.7	29.9	607.3	11.3	21.9	529.9	11.8	6.4	520.4	12.2	4.5	498.1	12.5
7	649.8	4.4	40.4	602.9	11.3	30.3	525.5	11.4	13.6	463.6	12.6	0.2	462.7	13.1
8	683.5	4.4	30.2	603.9	11.2	15.0	540.8	10.8	3.0	530.2	12.3	1.0	525.1	12.8
9	538.9	4.7	36.6	493.7	11.3	25.2	472.4	11.9	19.8	452.2	12.3	14.7	394.4	13.3
10	623.3	3.8	28.7	580.2	11.5	19.8	497.6	11.6	2.7	489.7	12.4	1.1	484.3	13.1
Avg.	631.7	3.9	30.8	584.4	11.3	21.1	527.5	11.4	9.8	500.1	12.3	3.7	483.4	13.0
40x40														
1	1285.8	8.4	28.6	1300.3	24.9	30.0	1220.0	25.2	22.0	1019.0	29.3	1.9	1000.2	31.1
2	1250.8	8.2	33.7	1250.0	23.1	33.6	1153.6	25.3	23.3	954.1	28.7	2.0	935.7	31.4
3	1242.5	7.8	36.1	1070.8	23.2	17.3	1070.9	25.2	17.3	912.6	29.1	0.0	912.6	31.0
4	1242.1	8.3	30.2	1159.7	22.7	21.5	1134.1	25.4	18.9	957.2	28.7	0.3	954.2	31.2
5	1322.4	8.4	33.9	1256.4	23.6	27.3	1199.6	25.2	21.5	1026.7	29.5	4.0	987.3	30.8
6	1210.2	8.5	32.0	1114.3	24.7	21.6	1074.3	25.5	17.2	967.2	28.7	5.5	916.7	31.5
7	1203.2	7.8	19.6	1234.7	22.8	22.8	1196.5	25.4	19.0	1013.2	28.6	0.7	1005.8	31.0
8	1228.8	8.2	36.6	1165.1	25.9	29.5	1107.5	25.7	23.1	977.5	29.2	8.6	899.8	31.6
9	1179.9	8.1	17.3	1198.2	24.1	19.1	1204.8	25.8	19.8	1059.9	29.4	5.4	1005.8	30.7
10	1344.0	8.4	33.7	1265.1	23.9	25.9	1243.8	25.8	23.8	1069.4	28.8	6.4	1004.9	31.1
Avg.	1251.0	8.2	30.2	1201.5	23.9	24.9	1160.5	25.5	20.6	995.7	29.0	3.5	962.3	31.1
80x80														
1	2591.9	13.8	40.0	2385.7	56.8	28.9	2411.6	58.4	30.3	1928.4	68.3	4.2	1850.8	73.1
2	2445.4	13.0	31.0	2351.4	56.7	26.0	2274.1	58.3	21.9	1875.2	72.0	0.5	1866.2	73.0
3	2547.6	12.6	33.9	2481.1	58.5	30.4	2376.6	59.3	24.9	1894.8	72.8	-0.4	1902.6	72.8
4	2723.9	13.0	33.7	2572.2	57.3	26.2	2373.0	58.1	16.5	2046.8	73.0	0.4	2037.7	73.2
5	2586.0	12.8	32.2	2409.3	56.9	23.2	2284.2	58.2	16.8	2020.1	73.2	3.3	1956.0	73.8
6	2617.7	13.0	40.9	2505.7	56.7	34.9	2404.0	58.2	29.4	2002.7	69.4	7.8	1857.3	73.2
7	2585.4	13.5	35.6	2346.1	57.5	23.1	2347.7	58.9	23.1	1914.3	73.7	0.4	1906.4	73.2
8	2521.5	13.5	30.2	2462.6	56.7	27.1	2326.2	58.3	20.1	1961.4	71.8	1.3	1937.0	73.9
9	2663.4	13.0	34.1	2458.1	56.9	23.7	2420.7	55.9	21.9	1998.1	73.8	0.6	1986.4	73.4
10	2517.1	13.8	23.9	2476.8	56.1	21.9	2503.7	56.8	23.2	2060.8	73.1	1.4	2032.3	73.0
Avg.	2580.0	13.3	33.6	2444.9	57.0	26.5	2372.2	58.0	22.8	1970.3	72.1	1.9	1933.3	73.3
160x160														
1	4977.9	17.8	32.2	4842.8	136.5	28.6	4838.6	142.3	28.5	3836.2	169.1	1.9	3764.7	173.7
2	5104.9	17.9	28.0	5128.7	139.5	28.6	4946.5	136.5	24.0	3993.1	174.2	0.1	3989.4	176.6
3	4892.5	17.5	28.5	4867.6	134.7	27.8	4818.6	134.7	26.5	3808.7	175.6	2.2	3726.1	172.5
4	5064.1	16.6	31.8	4959.0	138.5	29.0	4799.2	137.9	24.9	3857.4	177.2	0.4	3842.7	172.7
5	5160.7	17.9	37.9	4959.1	136.1	32.5	4863.4	141.6	29.9	3761.0	177.8	0.5	3743.6	181.6
6	4997.6	17.8	32.3	4981.5	134.2	31.8	4840.1	132.1	28.1	3790.1	168.8	0.3	3778.3	173.6
7	5200.2	17.8	36.1	4930.7	138.8	29.0	4857.5	139.4	27.1	3829.5	173.8	0.2	3822.2	182.4
8	5019.2	17.7	29.4	5013.0	135.7	29.3	4846.9	134.7	25.0	3878.1	174.6	-0.5	3857.3	181.5
9	5120.7	17.6	29.9	4956.7	135.1	25.7	4875.5	137.1	23.7	3941.0	173.3	0.0	3942.5	172.8
10	5122.4	17.9	30.4	4921.8	134.0	25.3	4921.4	139.7	25.3	3929.4	173.3	0.0	3927.8	178.3
Avg.	5066.0	17.7	31.6	4956.1	136.3	28.8	4860.8	137.6	26.3	3862.5	173.8	0.5	3839.5	176.6

Z: the objective function value; T: time (s); G: gap to the Hybrid3 (WOA+PSO)

Big Order Analysis shows that the three hybridizations have the most computational complexity, followed by PWOA, WOA, and PSO. For the GA, its computational complexity depends on the maximum generation (G), the number of populations (P), and the number of dimensions (N). The computational complexity is O($G \times P \times N$), similar to WOA and PSO. The experimental results are found not far from the above analysis. Hybrid3 (WOA+PSO) takes the most computation time due to having the highest computational complexity and using the adaptive movements for whales.

7) This research shows that the proposed framework is beneficial for developing simulation-based optimization approaches. The framework can integrate metaheuristic, simulation, and optimization seamlessly. Based on this framework, the developed approaches are proven able to solve the DCBSP for an S/R machine in a unit-load aisle-captive AS/RS.

8) The sensitivity analysis shows that under the same total populations of seabirds and/or whales, the three hybrid approaches remain better than the standard WOA and


Z values of differet approaches with an equal total population number of seabirds and/or whales (20x20)

Z values of different approaches with an equal total population number of seabirds and/or whales (40x40)

Z values of different approaches with an equal total population number of seabirds and/or whales (80x80)

Z values of different approaches with an equal population number of seabirds and/or whales (160x160)

(a). 20x20

(b). 40x40

(c). 80x80

(d). 160x160

FIGURE 7. The Z values of different approaches with an equal total population number of seabirds and/or whales under different problem sizes: (a) 20 × 20 (b) 40 × 40 (c) 80 × 80 (d) 160 × 160.

PSO. And, Hybrid3 (WOA+PSO) outperforms the other approaches.

9) This research only focuses on a static framework. Nevertheless, a dynamic framework can be achieved when the static framework is used iteratively to process incoming storage and retrieval requests. In practice, the number of storage and retrieval requests (N) in each block can be adjusted dynamically to adapt to environmental conditions. This helps form the dual commands.

10) The [6] reviewed 82 WOA-based articles (during 2016–2020). It is found that 61% of them were devoted to modification, 27 % to hybridization, and 12 % to multiple objective techniques. The hybridizations are not much. This research has an additional contribution.

VI. CONCLUSION

This research solves the DCBSP for an S/R machine in a unit-load aisle-captive AS/RS. First, a MILP model is formulated for the DCBSP. The objective is to minimize the total operational time. A framework was proposed to closely couple metaheuristic, simulation, and optimization. This framework helps develop simulation-based optimization approaches in which metaheuristics including WOA, PSO, Hybrid1 (WOA+PSO), Hybrid2 (WOA+PSO), and Hybrid3 (WOA+PSO) serve as alternative scheduling methods. Numerical experiments were conducted to compare their effectiveness. The Hybrid3 (WOA+PSO) was found to outperform the others. The contributions of this paper include: (1) the formulation of a MILP for DCBSP; (2) the proposal of a framework for seamless integrating metaheuristic, simulation, and optimization; (3) the development of Hybrid1 (WOA+PSO), Hybrid2 (WOA+PSO), and Hybrid3 (WOA+PSO); (4) the conduction of extensive experiments to verify the effectiveness of these approaches; (5) the successful application of these approaches to deal with the DCBSP. Some directions are available for future research. First, they improve the hybrid models continuously, such as enabling seabirds to have adaptive movement. Second, incorporating a storage position assignment strategy. Third, applying these hybrid models to solve problems in other areas. Forth, comparing these hybrid models to other metaheuristics. Sixth, extending single-block scheduling to multiple-block scheduling. Finally, In Hybrid3 (WOA+PSO), one additional feature is the use of adaptive movements for the whales in the WOA. This results in a little improvement. In future research, one application direction is applying adaptive improvements to

TABLE 13. The distance table of (*i,j*).

j *i*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	18.0	21.4	0.0	24.5	13.0	12.5	6.7	7.1	27.2	17.5	23.0	9.2	25.0	7.1	13.9	7.6	19.0	13.9	5.0	26.9
2	12.2	16.2	6.3	18.2	7.1	6.4	9.8	7.1	21.6	13.5	17.3	15.0	20.1	7.1	7.8	5.8	13.0	9.9	6.7	21.9
3	7.2	10.0	22.8	2.2	11.0	11.0	25.0	21.0	10.3	14.3	7.1	31.3	13.2	20.2	10.0	19.0	5.8	14.2	21.5	13.6
4	24.0	27.5	6.1	30.3	18.0	17.7	5.8	9.2	33.2	23.4	29.1	3.2	31.0	12.5	19.0	10.8	25.0	19.9	10.3	33.0
5	10.0	13.9	8.1	16.5	6.7	5.8	12.1	9.2	19.4	11.3	15.1	17.0	17.9	6.7	7.2	7.8	11.0	7.8	7.1	19.7
6	3.6	6.3	15.1	11.4	9.8	8.9	19.7	16.6	12.0	5.1	8.1	24.3	10.3	10.4	9.5	15.0	5.0	3.6	12.2	12.1
7	6.1	5.8	24.0	4.0	14.3	13.9	27.5	23.8	5.0	11.3	2.2	33.0	8.0	19.9	13.4	21.8	5.0	12.5	21.6	8.2
8	5.4	10.0	13.2	11.4	5.4	4.5	16.5	13.0	14.9	9.5	10.4	22.0	14.2	10.6	5.1	11.2	6.1	7.0	11.7	15.8
9	1.4	5.4	19.0	6.4	10.0	9.4	22.6	19.0	8.9	8.5	4.5	28.0	9.4	15.2	9.2	17.1	0.0	8.5	16.8	10.6
10	25.2	28.0	7.6	32.0	20.6	20.1	9.8	12.8	34.0	23.3	30.1	5.4	31.3	12.2	21.5	14.1	26.3	20.1	10.0	33.2
11	5.8	10.8	15.8	9.0	4.5	4.1	18.0	14.1	14.4	12.0	10.0	24.2	15.0	14.1	3.6	12.2	5.7	10.2	15.0	16.3
12	20.4	24.4	4.5	26.0	13.3	13.2	2.2	4.2	29.8	21.1	25.5	7.3	28.2	11.4	14.3	5.8	21.2	17.5	9.4	30.1
13	7.2	7.3	15.2	14.9	13.3	12.4	20.8	18.4	13.3	2.2	10.3	24.4	10.0	9.1	13.2	17.0	8.6	1.4	11.2	12.0
14	8.1	4.0	25.3	9.1	17.8	17.2	29.7	26.4	2.2	9.5	3.6	34.5	3.2	20.0	17.0	24.6	7.8	12.0	22.0	3.2
15	27.1	30.1	9.2	33.7	21.9	21.5	10.0	13.5	36.0	25.5	32.0	4.0	33.4	14.3	22.8	15.0	28.2	22.2	12.2	35.4
16	7.8	9.5	24.5	0.0	13.2	13.0	27.0	23.1	8.5	14.4	6.1	33.1	12.0	21.5	12.2	21.1	6.4	14.9	22.8	12.2
17	17.7	21.9	5.1	23.0	10.2	10.0	4.1	2.0	27.2	19.2	22.8	10.4	25.9	10.8	11.2	2.8	18.4	15.6	9.2	27.7
18	11.7	14.0	8.1	19.2	11.7	10.8	14.1	12.5	20.0	9.5	16.3	17.1	17.3	2.2	12.1	11.7	13.0	6.1	4.0	19.2
19	6.7	6.3	16.2	14.2	13.6	12.6	21.6	19.1	12.4	1.4	9.4	25.3	9.1	10.0	13.3	17.7	8.1	2.2	12.2	11.0
20	4.5	1.0	20.4	9.8	14.2	13.5	25.1	22.0	7.1	5.0	4.2	29.6	5.0	15.0	13.6	20.2	5.1	7.1	17.0	6.7

the seabirds in the PSO. This is expected to improve the Hybrid3 (WOA+PSO) further.

APPENDIX A

See Table 13.

APPENDIX B

See Algorithm 5.

Algorithm 5 . The Logic Flow of the PWOA

```
for i= to  population do
    population ← random(  )
end for
set i=1
while i < 1/2 max_iteration do
    Calculate gbest and pgest
    Update particle velocity
    Update particle position
    i=i+1
End while
Set leaderpos=gbest
while i< max_iteration do
    calculate leaderpos
    update A, C, l, r, $a$, $|\vec{A}|$, $\vec{C}$, $l$, and $p$
    update positions of whales
    i=i+1
End while
Output leaderpos
```

REFERENCES

[1] A. Eynan and M. J. Rosenblatt, "Establishing zones in single-command class-based rectangular AS/RS," *IIE Trans.*, vol. 26, no. 1, pp. 38–46, Jan. 1994.

[2] K. J. Roodbergen and I. F. A. Vis, "A survey of literature on automated storage and retrieval systems," *Eur. J. Oper. Res.*, vol. 194, no. 2, pp. 343–362, Apr. 2009.

[3] S. Hsieh and K.-C. Tsai, "A BOM oriented class-based storage assignment in an automated storage/retrieval system," *Int. J. Adv. Manuf. Technol.*, vol. 17, no. 9, pp. 683–691, May 2001.

[4] S. C. Graves, W. H. Hausman, and L. B. Schwarz, "Storage-retrieval interleaving in automatic warehousing systems," *Manag. Sci.*, vol. 23, no. 9, pp. 935–945, May 1977.

[5] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[6] N. Rana, M. S. A. Latiff, S. M. Abdulhamid, and H. Chiroma, "Whale optimization algorithm: A systematic review of contemporary applications, modifications and developments," *Neural Comput. Appl.*, vol. 32, no. 20, pp. 16245–16277, Oct. 2020.

[7] S. Bansal and H. A. Aggarwal, "Hybrid particle whale optimization algorithm for workflow scheduling in cloud-fog-mobile computing environment," *Res. Square*, pp. 1–23, 2022, doi: 10.21203/rs.3.rs-2135828/v1.

[8] Y. A. Bozer and J. A. White, "Travel-time models for automated storage/retrieval systems," *IIE Trans.*, vol. 16, no. 4, pp. 329–338, Dec. 1984.

[9] B. C. Park, "Optimal dwell point policies for automated storage/retrieval systems with dedicated storage," *IIE Trans.*, vol. 31, no. 10, pp. 1011–1113, Oct. 1999.

[10] S.-G. Koh, H.-M. Kwon, and Y.-J. Kim, "An analysis of the end-of-aisle order picking system: Multi-aisle served by a single order picker," *Int. J. Prod. Econ.*, vol. 98, no. 2, pp. 162–171, Nov. 2005.

[11] T. Lerher, M. Sraml, J. Kramberger, I. Potrc, M. Borovinsek, and B. Zmazek, "Analytical travel time models for multi aisle automated storage and retrieval systems," *Int. J. Adv. Manuf. Technol.*, vol. 30, nos. 3–4, pp. 340–356, Sep. 2006.

[12] R. Manzini, R. Accorsi, M. Gamberi, and S. Penazzi, "Modeling class-based storage assignment over life cycle picking patterns," *Int. J. Prod. Econ.*, vol. 170, pp. 790–800, Dec. 2015, doi: 10.1016/j.ijpe.2015.06.026.

[13] M. Foumani, A. Moeini, M. Haythorpe, and K. Smith-Miles, "A cross-entropy method for optimising robotic automated storage and retrieval systems," *Int. J. Prod. Res.*, vol. 56, no. 19, pp. 6450–6472, Oct. 2018, doi: 10.1080/00207543.2018.1456692.

[14] M. H. Han, L. F. McGinnis, J. S. Shieh, and J. A. White, "On sequencing in an automated storage/retrieval system," *IIE Trans.*, vol. 19, no. 1, pp. 56–66, 1987.

[15] H. Hwang and C. S. Ko, "A study on multi-aisle system served by a single storage/retrieval machine," *Int. J. Prod. Res.*, vol. 26, no. 11, pp. 1727–1737, Nov. 1988.

[16] B. R. Sarker, L. Mann Jr., and J. R. L. Dos Santos, "Evaluation of a class-based storage scheduling technique applied to dual-shuttle automated storage and retrieval systems," *Prod. Planning Control*, vol. 5, no. 5, pp. 442–449, Sep. 1994, doi: 10.1080/09537289408919516.

[17] T. N. Larson, H. March, and A. Kusiak, "A heuristic approach to warehouse layout with class-based storage," *IIE Trans.*, vol. 29, no. 4, pp. 337–348, 1997, doi: 10.1023/A:1018591700748.

[18] S. Mahajan, B. V. Rao, and B. A. Peters, "A retrieval sequencing heuristic for miniload end-of-aisle automated storage/retrieval systems," *Int. J. Prod. Res.*, vol. 36, no. 6, pp. 1715–1731, Jun. 1998, doi: 10.1080/002075498193246.

[19] P. Yang, L. Miao, Z. Xue, and B. Ye, "Variable neighborhood search heuristic for storage location assignment and storage/retrieval scheduling under shared storage in multi-shuttle automated storage/retrieval systems," *Transp. Res. E, Logistics Transp. Rev.*, vol. 79, pp. 164–177, Jul. 2015.

[20] T. Wauters, F. Villa, J. Christiaens, R. Alvarez-Valdes, and G. V. Berghe, "A decomposition approach to dual shuttle automated storage and retrieval systems," *Comput. Ind. Eng.*, vol. 101, pp. 325–337, Nov. 2016.

[21] L. Li and Z. Chen, "Hungarian-based heuristics for single-machine flow-rack AS/RS with determined storage and retrieval locations," in *Proc. 3rd Int. Conf. Comput. Sci. Appl. Eng.*, Oct. 2019, pp. 1–7, doi: 10.1145/3331453.3361669.

[22] P. Yang, L. X. Mio, and M. Qi, "Slotting optimization in a multi-shuttle automated storage and retrieval system," *Comput. Integr. Manuf. Syst.*, vol. 17, pp. 1050–1055, May 2011.

[23] K.-Y. Wu, S. S.-D. Xu, and T.-C. Wu, "Optimal scheduling for retrieval jobs in double-deep AS/RS by evolutionary algorithms," *Abstract Appl. Anal.*, vol. 2013, pp. 1–17, 2013, doi: 10.1155/2013/634812.

[24] D. Popovic, M. Vidovic, and N. Bjelic, "Application of genetic algorithms for sequencing of AS/RS with a triple-shuttle module in class-based storage," *Flexible Services Manuf. J.*, vol. 26, no. 3, pp. 432–453, Sep. 2014, doi: 10.1007/s10696-012-9139-2.

[25] S. Brezovnik, J. Gotlih, J. Balic, K. Gotlih, and M. Brezocnik, "Optimization of an automated storage and retrieval systems by swarm intelligence," *Proc. Eng.*, vol. 100, pp. 1309–1318, Jan. 2015.

[26] A. R. Nia, H. Haleh, and A. Saghaei, "Dual command cycle dynamic sequencing method to consider GHG efficiency in unit-load multiple-rack automated storage and retrieval systems," *Comput. Ind. Eng.*, vol. 111, pp. 89–108, Sep. 2017, doi: 10.1016/j.cie.2017.07.007.

[27] F. Chen, G. Xu, and Y. Wei, "An integrated metaheuristic routing method for multiple-block warehouses with ultranarrow aisles and access restriction," *Complexity*, vol. 2019, pp. 1–14, Jun. 2019, doi: 10.1155/2019/1280285.

[28] M. Cunkas and O. Ozer, "Optimization of location assignment for unit-load AS/RS with a dual-shuttle," *Int. J. Intell. Syst. Appl. Eng.*, vol. 7, no. 2, pp. 66–71, Jun. 2019.

[29] H. H. Tostani, H. Haleh, S. M. H. Molana, and F. M. Sobhani, "A bi-level bi-objective optimization model for the integrated storage classes and dual shuttle cranes scheduling in AS/RS with energy consumption, workload balance and time Windows," *J. Cleaner Prod.*, vol. 257, Jun. 2020, Art. no. 120409, doi: 10.1016/j.jclepro.2020.120409.

[30] L. Hojaghania, J. Nematian, A. A. Shojaiea, and M. Javadi, "Metaheuristics for a new MINLP model with reduced response time for on-line order batching," *Scientia Iranica*, vol. 28, no. 5, pp. 2789–2811, 2021, doi: 10.24200/sci.2019.51452.2185.

[31] J. Ashayeri, L. Gelders, and L. V. Wassenhove, "A microcomputer-based optimization model for the design of automated warehouses," *Int. J. Prod. Res.*, vol. 23, no. 4, pp. 825–839, Jan. 1985, doi: 10.1080/00207548508904750.

[32] A. Azzi, D. Battini, M. Faccio, A. Persona, and F. Sgarbossa, "Innovative travel time model for dual-shuttle automated storage/retrieval systems," *Comput. Ind. Eng.*, vol. 61, no. 3, pp. 600–607, Oct. 2011.

[33] S. U. Randhawa and R. Shroff, "Simulation-based design evaluation of unit load automated storage/retrieval systems," *Comput. Ind. Eng.*, vol. 28, no. 1, pp. 71–79, Jan. 1995.

[34] Y. H. Lee, M. H. Lee, and S. Hur, "Optimal design of rack structure with modular cell in AS/RS," *Int. J. Prod. Econ.*, vol. 98, no. 2, pp. 172–178, Nov. 2005.

[35] K. Hachemi, Z. Sari, and N. Ghouali, "A step-by-step dual cycle sequencing method for unit-load automated storage and retrieval systems," *Comput. Ind. Eng.*, vol. 63, no. 4, pp. 980–984, Dec. 2012.

[36] N. Boysen and K. Stephan, "A survey on single crane scheduling in automated storage/retrieval systems," *Eur. J. Oper. Res.*, vol. 254, no. 3, pp. 691–704, Nov. 2016, doi: 10.1016/j.ejor.2016.04.008.

[37] H.-P. Hsu, "A fuzzy knowledge-based disassembly process planning system based on fuzzy attributed and timed predicate/transition net," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 8, pp. 1800–1813, Aug. 2017.

[38] H.-P. Hsu, C.-N. Wang, C.-C. Chou, Y. Lee, and Y.-F. Wen, "Modeling and solving the three seaside operational problems using an object-oriented and timed predicate/transition net," *Appl. Sci.*, vol. 7, no. 3, p. 218, Feb. 2017.

HSIEN-PIN HSU received the Ph.D. degree from the Department of Industrial Engineering and Management, National Chiao-Tung University, Hsinchu, Taiwan, in 2004. He is currently a full-time Professor with the Department of Supply Chain Management, National Kaohsiung University of Science and Technology, Taiwan. He is also familiar with Manufacturing Execution Systems (MES), such as WORKSTREAM and PROMIS. He had ever worked as a MIS manager in some semiconductor assembly and testing firms in Taiwan. He has published many articles in international journals, including IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Systems, Man, and Cybernetics: Systems, IEEE Transactions on Semiconductor Manufacturing, IEEE Access, *IJPR*, *SAEC*, *ESWA*, and *IJAMT*. His research interests include MES, simulation, system modeling, petri net, predicate/transition net, soft computing, metaheuristics application, and operations research in the factory and container terminal areas.

CHIA-NAN WANG received the Ph.D. degree from the Department of Industrial Engineering and Management, National Chiao-Tung University, Taiwan, in 2004. He is currently a Professor with the Department of Industrial Engineering and Management, National Kaohsiung University of Science and Technology, Taiwan. He has published 80 journal articles and 40 patents. His research interests include manufacturing automation and industrial applications. He achieves around 25 projects of research and academy and has some successful cases of technology transfer. He also achieves awards from many international invention competitions (two gold medal awards, five silver medal awards, and several bronze medal awards). In addition, he has plenty of practical experience in many industries and provides consultant services for several stock companies in Taiwan.