

Article

Simulation-Based Optimization Approaches for Dealing with Dual-Command Crane Scheduling Problem in Unit-Load Double-Deep AS/RS Considering Energy Consumption

Hsien-Pin Hsu ^{1,*} , Chia-Nan Wang ^{2,*}  and Thanh-Tuan Dang ³ 

¹ Department of Supply Chain Management, National Kaohsiung University of Science and Technology, Kaohsiung 81157, Taiwan

² Department of Industrial Engineering and Management, National Kaohsiung University of Science and Technology, Kaohsiung 807618, Taiwan

³ Department of Logistics and Supply Chain Management, Hong Bang International University, Ho Chi Minh 72320, Vietnam

* Correspondence: hphsu@nkust.edu.tw (H.-P.H.); cn.wang@nkust.edu.tw (C.-N.W.)

Abstract: Automated storage/retrieval systems (AS/RS) have been increasingly used to support operations in manufacturing firms, warehouses, and distribution centers. Usually, AS/RSs are expensive. To achieve a good return on investment (ROI), an AS/RS must operate optimally. This research focuses on solving the crane scheduling problem, which has a great and immediate impact on the performance of an AS/RS. To optimize the design and operations of an AS/RS, many past studies have applied the simulation approach. However, the simulation and optimization have been often loosely coupled, resulting in a rigorous and labor-intensive optimization procedure. Using population- and evolution-based metaheuristics to deal with the crane scheduling problem of an AS/RS is one of the research trends. However, the whale optimization algorithm (WOA) and its variants have not been used for this purpose. To address the said gaps, this research first proposes a framework for coupling the simulation and optimization closely, in which various heuristics/metaheuristics, including first-come first-serve (FCFS), RANDOM, WOA, genetic algorithms (GAs), particle swarm optimization (PSO), and especially an improved WOA (IWOA), together with dynamic programming (DP), have been used as alternative sequencing methods. Based on this framework, different simulation-based optimization approaches have been developed for solving the dual-command crane scheduling problem in a unit-load double-deep AS/RS. The experimental results show that IWOA+DP outperforms the others in terms of energy consumption.

Keywords: automated storage and retrieval (AS/RS); metaheuristics; whale optimization algorithm (WOA); crane scheduling problem

MSC: 68U35; 46N20; 74P05; 78M50



Citation: Hsu, H.-P.; Wang, C.-N.; Dang, T.-T. Simulation-Based Optimization Approaches for Dealing with Dual-Command Crane Scheduling Problem in Unit-Load Double-Deep AS/RS Considering Energy Consumption. *Mathematics* **2022**, *10*, 4018. <https://doi.org/10.3390/math10214018>

Academic Editors: Xavier Blasco, Ana M. Madureira, Joao Ferreira and André Santos

Received: 19 September 2022

Accepted: 25 October 2022

Published: 29 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Since the 1950s, automated storage/retrieval systems (AS/RSs) have been increasingly used for material handling in manufacturing firms, warehouses, and distribution centers [1]. The popularity of AS/RSs comes from their advantages, such as high throughput, space-saving ability, high reliability, low inventory cost, and low labor cost [2,3]. An AS/RS is very useful. It can be used to move and accommodate different stock units (SKUs), such as raw materials, work-in-process goods, and finished goods. Usually, AS/RSs are expensive. To have a good return on investment (ROI), an AS/RS must operate optimally.

An AS/RS is composed of hardware and software components that determine the overall performance. The common hardware components of an AS/RS include racks, storage bins, cranes, input/output stations (I/O stations), and a computer. However,

software components such as storage and retrieval rules (S/R rules) and storage allocation policies are essential for controlling and using the hardware components of an AS/RS.

For an AS/RS to operate optimally, it needs to deal with relevant problems, including layout, dwell-point location, storage location allocation, batching, and crane scheduling [4]. The first three are long-term problems while the last two are short-term problems. The layout of an AS/RS has a fundamental impact on the AS/RS, which is difficult to change. The dwell-point problem, which involves arranging a place for a crane to stay and wait for the next task, can be considered a part of the layout. Given the layout of an AS/RS, the allocation of storage positions for stocks is another essential long-term problem. Each kind of SKU in an AS/RS can have one/multiple dedicated/open storage position(s). In terms of the short-term problem, the batching problem considers how multiple orders (requests) be combined into one single tour for a human picker or one command cycle for an automated picking device (such as the crane). However, a crane usually has few/no batching options due to limited capacity (one or two shuttles). Another short-term problem, the crane scheduling problem, considers how the storage and retrieval requests can be arranged for a crane. It has an immediate impact on an AS/RS. The performance of a crane is also affected by the command mode used. Single-command and dual-command modes are two common command modes used by a crane. The single-command mode accomplishes one storage or retrieval request within one command cycle while the dual-command mode can complete one storage request and one retrieval request within one cycle. The dual-command mode is advantageous in terms of travel times [5]. A reduction of about 50% of the travel time or an increment of 10-15% of the throughput is possible [6]. Thus, the dual-command mode is focused on in this research. The crane scheduling problem of an AS/RS can be considered a Chebyshev traveling salesman problem, which is known to be NP-complete [7].

Our literature review shows that analytical models, heuristics, metaheuristics, and simulation have been commonly used to deal with crane scheduling problems in an AS/RS. The analytical models have been applied for different kinds of AS/RSs, including unit-load AS/RSs [8–13], end-of-aisle mini-load AS/RSs [7,14–16], and man-on-board AS/RSs [17]. Heuristics are another commonly used method. Sarker et al. [18] used a simple nearest-neighbor heuristic and a perimeter heuristic to sequence storage and retrieval requests in a dual-shuttle AS/RS. Mahajan et al. [19] proposed a nearest-neighbor heuristic to form dual commands. The heuristic methods were used to solve the crane scheduling problem for man-on-board AS/RSs [20–22].

Besides the analytical models and heuristics, population- and evolution-based metaheuristics have been increasingly used to deal with AS/RS problems. Such metaheuristics can improve solution quality through the use of a population of agents and an iterative mechanism. To deal with the crane scheduling problem, Yang et al. [23] used a hybrid genetic algorithm (HGA) for a multi-shuttle AS/RS. The objective was to minimize the travel time needed to complete all storage and retrieval operations. Wu et al. [24] scheduled retrieval requests for a double-deep AS/RS in a flexible manufacturing system. A genetic algorithm (GA), immune GA (IGA), and particle swarm optimization (PSO) were investigated. The objective was to minimize the retrieval time. The IGA was found to be the best in terms of travel distance while the PSO was the best in terms of computational cost. Popović et al. [25] used a GA to sequence requests in a triple-shuttle AS/RS, which used a class-based storage system under a modified sextuple command cycle policy with a planning horizon that comprises the realization of several successive cycles of the S/R machine. Brezovnik et al. [26] optimized AS/RS operations by using an ant colony optimization (ACO) metaheuristic, taking the factor of inquiry (FOI), product height (PH), storage space usage (SSU), and path to dispatch (PD) into consideration. Nia et al. [27] also employed ACO to find a near-optimal solution. The objective was to minimize the total cost of GHG efficiency for the dual-command crane scheduling problem in a unit-load multiple-rack AS/RS. ACO was found to be better than the GA. Chen et al. [28] hybridized ACO with a GA to deal with the routing problem of the crane in a multiple-block warehouse with ultranarrow aisles and access restrictions. Simulations have been used to evaluate

12 warehouse layouts. The experimental results show the superiority of the proposed approach over the dedicated heuristics in terms of solution quality. Cunkas and Ozer [29] used PSO to optimize location assignments for SKUs in a unit-load dual-shuttle AS/RS. The quadruple-command cycle was used. PSO was found to outperform the binary-coded GA (BGA) and real-coded GA (RGA). Tostani et al. [30] proposed a modified cooperative coevolutionary algorithm (MCoBRA) to deal with the dual-shuttle crane scheduling problem in an AS/RS. A bi-level optimization model considering total cost and energy consumption was proposed. The upper level determines the locations of SKUs based on a class-based storage policy. The lower level focuses on solving the crane scheduling problem. Hojaghani et al. [31] formulated a novel mixed-integer nonlinear program (MINLP) for online order batching. The aim was to improve the response and idle times. However, the MINLP was solved by two metaheuristics, ACO and artificial bee colony (ABC), and due to NP-hard, ACO was found better than the ABC. The above literature shows that using metaheuristics to solve the crane scheduling problem is a current trend. However, the whale optimization algorithm (WOA) has never been used for this purpose. Proposed by Mirjalili and Lewis [32], the WOA is a new swarm-intelligence (SI)-based metaheuristic that models the hunting behavior of humpback whales. This algorithm is found to have advantages such as a simple structure, fewer operators, a fast convergence speed, and balanced exploration and exploitation (Mirjalili and Lewis [32]). However, the standard WOA cannot move a whale adaptively, in addition to lacking the mutating capability to increase solution diversity. This kind of metaheuristics has never been used to deal with the crane scheduling problem in an AS/RS.

Simulation is useful to optimize the design and operations of an AS/RS. However, a thorough simulation study is usually necessary before achieving optimality. Ashayeri et al. [33] used simulation for the optimal design of a unit-load automated warehouse for a large oil company in Belgium. Han et al. [6] applied Monte Carlo simulation to improve the throughput of a unit-load AS/RS. FCFS is used as the sequencing rule. A nearest-neighbor sequencing rule was used as an alternative. Azzi et al. [34] also applied Monte Carlo simulation to estimate the travel time of a unit-load multiple-shuttle AS/RS. Randhawa and Shroff [35] used simulation to examine the effects of sequencing rules on six layout configurations of a unit-load AS/RS with different I/O stations, SKU distribution, rack distribution, etc. Lee [36] used the ARENA simulation software to optimize the number of rail-guided vehicles (RGVs) deployed. Hachemi et al. [37] used simulation to study the dual-command crane scheduling problem in a unit-load multiple-rack AS/RS. The objective was to minimize the travel time. FCFS is used for storage requests while the retrieval requests are gathered by block. Yang et al. [38] examined the joint optimization of the crane scheduling problem and storage location assignment problem in a unit-load multi-shuttle AS/RS. They proposed an effective variable neighborhood search (VNS) algorithm. However, in past studies, the simulation and optimization have been often loosely coupled, resulting in a rigorous and labor-intensive solution procedure. A framework for coupling simulation and optimization closely is thus necessary.

Environmental issues are recent global concerns. Boysen and Stephan [4] highlighted the need for energy-efficient crane scheduling for an AS/RS. Nia et al. [27] conducted a unique study that considered the GHG efficiency when dealing with the dual-command crane scheduling problem. However, there are few such studies, necessitating more research incorporating the energy consumption issue.

This research deals with the dual-command crane scheduling problem in a unit-load double-deep AS/RS. A MILP model is firstly formulated after defining this problem. The objective was to minimize energy consumption. Then, a framework is proposed to couple the simulation and optimization closely. Based on this, various simulation-based optimization approaches have been developed, in which various heuristics/metaheuristics, including FCFS, RANDOM, WOA, GA, PSO, and an improved WOA (IWOA), together with DP, have been used as sequencing methods. The experimental results show that the IWOA outperforms the others.

The contributions of this research are highlighted as follows:

- (1) A MILP is formulated for this problem;
- (2) A framework is proposed to couple the simulation and optimization closely;
- (3) An improved WOA (IWOA) with features of adaptive movement and mutation is developed for solving discrete problems using discrete operators;
- (4) Experiments are conducted to investigate the effectiveness of different simulation-based optimization approaches;
- (5) The successful application of the proposed approaches to solve the dual-command crane scheduling problem in the unit-load double-deep AS/RS is carried out.

The rest of this paper is organized as follows: Section 2 reviews background information on an AS/RS; Section 3 first defines the problem and then formulates a MILP model; Section 4 proposes a simulation-based optimization framework and develops an improved WOA (IWOA) as a sequencing method; Section 5 conducts experiments and analyzes the results; Section 6 presents conclusions and suggestions for the future.

2. Background

2.1. The Common Components of an AS/RS

As shown in Figure 1, an AS/RS includes hardware and software components, which are introduced as follows.



Figure 1. A single-aisle/double-rack AS/RS.

(1) Hardware components:

- Rack: a mainframe structure used to create a three-dimension space for the accommodation of SKUs. It is the main hardware component of an AS/RS. Each rack is assigned a rack number for identification. A rack can be designed with single, dual, or multiple depths.
- Bin: a cell or basic storage unit partitioned from a rack for storing stock items. Each bin is assigned a bin number for identification or can be identified by a coordinate denoted as (x, y, z) , where the x indicates a column number, y indicates a tier number, and z indicates a depth number.
- Aisle: a lane used to separate racks and provide moving space for a crane to reach a target bin position in a rack.
- Crane: also known as storage/retrieval (S/R) machine which is the main device to transport and store/retrieve SKUs into/from the rack of an AS/RS.
- Shuttles: a mechanical device on a crane for transferring SKUs between the crane and rack. A crane can have one, dual, or multiple shuttles, which affects the batching capability of a crane.
- I/O station: a loading/unloading station used to store/retrieve SKUs into/from the AS/RS. In this research, the I/O station defaults as the dwelling point of the crane.

- Computer: a device receiving service requests and controlling the operations of the cranes in the AS/RS.

(2) Software components

Besides the hardware components, the software components are essential for managing and controlling the hardware of an AS/RS. They include:

- Storage rule: a simple rule which decides the storage position or sequence of an SKU. For example, FCFS decides that the earliest arriving SKU should be stored first.
- Retrieval rule: a simple rule which decides the retrieval position or sequence of an SKU. For example, first-in first-out (FIFO) determines that the SKU with the earliest arrival should be retrieved first.
- Scheduling method: a method used to determine the sequences of requested services for a crane. A simple storage rule such as FCFS or Random can be used to schedule the storage requests. However, more comprehensive approaches, such as metaheuristics and simulation, can be used to achieve a more efficient schedule for a crane.
- Zoning policy: a policy arranging available storage space for various kinds of SKUs in an AS/RS. The overall storage space of an AS/RS can be classified into different zones for storing different kinds of SKUs based on their characteristics, such as value or frequency of usage of the SKUs, etc.
- Command mode: a command mode can affect the behavior of a crane. Single-command, dual-command, and multiple-command modes are different command modes of a crane.
- Batching: a method for combining multiple service requests into one tour of service by a picker or a crane.

2.2. The Classification of AS/RS

To solve an AS/RS problem, one first needs to understand the essence of the AS/RS. Referring to Roodbergen and Vis [39], Figure 2 shows a classification scheme based on the attributes of the crane, handling, and rack of AS/RSs. Considering the “shuttle” attribute, a crane can have one single, dual, or multiple shuttles. Considering the “movement” attribute, a crane can be “aisle dedicated” or “aisle shareable” between aisles. Considering the “handling” attribute, an AS/RS can have three types: man-on-board (MOB), end-of-aisle (EOA), and unit-load (UL). The MOB AS/RS has a man on board to pick up SKUs; the EOA AS/RS allows SKUs to be picked up at the end of an aisle; the UL AS/RS accesses the SKUs in pallet or bin units. Considering the “mobility” attribute, a rack can be a “stationary” or “moveable” one. The stationary rack can be “single-deep” or “double-deep.” A movable rack can be the “rotate” or “mobile” kind. The former kind allows rotation in the “horizontal” or “vertical” direction while the latter kind can move in various ways. For example, Kiva robots are used in the Amazon warehouse to move racks.

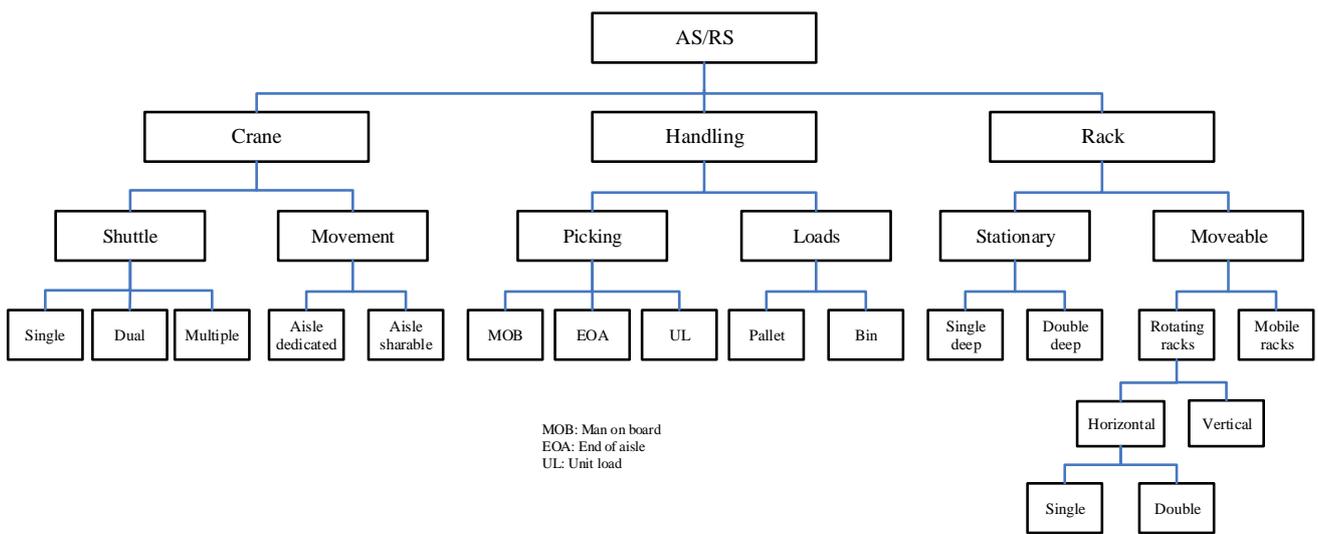


Figure 2. A classification scheme of AS/RSs.

2.3. Operational Time Analysis for a Dual-Command Crane in a Unit-Load Double-Deep AS/RS

Figure 3 shows the moving paths a crane uses to complete a dual command. First, the crane performs the storage request i by moving from the I/O station to the storage position S1, (x_i, y_i, z_i) . This travel time is denoted as T1. Then, the crane performs the retrieval request i' by moving from position S1 to position R1, $(x_{i'}, y_{i'}, z_{i'})$. This travel time is denoted as T2. Finally, the crane moves the retrieved SKU back to the I/O station. The travel time of the last movement is denoted as T3. The total time to complete this dual command is formulated as Equation (1).

$$T(DC)_{ii'} = T1 + ts + T2 + tr + T3$$

$$= \text{Max} \left\{ \frac{W \cdot x_i}{v_x}, \frac{H \cdot y_i}{v_y} \right\} + ts + \text{Max} \left\{ \frac{W \cdot |x_i - x_{i'}|}{v_x}, \frac{H \cdot |y_i - y_{i'}|}{v_y} \right\} + tr + \text{Max} \left\{ \frac{W \cdot x_{i'}}{v_x}, \frac{H \cdot y_{i'}}{v_y} \right\} \quad (1)$$

v_x (meter/second) is the moving speed of the crane along the horizontal direction while v_y (meter/second) is the moving speed of the crane along the vertical direction. W and H indicate the width and height (in the unit of meters) of each storage bin, respectively.

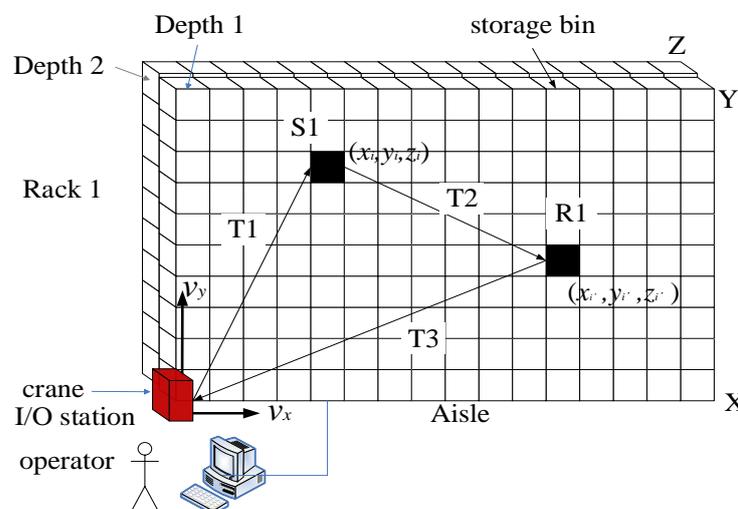


Figure 3. A unit-load and double-deep AS/RS.

ts is the time needed for the shuttle to store the SKU on a rack. This time is affected by the depth of the storage position of the SKU to be stored. Thus, ts is estimated by Equation (2).

$$ts = \begin{cases} 2D/v_z, & \text{if } z_i = 1 \text{ (first depth)} \\ (2D/v_z) * Cs, & \text{if } z_i = 2 \text{ (second depth)} \end{cases} \quad (2)$$

D indicates the depth of a storage position in a rack. v_z is the moving speed of the shutter along the Z direction. Cs is a coefficient for extending the storage time required for the SKU assigned at the second depth (i.e., $z_i = 2$) in a rack.

tr is the time needed for the shuttle to retrieve an SKU from a rack. Additionally, this time can be affected by the storage depth of the SKU in the rack. It is estimated by Equation (3).

$$tr = \begin{cases} 2D/v_z, & \text{if } z_{i'} = 1 \text{ (first depth)} \\ (2D/v_z) * Cr, & \text{if } z_{i'} = 2 \text{ (second depth)} \end{cases} \quad (3)$$

The Cr is a coefficient for extending the retrieval time required for the SKU stored at the second depth (i.e., $z_{i'} = 2$) in a rack.

2.4. Energy Consumption Calculation

Based on Nia et al. [27], the tax cost of energy consumption for the crane is calculated by Equation (4).

$$T(E)_{ii'} = P_c \times T(\text{DC})_{ii'} \times f_e \times C_e \quad (4)$$

where:

P_c : power consumption of crane (k watts/second);

f_e : GHG conversion factor;

C_e : GHG emission cost.

The penalty cost of exceeding GHG emissions and the discount for producing less GHG emissions than expected are not taken into account in this research.

3. Problem Definition and Mathematical Model Formulation

3.1. Definition of the Dual-Command Crane Scheduling Problem

In this research, dual-command crane scheduling is defined as a problem of forming the storage and retrieval service requests into dual-commands and sequencing their execution orders for a crane in a unit-load and double-deep AS/RS. Each crane is dedicated to serving two racks on both sides of a specific aisle. The racks have double the depth, i.e., double-deep racks (as shown in Figure 3). The crane moves in the horizontal and vertical directions simultaneously, with the moving distance measured by the Chebychev metric. Based on the classification notation proposed by Boysen and Stephan [4], this problem is denoted as

$$[F] \text{ O}^2 \mid \text{Open} \mid \sum T(E)_{ii'},$$

where:

F: I/O station at the front end;

Open: any open shelf is a potential location for storage requests;

O²: double commands;

$\sum T(E)_{ii'}$: the total cost of energy consumption for the crane using a dual command (storage request i and retrieval request i'). This is a new objective function.

The objective $\sum T(E)_{ii'}$ considers the total energy consumption cost of the crane.

Figure 4 represents a solution to this problem. The SR and RR are the sequences of storage and retrieval service requests, respectively. The parameter $N = 10$ indicates the total number of dual commands of storage and retrieval service requests. In this solution, the dual command of storage and retrieval requests is executed in the following sequence: (5,2),(3,6),(4,1),(1,10),(7,7),(2,4),(9,8),(6,5),(8,3),(10,9). If the storage and retrieval requests are sequenced according to their arrival times, then this is a solution obtained from the FCFS

rule. Each dual command is executed by the crane with the schedule $[S_i, E_{i'}]$, where S_i and $E_{i'}$ are the start and end times, respectively.

N	1	2	3	4	5	6	7	8	9	10
SR	5	3	4	1	7	2	9	6	8	10
RR	2	6	1	10	7	4	8	5	3	9

Figure 4. The representation of sequences of storage requests (SRs) and retrieval requests (RRs).

3.2. Mathematical Model Formulation

This section focuses on formulating a mathematical model for the dual-command crane scheduling problem. Before doing this, the required assumptions, indices, sets, parameters, and decision variables are first defined as follows.

Assumptions:

- Each storage/retrieval request is associated with one SKU.
- The position (x,y,z) of each SKU to be stored/retrieved is specified in each storage/retrieval request.
- The crane moves in the X (horizontal) and Y (vertical) directions simultaneously and the distance is measured by the Chebychev metric.
- No interruption occurs during the transport, storage, and retrieval operations.
- There are N storage and N retrieval requests to be handled by a specific crane.

The indices, sets, parameters, and decision variables required for the MILP are defined as follows.

Indices:

i, j	a storage request ID; $i, j \in \text{SR}$;
i', j'	a retrieval request ID; $i', j' \in \text{RR}$;
r	a rack number; $r \in \text{R}$;
x	a column number on a rack; $x \in \text{X}$;
y	a tier number on a rack; $y \in \text{Y}$;
z	a deep number on a rack; $z \in \text{Z}$.

Sets:

SR	a set of storage requests; $\text{SR} = \{1, \dots, N\}$;
RR	a set of retrieval requests; $\text{RR} = \{1, \dots, N\}$;
R	a set of rack; $\text{R} = \{1, \dots, \ \text{R}\ \}$;
X	a set of columns; $\text{X} = \{1, \dots, \ \text{X}\ \}$;
Y	a set of tiers on a rack; $\text{Y} = \{1, \dots, \ \text{Y}\ \}$;
Z	a set of depth numbers; $\text{Z} = \{1, \dots, \ \text{Z}\ \}$.

Parameters (Input data):

$\ \text{R}\ $	the number of racks served by a crane;
$\ \text{X}\ $	the total number of columns of a rack;
$\ \text{Y}\ $	the total number of tiers of a rack;
$\ \text{Z}\ $	the total number of depths in a rack; $\ \text{Z}\ = 1$ (1st depth); $\ \text{Z}\ = 2$ (2nd depth)
N	the total number of storage/retrieval service requests
(x_i, y_i, z_i)	the storage position of an SKU i in a rack;
$(x_{i'}, y_{i'}, z_{i'})$	the storage position of an SKU i' in a rack;
v_x	the moving speed of a crane along the X horizontal (column) direction;
v_y	the moving speed of a crane along the Y vertical (tier) direction;

v_z	the moving speed of a shuttle along the Z (depth) direction;
t_s	the shuttle time to store an SKU into the AS/RS;
t_r	the shuttle time to retrieve an SKU from the AS/RS;
f_e	GHG conversion rate;
C_e	penalty cost of GHG emissions;
C_s	a coefficient extending the storage time required for the SKU which is to be stored in the second depth in the rack;
C_r	a coefficient extending the retrieval time required for an SKU stored in the second depth in a rack.

Decision variables:

X_{ij}	= 1, if the storage request i is performed before the storage request j ; = 0, otherwise;
$Y_{i'j'}$	= 1, if the retrieval request i' is performed before the retrieval request j' ; = 0, otherwise;
$Z_{ii'}$	= 1, if the storage request i and the retrieval request i' is a pair of dual-command; = 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:

$$\text{Min } Z = \sum_{i=1}^N \sum_{i'=1}^N T(E)_{ii'} \cdot Z_{ii'} \tag{5}$$

s.t.

$$T(E)_{ii'} \geq 0 \quad \forall i, i' \in \text{SR} \tag{6}$$

$$\sum_{i=1}^N X_{ij} = 1 \quad \forall j \in \text{SR} \tag{7}$$

$$\sum_{j=1}^N X_{ij} = 1 \quad \forall i \in \text{SR} \tag{8}$$

$$X_{ij} \leq (N - 1)X_{ji} \quad \forall i, j \in \text{SR} \tag{9}$$

$$\sum_{j=1}^N X_{ji} = \sum_{j=1}^N X_{ji} + 1 \quad \forall i \in \text{SR} \tag{10}$$

$$\sum_{i'=1}^N Y_{i'j'} = 1 \quad \forall j' \in \text{RR} \tag{11}$$

$$\sum_{j'=1}^N Y_{ij'} = 1 \quad \forall i' \in \text{RR} \tag{12}$$

$$Y_{i'j'} \leq (N - 1)Y_{i'j'} \quad \forall i', j' \in \text{RR} \tag{13}$$

$$\sum_{j'=1}^N Y_{i'j'} = \sum_{j'=1}^N Y_{i'j'} + 1 \quad \forall i', j' \in \text{RR} \tag{14}$$

$$\sum_{i=1}^N \sum_{i'=1}^N Z_{ii'} = N \tag{15}$$

$$\sum_{i=1}^N Z_{ii'} = 1 \quad \forall i' \in \text{RR} \quad (16)$$

$$\sum_{i'=1}^N Z_{ii'} = 1 \quad \forall i \in \text{SR} \quad (17)$$

$$X_{ij}, Y_{i'j'}, Z_{ii'} \in \{0, 1\} \quad \forall i, j \in \text{SR} \quad \forall i', j' \in \text{RR} \quad (18)$$

Equation (5) is the objective function of minimizing the total cost of energy consumption of the crane. Equations (1)–(4) are also included. Constraint (6) defines the condition for the variable $T(E)_{ii'}$. Constraints (7)–(10) relate to the variables X_{ij} . Constraint (7) ensures that each storage request has only one predecessor. Constraint (8) ensures each storage request has only one successor. Constraints (9)–(10) relate to the variable X_{ij} which assigns a sequence of storage requests from a network view. Constraint (9) states that on leaving node i , there are no more than $N - 1$ remaining storage requests to be assigned. Equation (10) is a conservation constraint for node i . Similarly, constraints (11) and (14) relate to the variable $Y_{i'j'}$. Constraint (11) ensures that each retrieval request has only one predecessor. Constraint (12) ensures each retrieval request has only one successor. Constraints (13)–(14) are associated with the variable $Y_{i'j'}$, which assigns retrieval requests from a network view. Constraint (13) states that on leaving node i' , there are no more than $N - 1$ remaining retrieval requests to be assigned. Equation (14) is a conservation constraint for node i' . Constraints (15)–(17) are associated with the variable $Z_{ii'}$. Constraint (15) ensures there are exactly N pairs of dual commands. Constraint (16) ensures that a retrieval request is assigned with one storage request only. Constraint (17) ensures that a storage request is assigned with one retrieval request only. Constraint (18) stipulates that X_{ij} , $Y_{i'j'}$, and $Z_{ii'}$ are binary variables.

This problem has a solution space of $N! \times N!$, which tends to become computationally intractable if the N is big enough. Thus, a simulation-based optimization approach is proposed as an alternative approach.

4. Simulation-Based Optimization Approaches

4.1. A Simulation-Based Optimization Framework

Due to NP-complete, this section proposes a framework (Figure 5) that couples the simulation and optimization closely for developing alternative simulation-based optimization approaches. The main layers/functions of this framework are introduced as follows:

- **Input:** this is an initial step to creating input data for experiments. Configuration data of the AS/RS, such as the number of racks, the dimension of a rack, the moving speed of the crane, and the parameter values of heuristics/metaheuristics, are set manually. On the other hand, the storage and retrieval requests of SKUs and their target positions (x, y, z) are generated by a computer automatically. This simulates the randomness of service requests in the real world.
- **Requests sorting layer:** this layer sorts out the storage and retrieval service requests specifically for each crane based on the SKU's positions. Since the SKUs to be stored/retrieved are distributed across racks in the AS/RS, it needs first to sort out the storage and retrieval requests for each crane. As said previously, each crane is dedicated to serving two specific racks on both sides of an aisle. The sorting is based on the rack positions of the target SKUs. The storage and retrieval requests specifically for each crane are inputted to the sequencing layer.
- **Sequencing layer:** this layer focuses on generating alternative sequences of dual commands (one storage and one retrieval request) for a crane. Two simple heuristics (Random and FCFS) and four metaheuristics (WOA, IWOA+DP, GA, and PSO) are used in this research to generate and sequence dual commands. Conversely, the IWOA+DP uses a two-stage procedure. First, the IWOA finds a sequence of storage requests, based on which the dynamic programming (DP) method (detailed in Section 4.4) is used in the second stage to assign each storage request a retrieval

request to become a dual command. The sequence of dual commands is inputted into the simulation layer. As each sequence of storage and retrieval requests is NP-complete, the use of an analytical model to solve this problem optimality tends to become computationally intractable.

- **Simulation layer:** with the sequence of dual commands as input, this layer simulates the crane operations and evaluates the results. The best solution can be identified in this layer.
- **Termination check:** a check function is used to check whether the stop condition is reached. A total number of iterations is used as a stop condition in this research.
- **Output:** this step outputs the best solution.

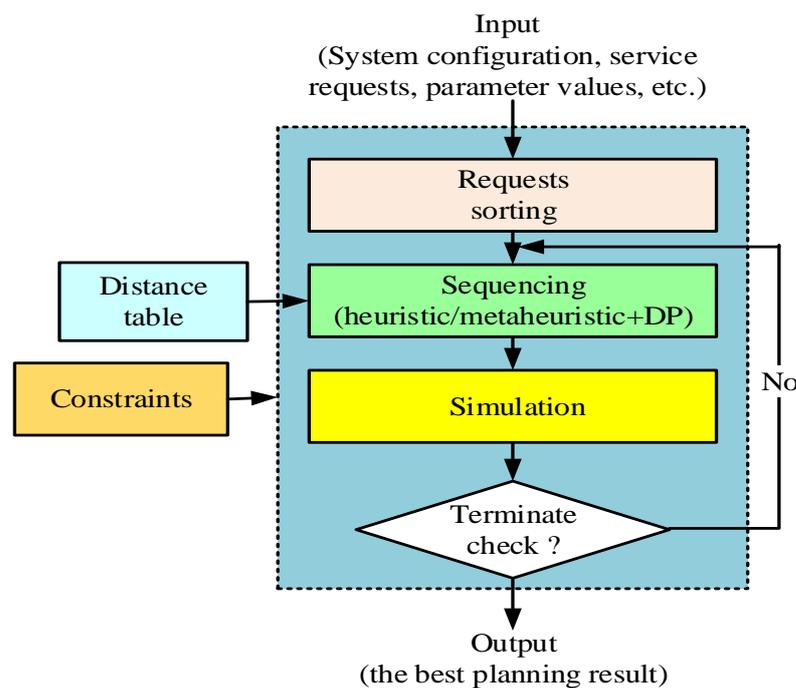


Figure 5. The simulation-based optimization framework.

The framework can be transformed into a workflow diagram with input and output data flows represented as dotted red arrow lines, as shown in Figure 6. For example, the Simulation module inputs distance data of each dual command into the Energy calculation module which calculates and returns the energy cost of dual commands.

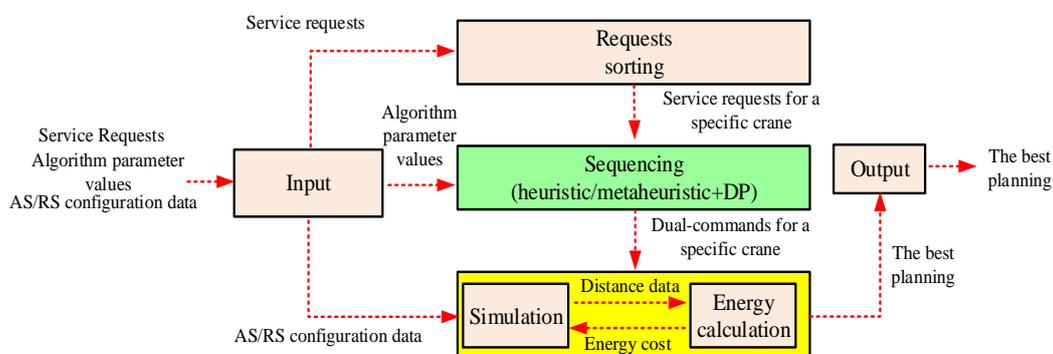


Figure 6. The workflow diagram of modules with input and output data flows.

4.2. Position Scheme

Figure 7 shows the position scheme of whales to search for a solution space. It includes two segments, each with the dimension N , where N is the total number of storage/retrieval requests.

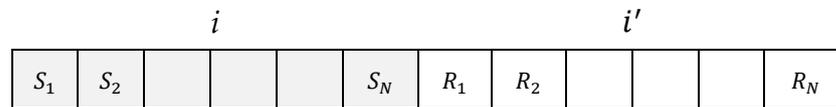


Figure 7. The position scheme used by whales.

The left segment represents the sequence of storage requests, in which S_i indicates the order of the storage request i ($i \in [1, N]$). The right segment indicates the sequence of retrieval requests, in which $R_{i'}$ indicates the order of the retrieval request i' ($i' \in [1, N]$). A series of dual commands are then formed and represented as $(S_i, R_{i'})$, where $i, i' \in [1, N]$.

4.3. WOA and Improved WOA (IWOA)

The WOA and IWOA are further detailed as follows.

4.3.1. WOA

Figure 8 shows the spiral-shaped movement of whales when they are circling and attacking their prey. Each whale’s position represents a solution in the solution space. Then, the distance between a whale and the prey (fish schools) represents an objective function value. The aim is to minimize the distance. The whales change their positions continuously during the hunting process, which contains three phases: encircling, prey, bubble-net attack (exploitation phase), and the search for prey (exploration phase) [32].

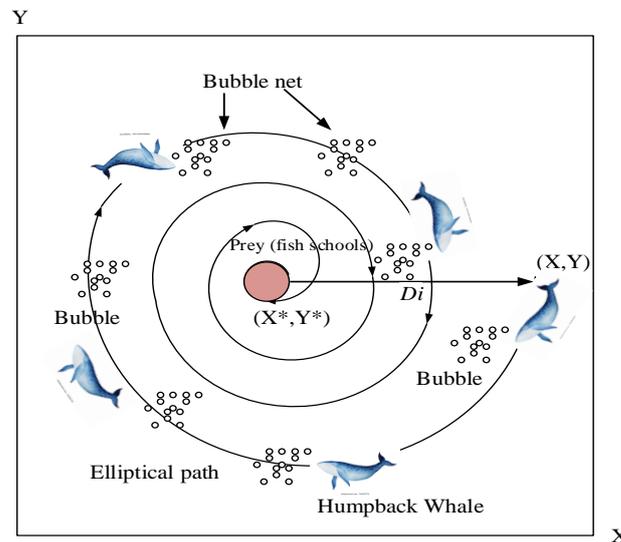


Figure 8. The spiral movements of humpback whales when hunting prey.

(1) Encircling prey

After noting the prey (school fish), the humpback whales set out to encircle and attack them. However, not knowing the prey’s exact position, it is assumed that the global best whale w^* is closest to the prey. Equation (19) shows the distance between the whale i and the target prey.

$$\vec{D}_i = \left| \vec{C} \cdot \vec{X}_{w^*}(t) - \vec{X}_i(t) \right| \tag{19}$$

$\vec{X}_i(t)$ and $\vec{X}_{w^*}(t)$ are position vectors of the whale i and the global best whale w^* , respectively. \vec{C} is a coefficient vector defined by Equation (20).

$$\vec{C} = 2 \cdot \vec{r} \tag{20}$$

$\vec{r} \in [0,1]$ is a vector of random numbers. Given the distance \vec{D}_i , the next position of the whale i is determined by Equation (21).

$$\vec{X}_i(t+1) = \vec{X}_i(t) - \vec{A} \cdot \vec{D}_i \tag{21}$$

\vec{A} is a coefficient vector defined by Equation (22).

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \tag{22}$$

where the \vec{a} is a linear value decreasing from 2 to 0 iteratively.

(2) Bubble-net attack (exploitation phase)

Equation (23) defines the use of shrinking circle- or spiral-shaped paths for whales during the exploitation phase.

$$\vec{X}_i(t+1) = \begin{cases} \vec{X}_{w^*}(t) - \vec{A} \cdot \vec{D}_i, & \text{if } p < 0.5 \\ \vec{D}_i \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}_{w^*}(t), & \text{if } p \geq 0.5 \end{cases} \tag{23}$$

The parameter random number $p \in [0,1]$ controls the selection of the shrinking circle- or spiral-shaped path. If $p < 0.5$, then the shrinking circle is used; otherwise, the spiral-shaped path is used. The parameter b (a constant value) defines the shape of the logarithmic spiral. The parameter $l \in [0,1]$ is a random number.

(3) Search for prey (exploration phase)

$\left| \vec{A} \right|$ controls the use of either an exploration or exploitation search for a whale. If $\left| \vec{A} \right| \geq 1$, Equations (24) and (25) are used for the exploration search.

$$\vec{D}_i = \left| \vec{C} \cdot \vec{X}_R(t) - \vec{X}_i(t) \right| \tag{24}$$

$$\vec{X}_i(t+1) = \vec{X}_R(t) - \left| \vec{A} \right| \cdot \vec{D}_i \tag{25}$$

$\vec{X}_R(t)$ is the position of a randomly selected whale. If $\left| \vec{A} \right| < 1$, then replace $\vec{X}_R(t)$ in Equations (24) and (25) with $\vec{X}_{w^*}(t)$ for the exploitation search.

Figure 9 shows the workflow of Algorithm A1 of the WOA. Appendix A shows the pseudo-code of the WOA.

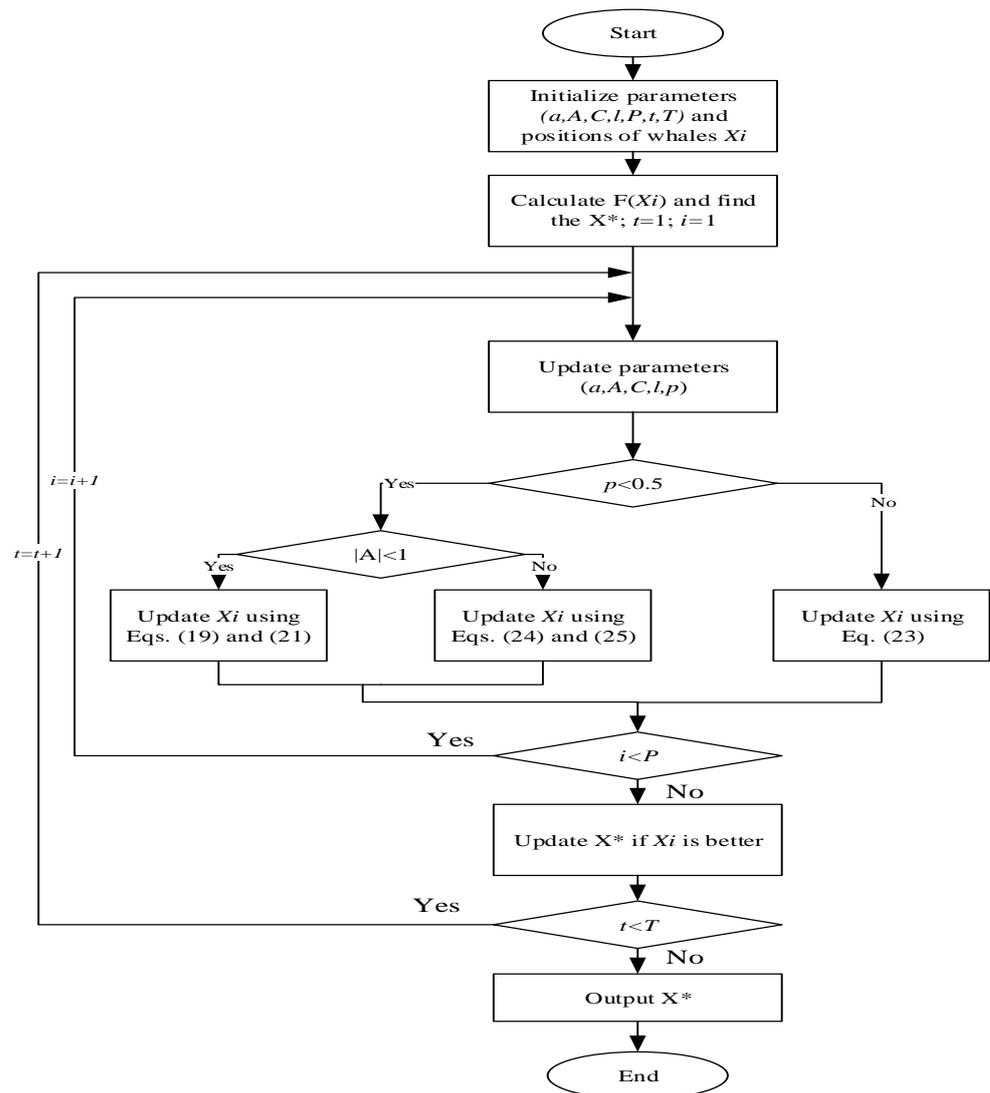


Figure 9. The workflow of Algorithm A1 of the WOA.

4.3.2. Improved WOA (IWOA)

The IWOA allows the humpback whales to be guided by a set of elite whales (denoted as ES), instead of only the global best whale w^* . Whale i can develop adaptive steps by referring to an elite whale $w' \in ES$.

- Adaptive movement of whales:

Referring to an elite whale $w' \in ES$, whale i develops an adaptive movement through the following three steps:

(1) Determine the Hamming distance: the first step is measuring the Hamming distance between the whale i and the elite whale w' , based on their current position vectors. The Hamming distance is defined in Equation (26).

$$HD(\vec{X}, \vec{Y}) = \sum_{i=1}^D \text{XOR}(\vec{X}_i, \vec{Y}_i) \tag{26}$$

It counts the total number of different elements between the two position vectors \vec{X} and \vec{Y} , where the D is a dimension of the vectors. \vec{X}_i and \vec{Y}_i are the i -th elements in \vec{X} and \vec{Y} , respectively. The XOR is an operator leading to either 1 ($\vec{X}_i \neq \vec{Y}_i$) or 0 (if $\vec{X}_i = \vec{Y}_i$).

(2) Determine a change rate (CR): the CR is a change rate controlling the replacement of the k -th element in $\vec{X}_i(t+1)$ with the k -th element in $\vec{X}_{w'}(t)$, simulating the movement toward the w' . A whale that is farther out requires a bigger CR, as defined in Equation (27).

$$CR = HD(i, w') / D \tag{27}$$

(3) Determine the next position of the whale i :

Based on the CR, this step determines the k -th element in the next position of whale i using Equation (28).

$$\vec{X}_i(t + 1) = \begin{cases} X_{w',k}(t), & \text{if } X_{i,k}(t) \neq X_{w',k}(t) \text{ and } R_k < CR \\ X_{i,k}(t), & \text{otherwise} \end{cases} \tag{28}$$

$R_k \in [0, 1]$ is a random number used for determining the k -th element in $\vec{X}_i(t + 1)$. Equation (28) provides a whale that is farther out with a higher opportunity to make a bigger step, while a closer whale is more likely to make a smaller step toward the w' . However, if $X_{w',k}(t)$ replaces $X_{i,k}(t)$ in $X_{i,k}(t + 1)$, $X_{i,k}(t)$ will take the place of the original $X_{w',k}(t)$ to ensure a feasible solution.

For example, given that $X_i(t) = [5,4,3,2,1]$ and $X_{w'}(t) = [3,4,2,5,1]$ are the current positions of the whale i and the elite whale w' , respectively, then we can derive $HD(i, w') = 2$, $CR = 3/5 = 0.6$. Additionally, the next position vector of the whale i will be determined by Equation (28). Given the random number $R_1 = 0.5$, then $X_{i,1}(t + 1)$ becomes 3, and element 5 in $X_i(t)$ will take the position of 3 in $X_i(t)$ and appear in $X_i(t + 1)$. Note that the derived solution remains feasible.

In addition, the IWOA includes the following features:

- The use of Sobol sequence:

This Sobol sequence enables the IWOA to initialize more uniform-distributed positions of whales in the solution space through the use of Sobol sequence numbers [40]. One weakness found in many past studies is that the random numbers generated by pseudorandom number generators embedded in the standard libraries of programming languages are not uniformly distributed [41].

- Mutation:

Mutation can diversify solutions. It avoids solutions sticking to local optima. Thus, the IWOA introduces a *mutation()* procedure that includes two kinds of mutations, swap mutation (SM) and inverse mutation (IM).

(1) **Swap mutation (SM):** this is a mutation that switches elements at two randomly selected points (such as $p1$ and $p2$) in a vector. For example, given $p1 = 2$ and $p2 = 6$, the position vector $[71324586]$ is mutated to $[75324186]$.

(2) **Inverse mutation (IM):** this mutation mutates elements in reverse order between two randomly selected points ($p1$ and $p2$) in a vector. The IM can have a higher degree of mutation. For example, given $p1 = 2$ and $p2 = 6$, the position vector $[71324586]$ is mutated to $[75423186]$. The IM has a bigger range of mutations.

Based on Hamming distance, a *similarity degree* (SD), as defined in Equation (29), is used to determine the use of SM or IM.

$$SD = HD(\vec{X}, \vec{Y}) / D \tag{29}$$

$HD(\vec{X}, \vec{Y})$ is defined in Equation (26).

Figure 10 shows the workflow of Algorithm A2 of the procedure *mutation()*. Appendix B shows the pseudo-code. The mutation happens when the condition, $R1 < \omega$, is true. $R1$ and $\omega \in [0,1]$, represent a random number and a mutation rate, respectively. The values of SD and θ control the use of either IM or SM. If $SD < \theta$ then the IM is used;

otherwise, the SM is used. The smaller the θ , the lower the probability of using the IM. A solution remains feasible after mutation due to a combination of sequence numbers from 1 to N .

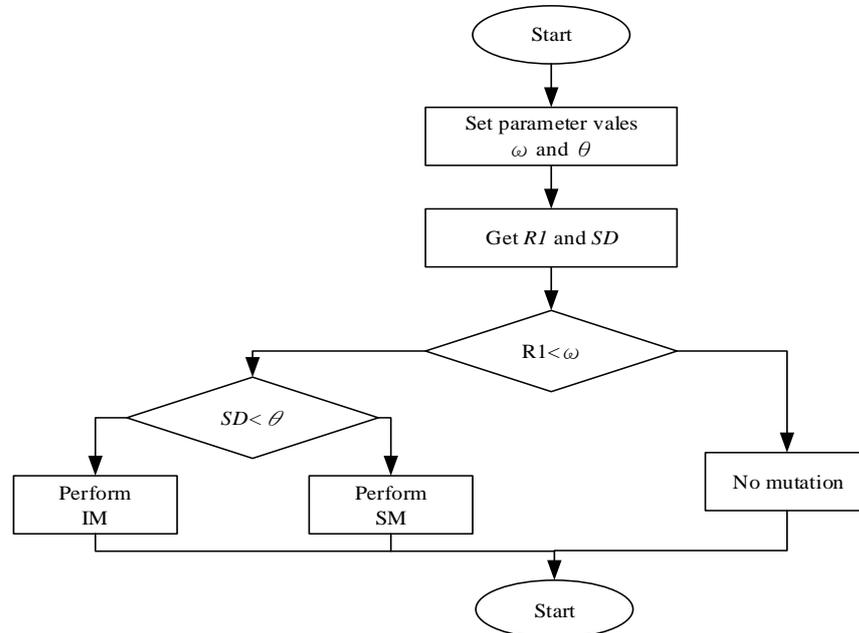


Figure 10. The workflow of Algorithm A2 of the procedure *mutation()*.

4.4. The Dynamic Programming

The DP works together with the IWOA to form a sequence of dual commands. With the output obtained from the IWOA in the first stage (i.e., the sequence of storage requests), the forward DP is used to form a sequence of dual commands by assigning each storage request with a retrieval service request through the N stages. The DP model contains Equations (30)–(32).

$$D_1(j) = d_{s_j}^1, j \in S_1 \tag{30}$$

$$D_k(j) = \min_{i \in S_{k-1}} \{d_{ij}^k + D_{k-1}(i)\}, j \in S_k, \forall k = 2, \dots, N - 1 \tag{31}$$

$$D_N = \min_{i \in S_{N-1}} \{c_{ij}^N + D_{N-1}(i)\}, j \in S_N \tag{32}$$

At each stage k ($k = 1, \dots, N$), the DP model assigns each storage request with a retrieval request at the k order, based on the distance table. In Equation (30), $D_1(j)$ indicates the total retrieval distance at stage 1; $d_{s_j}^1$ is the minimum distance at the state j of Stage 1, transitioned from the state s of the dummy stage 0; S_1 is a set of available states at Stage 1. In Equation (31), the $D_k(j)$ indicates the total retrieval distance at Stage k . d_{ij}^k is the minimum distance at state j of Stage k , transitioned from the state i of Stage $k-1$, where $2 \leq k \leq N - 1$. S_k indicates the set of available states at Stage k . In Equation (32), C_N indicates the total retrieval distance at the final Stage N . c_{ij}^N indicates the total retrieval distance at state j of the final Stage N transitioned from the state i at the previous Stage $N - 1$. The S_N indicates the set of available states at Stage N . At each stage k ($1 \leq k \leq N$), the DP forms a dual command consisting of one storage and one retrieval request.

4.5. The Main Logic Flow of the IWOA

Figure 11 shows the workflow of Algorithm A3 of the IWOA. Appendix C shows the pseudo-code. Compared with Algorithm A1, line 3 changes to using a Sobol sequence generator to initialize positions for whales; lines 8–19 include an additional loop, where

NE is the number of elites in the elite set ES; line 11, 14, and 17 change to use adaptive movements for whales; and line 24 adds a mutation procedure, *mutation()*. These are new features in the IWOA.

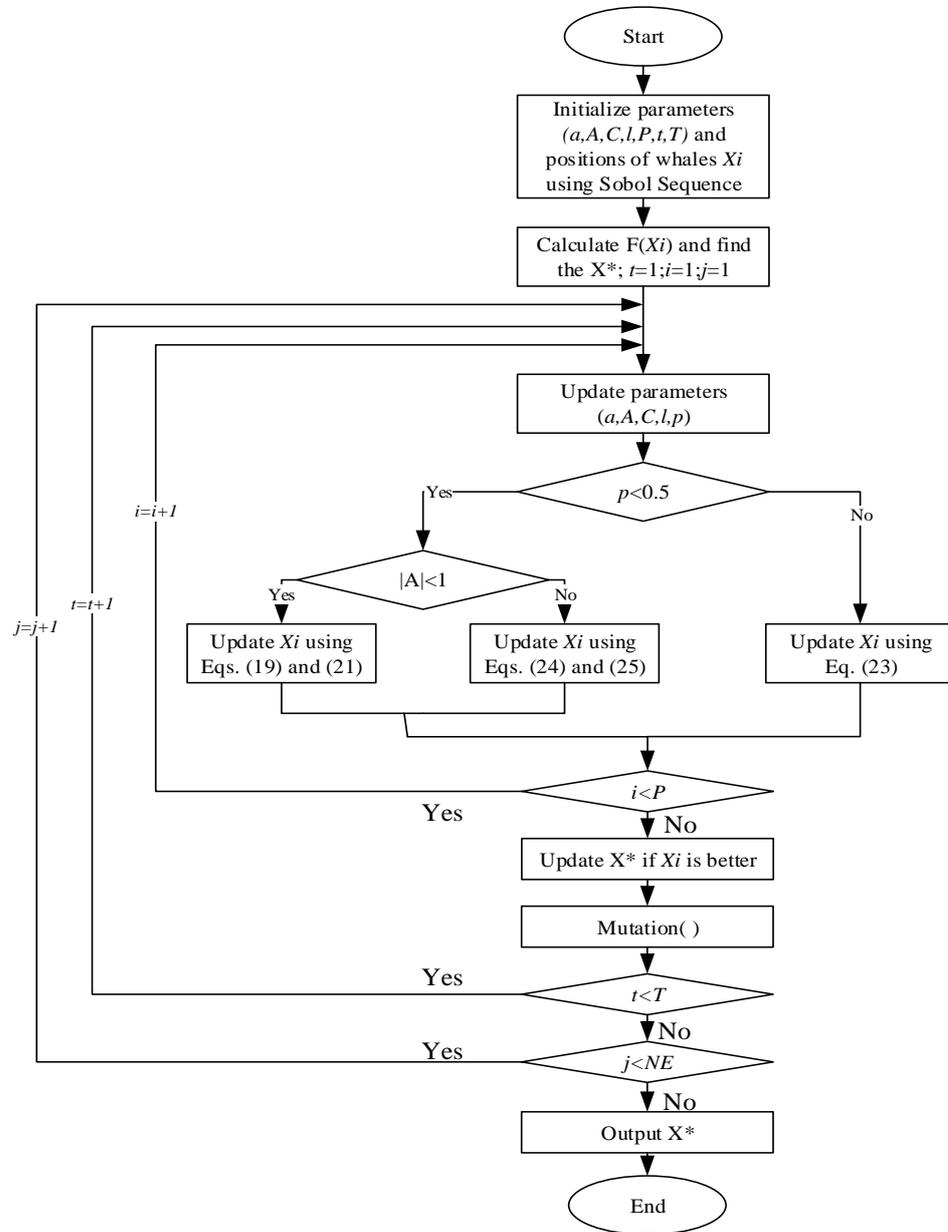


Figure 11. The workflow of Algorithm A3 of the IWOA.

4.6. Simulation Model

Figure 12 shows a timed predicate/transition net as a discrete-event simulation model for dealing with the dual-command crane scheduling problem. The main components of a timed predicate/transition net include the predicate, transition, and directed arc [42,43]. They are defined as follows:

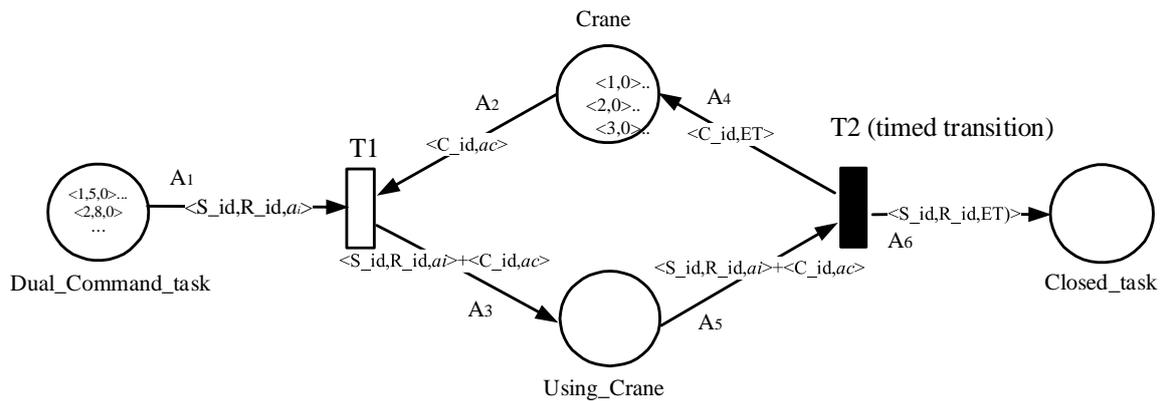


Figure 12. A discrete-event simulation model of timed predicate/transition net.

- Predicate = {Dual_Command_task, Crane, Using_Crane, Closed_task};
- Transition = {T1,T2};
- Arc = {A1, . . . , A6}.

Each predicate accompanied by a circle symbol asserts a fact. For example, the **Dual_command_task** predicate asserts the truth/false of the existence of dual-command task(s). It becomes true once a task token $\langle S_id,R_id,a_i \rangle$ appears in the corresponding circle symbol. The token’s attributes correspond to the storage_request_id, retrieval_request_id, and available_time of the request i , respectively. The **Crane** predicate asserts the truth/false of the existence of available crane(s). It becomes true once a crane token $\langle C_id,a_c \rangle$ appears in the corresponding circle symbol. The token’s attributes correspond to the crane_id and available_time of the crane c , respectively. The predicate **Closed_task** asserts the truth/false of the existence of completed task(s). It becomes true whenever a task token appears in the corresponding circle symbol. T1 and T2 are transitions that are represented by bar symbols. Transition T1 is “firable” when both the **Dual_Command_task** and **Crane** predicates become true simultaneously. T2 is firable once a combined token $\langle S_id,R_id,a_i \rangle + \langle C_id,a_c \rangle$ appears in the **Using_Crane** predicate. T2 is a timed transition as it takes time to complete the transition. After this transition, the crane token $\langle C_id,ET \rangle$ is returned to the **Crane** predicate; meanwhile, the task token $\langle S_id,R_id,ET \rangle$ is transitioned to the **Closing_task** predicate. When all tokens stay with the **Closing_task** predicate, the simulation stops running.

Let a_i and a_c be the available times of the SKU i ($i \in T$) and the crane c , respectively. Then, the start time of handling the i -th paired requests (ST_i) is determined by Equation (33).

$$ST_i = \text{Max}\{a_i, a_c\} \tag{33}$$

Let $T(DC)_{ii'}$ be the travel time for this pair of storage and retrieval requests. Equation (34) is used to estimate the completion time.

$$ET_{i'} = ST_i + T(DC)_{ii'} = \text{Max}\{a_i, a_r\} + T(DC)_{ii'} \tag{34}$$

5. Numerical Experiments

This section evaluates the effectiveness of FCFS, RANDOM, GA, PSO, WOA, and IWOA+DP used as sequencing methods in the simulation-based optimization framework. First, a small-size instance is used to demonstrate the result. Then, extensive experiments are used to conduct a thorough evaluation. These experiments were performed on a computer with Intel PENTIUM CPU 2117U (64 bits and 1.8 GHz) and 4 GB DRAMs. Java was used as the programming language to implement these approaches.

5.1. Parameter Value Settings for Experiments

After sorting out the requests, the requests for SKUs to be handled on a specific crane will be found. For experiments, we set the number of cranes to one and the number of racks to two ($R = \{1, 2\}$, where the R is a set of racks served by the crane). Referring to Nia et al. [27], $v_x = 4$ (meters/s), $v_y = 0.9$ (meters/s), $v_z = 4$ (meters/s), $P_c = 1172$ (k W/s), $C_e = 0.1$, and $f_e = 1.508 \times 10^5$ are used. In addition, we set the dimension of each rack to be $(X, Y, Z) = (40, 30, 2)$, indicating that there are $40 \times 30 \times 2$ storage bins. The dimensions of each bin are: $W = 1.15$ m (width), $H = 1.32$ m (height), and $D = 1.5$ m (depth). For the SKUs at the second depth of a rack, parameter values $C_s = C_r = 2.5$ are used.

Table 1 shows the parameter values set for different approaches, in which N indicates the total number of storage/retrieval requests; P indicates the total number of individuals in the population; I indicates the total number of iterations; \bar{V} indicates the upper limit of velocity; \underline{V} indicates the lower limit of velocity; Rm indicates a mutation rate for GA; Rc indicates a crossover rate for the GA; ω indicates the mutation rate for the IWOA; θ indicates the threshold for using IM or SM; NE indicates the number of elites in the elite set ES; - indicates not used.

Table 1. The parameter setting for different approaches.

Methods Parameters	FCFS	RANDOM	PSO	GA	WOA	IWOA
N	50, 100, 150	50, 100, 150	50, 100, 150	50, 100, 150	50, 100, 150	50, 100, 150
P	-	-	60	60	60	60
I	1	500	500	500	500	500
w	-	-	0.5	-	-	-
\bar{V}	-	-	2	-	-	-
\underline{V}	-	-	2	-	-	-
Rm	-	-	-	0.3	-	-
Rc	-	-	-	0.4	-	-
ω	-	-	-	-	-	0.5
θ	-	-	-	-	-	0.5
NE	-	-	-	-	-	3

5.2. Experimental Instance Generation

Based on the parameter value settings, a computer is used to generate random storage and request services. This simulates the randomness of real operations. A total number of N storage and retrieval requests of SKUs and the target position of these SKUs are generated. The problem size is defined as $N \times N$.

5.3. An Example of a Small Instance

A small-sized experimental instance ($N \times N = 15 \times 15$) is used to illustrate the obtained results.

Based on the parameter values $(X, Y, Z) = (40, 30, 2)$ and $RK = \{1, 2\}$ set in Section A, experimental instances are automatically generated by the computer for experiments. Table 2a,b show the storage and retrieval requests automatically generated by the computer, respectively. In the two tables, S_id indicates a storage request ID; R_id indicates a retrieval request ID; r indicates a rack number $r \in R$; x indicates a column number; y indicates a tier number; z indicates a depth number of an SKU.

Table 2. (a). The storage requests ($N = 15$). (b). The retrieval requests ($N = 15$).

(a)															
S_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
r	1	2	1	1	1	2	2	1	1	1	2	1	1	1	2
x	39	14	26	33	35	10	2	35	19	1	8	24	30	26	19
y	12	5	20	9	19	17	21	10	13	24	10	9	14	30	21
z	1	1	1	2	2	1	1	1	2	1	1	1	1	1	1
(b)															
R_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
r	1	1	1	2	2	1	2	1	1	1	1	2	2	1	1
x	23	10	6	1	6	39	8	4	23	12	17	9	14	21	14
y	26	8	28	29	6	24	12	17	22	14	26	14	5	30	29
z	1	2	2	1	1	1	2	1	1	2	2	1	1	2	1

Different approaches are then used to find the best solution. Table 3a–f show the experimental results obtained by (a) FCFS, (b) RANDOM, (c) the WOA, (d) the GA, (e) PSO, and (f) the IWOA+DP.

Table 3. (a). The best solution found by FCFS (problem size: $N \times N = 15 \times 15$). (b). The best solution found by RANDOM (problem size: $N \times N = 15 \times 15$). (c). The best solution found by WOA (problem size: $N \times N = 15 \times 15$). (d). The best solution found by GA (problem size: $N \times N = 15 \times 15$). (e). The best solution found by PSO (problem size: $N \times N = 15 \times 15$). (f). The best solution found by the IWOA+DP (problem size: $N \times N = 15 \times 15$).

(a)															
S_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
T1	17.6	7.3	29.3	13.2	27.9	24.9	30.8	14.7	19.1	35.2	14.7	13.2	20.5	44	30.8
ts	0.8	0.8	0.8	3.8	3.8	0.8	0.8	0.8	3.8	0.8	0.8	0.8	0.8	0.8	0.8
T2	20.5	4.4	11.7	29.3	19.1	10.3	13.2	10.3	13.2	14.7	23.5	7.3	13.2	1.4	11.7
tr	0.8	3.8	3.8	0.8	0.8	0.8	3.8	0.8	0.8	3.8	3.8	0.8	0.8	3.8	0.8
T3	38.1	11.7	41.1	42.5	8.8	35.2	17.6	24.9	32.3	20.5	38.1	20.5	7.3	44	42.5
T(DC)	77.8	28	86.6	89.6	60.2	71.9	66.1	51.4	69	74.9	80.8	42.6	42.6	93.9	86.6
TPT	77.8	105.7	192.4	281.9	342.2	414.1	480.2	531.5	600.6	675.5	756.2	798.8	841.4	935.3	1021.9
(b)															
S_id	1	5	15	6	11	7	10	3	8	13	2	12	14	9	4
R_id	12	8	2	9	1	15	14	4	5	10	7	11	6	13	3
T1	17.6	27.9	30.8	24.9	14.7	30.8	35.2	29.3	14.7	20.5	7.3	13.2	44	19.1	13.2
ts	0.8	3.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	3.8	3.8
T2	8.6	8.9	19.1	7.3	23.5	11.7	8.8	13.2	8.3	5.2	10.3	24.9	8.8	11.7	27.9
tr	0.8	0.8	3.8	0.8	0.8	0.8	3.8	0.8	0.8	3.8	3.8	0.8	0.8	0.8	3.8
T3	20.5	24.9	11.7	32.3	38.1	42.5	44	42.5	8.8	20.5	17.6	38.1	35.2	7.3	41.1
T(DC)	48.3	66.2	66.1	66	77.8	86.6	92.5	86.6	33.3	50.7	39.7	80.8	89.5	42.6	89.6
TPT	48.3	114.5	180.6	246.6	324.4	410.9	503.4	590	623.3	674	713.8	794.5	884	926.6	1016.3
(c)															
S_id	8	4	10	2	15	5	14	13	12	1	6	9	11	7	3
R_id	14	12	11	2	8	9	15	7	6	4	13	1	5	10	3
T1	14.7	13.2	35.2	7.3	30.8	27.9	44	20.5	13.2	17.6	24.9	19.1	14.7	30.8	29.3
ts	0.8	3.8	0.8	0.8	0.8	3.8	0.8	0.8	0.8	0.8	0.8	3.8	0.8	0.8	0.8
T2	29.3	7.3	4.6	4.4	5.9	4.4	3.4	6.3	22	24.9	17.6	19.1	5.9	10.3	11.7
tr	3.8	0.8	3.8	3.8	0.8	0.8	0.8	3.8	0.8	0.8	0.8	0.8	0.8	3.8	3.8
T3	44	20.5	38.1	11.7	24.9	32.3	42.5	17.6	35.2	42.5	7.3	38.1	8.8	20.5	41.1
T(DC)	92.5	45.6	82.4	28	63.1	69	91.5	49	71.9	86.6	51.4	80.8	30.8	66.1	86.6
TPT	92.5	138.1	220.5	248.5	311.6	380.6	472.1	521	592.9	679.5	730.9	811.6	842.5	908.6	995.2

Table 3. Cont.

(d)															
S_id	12	11	9	3	10	2	15	13	14	4	6	7	1	5	8
R_id	3	1	7	12	14	8	9	15	11	5	10	6	4	2	13
T1	13.2	14.7	19.1	29.3	35.2	7.3	30.8	20.5	44	13.2	24.9	30.8	17.6	27.9	14.7
ts	0.8	0.8	3.8	0.8	0.8	0.8	0.8	0.8	0.8	3.8	0.8	0.8	0.8	3.8	0.8
T2	27.9	23.5	3.2	8.8	8.8	17.6	1.5	22	5.9	7.8	4.4	10.6	24.9	16.1	7.3
tr	3.8	0.8	3.8	0.8	3.8	0.8	0.8	0.8	3.8	0.8	3.8	0.8	0.8	3.8	0.8
T3	41.1	38.1	17.6	20.5	44	24.9	32.3	42.5	38.1	8.8	20.5	35.2	42.5	11.7	7.3
T(DC)	86.6	77.8	47.3	60.2	92.5	51.4	66	86.6	92.5	34.3	54.4	78.1	86.6	63.2	30.8
TPT	86.6	164.4	211.7	271.9	364.4	415.8	481.8	568.4	660.9	695.1	749.5	827.6	914.2	977.4	1008.3

(e)															
S_id	12	8	9	14	11	3	2	6	5	15	13	7	1	10	4
R_id	7	5	4	15	12	6	10	13	3	11	2	1	8	14	9
T1	13.2	14.7	19.1	44	14.7	29.3	7.3	24.9	27.9	30.8	20.5	30.8	17.6	35.2	13.2
ts	0.8	0.8	3.8	0.8	0.8	0.8	0.8	0.8	3.8	0.8	0.8	0.8	0.8	0.8	3.8
T2	4.6	8.3	23.5	3.4	5.9	5.9	13.2	17.6	13.2	7.3	8.8	7.3	10.1	8.8	19.1
tr	3.8	0.8	0.8	0.8	0.8	0.8	3.8	0.8	3.8	3.8	3.8	0.8	0.8	3.8	0.8
T3	17.6	8.8	42.5	42.5	20.5	35.2	20.5	7.3	41.1	38.1	11.7	38.1	24.9	44	32.3
T(DC)	39.9	33.3	89.6	91.5	42.6	71.9	45.6	51.4	89.6	80.8	45.6	77.8	54.1	92.5	69
TPT	39.9	73.2	162.8	254.3	296.8	368.7	414.3	465.7	555.3	636.1	681.6	759.4	813.5	906	975

(f)															
S_id	11	2	1	3	13	9	14	10	4	15	8	6	7	5	12
R_id	7	13	6	9	1	10	14	4	2	11	12	8	3	15	5
T1	14.7	7.3	17.6	29.3	20.5	19.1	44	35.2	13.2	30.8	14.7	24.9	30.8	27.9	13.2
ts	0.8	0.8	0.8	0.8	0.8	3.8	0.8	0.8	3.8	0.8	0.8	0.8	0.8	3.8	0.8
T2	2.9	0	17.6	2.9	17.6	2	1.4	7.3	6.6	7.3	7.5	1.7	10.3	14.7	5.2
tr	3.8	0.8	0.8	0.8	0.8	3.8	3.8	0.8	3.8	3.8	0.8	0.8	3.8	0.8	0.8
T3	17.6	7.3	35.2	32.3	38.1	20.5	44	42.5	11.7	38.1	20.5	24.9	41.1	42.5	8.8
T(DC)	39.7	16.2	71.9	66	77.8	49.1	93.9	86.6	39	80.8	44.2	53.1	86.6	89.6	28.7
TPT	39.7	55.9	127.8	193.8	271.6	320.7	414.6	501.2	540.2	621	665.2	718.3	804.9	894.5	923.1

In Table 3(a–f), S_id indicates a storage request ID; R_id indicates a retrieval request ID; T1 indicates the travel time to the storage location; it indicates the storage shuttle time; T2 indicates the travel time from the storage location to the retrieval location; tr indicates the retrieval shuttle time; T3 indicates the travel time from the retrieval location to the I/O station; T(DC) indicates the total process time of each dual command; TPT indicates the accumulated total time of this sequence of dual commands.

Table 4 summarizes the experimental results. It is found that the IWOA finds the best solution with $Z = 1.63E+10$ at the computation cost of 3.476 s. The WOA finds the best solution with $Z = 1.76E+10$ at the computation cost of 2.26 s; the PSO finds the best solution with $Z = 1.72E+10$ at the computation cost of 2.25 s; the GA finds the solution with $Z = 1.78E+10$ at the computation cost of 1.78 s; the Random heuristic finds the best solution with $Z = 1.80E+10$ at the computation cost of 1.98 s; the FCFS finds the best solution with $Z = 1.81E+10$ at the computation cost of 0.06 s. This case shows that the IWOA outperforms the others in terms of Z value.

Table 4. The summary of different approaches.

	FCFS	Random	GA	WOA	PSO	IWOA
CPU	0.06	1.98	1.78	2.226	2.25	3.476
TPT	1021.9	1016.3	1008.3	995.0	975	923.1
Z	1.81E+10	1.80E+10	1.78E+10	1.76E+10	1.72E+10	1.63E+10

5.4. Extensive Experiments using Different Problem Sizes

Extensive experiments using different problem sizes have been further conducted to investigate the effectiveness of different approaches. Table 5 shows the experimental results, which are summarized are follows:

Table 5. The results obtained from different approaches using different problem sizes.

	FCFS			RANDOM			GA			WOA			PSO			IWOA+DP		
	Z	T	G	Z	T	G	Z	T	G	Z	T	G	Z	T	G	Z	T	
50 × 50																		
1	5.67E+10	0.1	24.75%	5.20E+10	16.8	14.54%	5.30E+10	3.3	16.68%	5.22E+10	7.3	14.91%	5.19E+10	0.8	14.31%	4.54E+10	63.0	
2	5.79E+10	0.1	25.38%	5.46E+10	16.4	18.32%	5.36E+10	3.4	16.17%	5.43E+10	7.3	17.57%	5.40E+10	0.76	16.84%	4.62E+10	63.5	
3	5.98E+10	0.1	24.96%	5.56E+10	17.5	16.19%	5.59E+10	3.6	16.68%	5.61E+10	7.3	17.21%	5.57E+10	0.799	16.29%	4.79E+10	67.7	
4	5.77E+10	0.1	13.59%	5.99E+10	16.5	17.79%	5.66E+10	3.4	11.43%	5.65E+10	7.3	11.22%	5.61E+10	0.821	10.54%	5.08E+10	62.3	
5	5.66E+10	0.1	23.17%	5.38E+10	16.6	17.02%	5.41E+10	3.5	17.67%	5.34E+10	7.2	16.14%	5.34E+10	0.827	16.09%	4.60E+10	64.8	
6	5.93E+10	0.1	20.13%	5.72E+10	16.6	15.23%	5.70E+10	3.5	15.48%	5.72E+10	7.2	15.73%	5.72E+10	0.81	15.88%	4.94E+10	60.4	
7	5.53E+10	0.1	17.39%	5.40E+10	16.5	14.72%	5.50E+10	3.3	16.88%	5.33E+10	7.4	13.23%	5.34E+10	0.952	13.46%	4.71E+10	64.6	
8	5.62E+10	0.1	19.12%	5.41E+10	16.3	14.51%	5.47E+10	3.4	15.90%	5.41E+10	7.3	14.58%	5.47E+10	0.805	15.83%	4.72E+10	61.8	
9	5.87E+10	0.1	17.79%	5.69E+10	16.6	14.07%	5.62E+10	3.3	12.78%	5.72E+10	7.1	14.77%	5.74E+10	0.777	15.12%	4.99E+10	71.2	
10	6.02E+10	0.1	21.15%	5.73E+10	16.6	15.23%	5.67E+10	3.4	14.05%	5.75E+10	7.1	15.71%	5.70E+10	0.862	14.68%	4.97E+10	61.0	
Avg.	5.79E+10	0.1	20.65%	5.55E+10	16.6	15.83%	5.53E+10	3.4	15.35%	5.52E+10	7.2	15.11%	5.51E+10	0.8213	14.88%	4.80E+10	64.0	
Min	5.53E+10	0.1	13.59%	5.20E+10	16.3	14.07%	5.30E+10	3.3	11.43%	5.22E+10	7.1	11.22%	5.19E+10	0.76	10.34%	4.54E+10	60.4	
Max	6.02E+10	0.1	25.38%	5.99E+10	17.5	18.32%	5.70E+10	3.6	17.67%	5.75E+10	7.4	17.57%	5.74E+10	0.952	16.84%	5.08E+10	71.2	
100 × 100																		
1	1.09E+11	0.2	24.86%	1.06E+11	38.8	21.13%	1.05E+11	6.0	20.19%	1.05E+11	13.4	19.73%	1.05E+11	1.225	20.55%	8.74E+10	154.1	
2	1.11E+11	0.2	18.30%	1.11E+11	38.9	18.94%	1.09E+11	5.9	16.42%	1.10E+11	13.8	17.72%	1.11E+11	1.225	18.17%	9.37E+10	149.9	
3	1.16E+11	0.2	23.51%	1.11E+11	38.1	18.29%	1.10E+11	6.3	17.40%	1.10E+11	13.5	16.91%	1.12E+11	1.262	19.35%	9.41E+10	148.2	
4	1.18E+11	0.2	20.48%	1.13E+11	38.8	16.26%	1.15E+11	6.1	17.65%	1.14E+11	13.4	16.77%	1.14E+11	1.226	16.74%	9.76E+10	147.0	
5	1.12E+11	0.2	21.41%	1.08E+11	39.0	16.61%	1.09E+11	6.3	18.42%	1.09E+11	13.7	18.08%	1.09E+11	1.229	17.58%	9.24E+10	142.6	
6	1.13E+11	0.2	23.43%	1.10E+11	38.7	20.19%	1.10E+11	6.1	19.96%	1.10E+11	14.7	19.81%	1.02E+11	1.236	11.59%	9.14E+10	147.6	
7	1.17E+11	0.2	22.55%	1.18E+11	40.3	24.08%	1.12E+11	6.3	17.85%	1.13E+11	13.6	18.90%	1.13E+11	1.29	18.72%	9.52E+10	161.6	
8	1.09E+11	0.2	21.08%	1.06E+11	40.1	18.51%	1.08E+11	6.1	20.45%	1.07E+11	14.0	18.76%	1.08E+11	1.246	20.33%	8.98E+10	160.9	
9	1.14E+11	0.2	21.76%	1.12E+11	40.4	19.61%	1.11E+11	6.1	18.39%	1.13E+11	13.9	20.05%	1.12E+11	1.231	19.65%	9.39E+10	164.9	
10	1.10E+11	0.2	20.21%	1.06E+11	47.5	16.48%	1.05E+11	6.2	15.24%	1.05E+11	13.9	15.34%	1.05E+11	1.234	15.38%	9.12E+10	170.0	
Avg.	1.13E+11	0.2	21.76%	1.10E+11	40.0	19.00%	1.10E+11	6.1	18.20%	1.10E+11	13.8	18.21%	1.09E+11	1.2	17.81%	9.27E+10	154.7	
Min	1.09E+11	0.2	18.30%	1.06E+11	38.1	16.26%	1.05E+11	5.9	15.24%	1.05E+11	13.4	15.34%	1.02E+11	1.225	11.59%	8.74E+10	142.6	
Max	1.18E+11	0.2	24.86%	1.18E+11	47.5	24.08%	1.15E+11	6.3	20.45%	1.14E+11	14.7	20.05%	1.14E+11	1.3	20.55%	9.76E+10	170.0	
150 × 150																		
1	1.63E+11	0.3	23.92%	1.57E+11	66.2	19.18%	1.59E+11	8.9	20.90%	1.58E+11	21.2	20.49%	1.57E+11	1.774	19.42%	1.32E+11	290.7	
2	1.73E+11	0.4	22.75%	1.69E+11	62.9	19.59%	1.68E+11	8.9	19.46%	1.70E+11	21.4	20.50%	1.69E+11	1.756	19.96%	1.41E+11	282.6	
3	1.75E+11	0.3	22.30%	1.71E+11	63.8	19.24%	1.71E+11	9.2	19.64%	1.70E+11	22.3	18.86%	1.71E+11	1.77	19.58%	1.43E+11	266.3	
4	1.62E+11	0.4	22.07%	1.59E+11	63.0	19.52%	1.59E+11	8.9	19.37%	1.56E+11	21.2	17.82%	1.58E+11	1.775	19.16%	1.33E+11	248.9	
5	1.67E+11	0.3	21.06%	1.65E+11	67.1	19.64%	1.65E+11	8.8	19.32%	1.65E+11	21.1	19.22%	1.65E+11	1.834	19.78%	1.38E+11	288.1	
6	1.72E+11	0.5	23.57%	1.70E+11	72.5	21.87%	1.68E+11	12.8	20.44%	1.67E+11	25.9	19.81%	1.67E+11	2.518	19.80%	1.39E+11	574.1	
7	1.71E+11	0.3	20.29%	1.69E+11	65.0	18.99%	1.68E+11	9.1	18.46%	1.68E+11	20.1	18.29%	1.68E+11	1.78	18.51%	1.42E+11	250.9	
8	1.62E+11	0.3	19.26%	1.62E+11	62.3	19.22%	1.61E+11	8.9	18.31%	1.62E+11	20.3	18.71%	1.61E+11	1.733	18.56%	1.36E+11	243.6	
9	1.74E+11	0.4	22.81%	1.70E+11	63.2	19.90%	1.68E+11	9.7	18.95%	1.69E+11	24.4	19.23%	1.69E+11	1.803	19.68%	1.42E+11	251.7	
10	1.67E+11	0.3	23.90%	1.64E+11	65.9	21.98%	1.64E+11	9.5	21.92%	1.65E+11	21.4	22.78%	1.63E+11	1.76	21.43%	1.35E+11	243.3	
Avg.	1.69E+11	0.4	22.19%	1.66E+11	65.2	19.91%	1.65E+11	9.4	19.68%	1.65E+11	21.9	19.57%	1.65E+11	1.85	19.59%	1.38E+11	294.0	
Min	1.62E+11	0.3	19.26%	1.57E+11	62.3	18.99%	1.59E+11	8.7	18.31%	1.56E+11	20.1	17.82%	1.57E+11	1.73	18.51%	1.32E+11	243.3	
Max	1.75E+11	0.5	23.92%	1.71E+11	72.5	21.98%	1.71E+11	12.8	21.92%	1.70E+11	25.9	22.78%	1.71E+11	2.52	21.43%	1.43E+11	574.1	

Z: the objective function value; T: time (s); G: gap to the IWOA+DP.

- At the problem size 50 × 50, the IWOA+DP has average edges of 16.84%, 17.57%, 17.67%, 18.32%, and 25.38% over the PSO, WOA, GA, RANDOM, and FCFS, respectively. Figure 13 shows the box-whisker plot of the Z values of the problem size 50 × 50 obtained using different approaches.

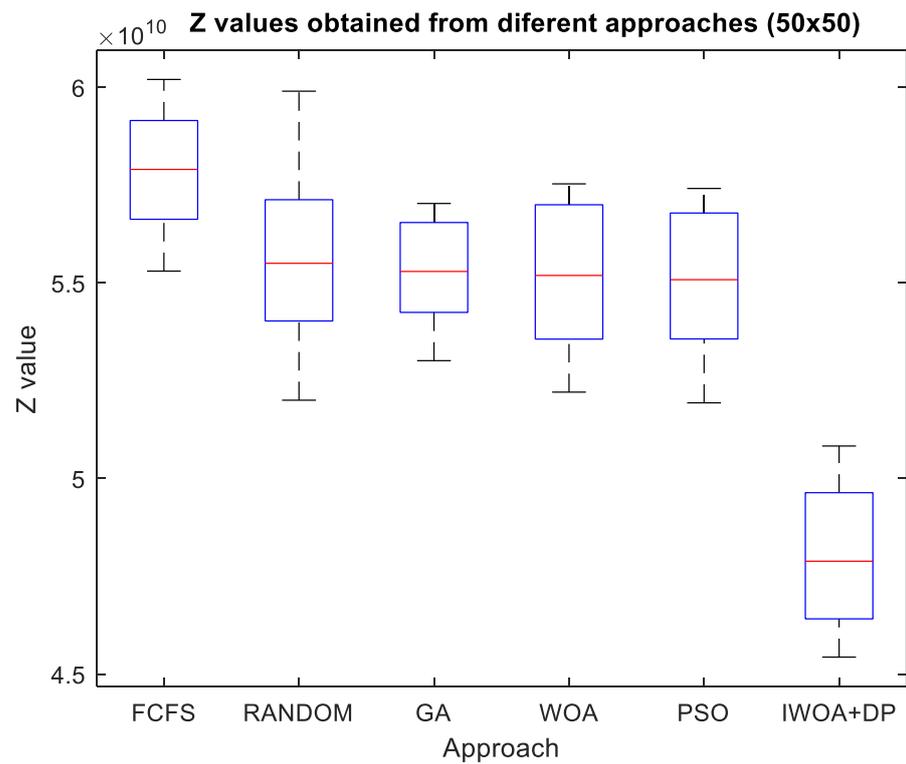


Figure 13. The box-whisker plot chart of Z values found for the problems (50 × 50).

- At the problem size 100 × 100, the IWOA+DP has average edges of 20.55%, 20.05%, 20.45%, 24.08%, and 24.86% over the PSO, WOA, GA, RANDOM, and FCFS, respectively. Figure 14 shows the box-whisker plot of the Z values of the problem size 100 × 100 obtained using different approaches.

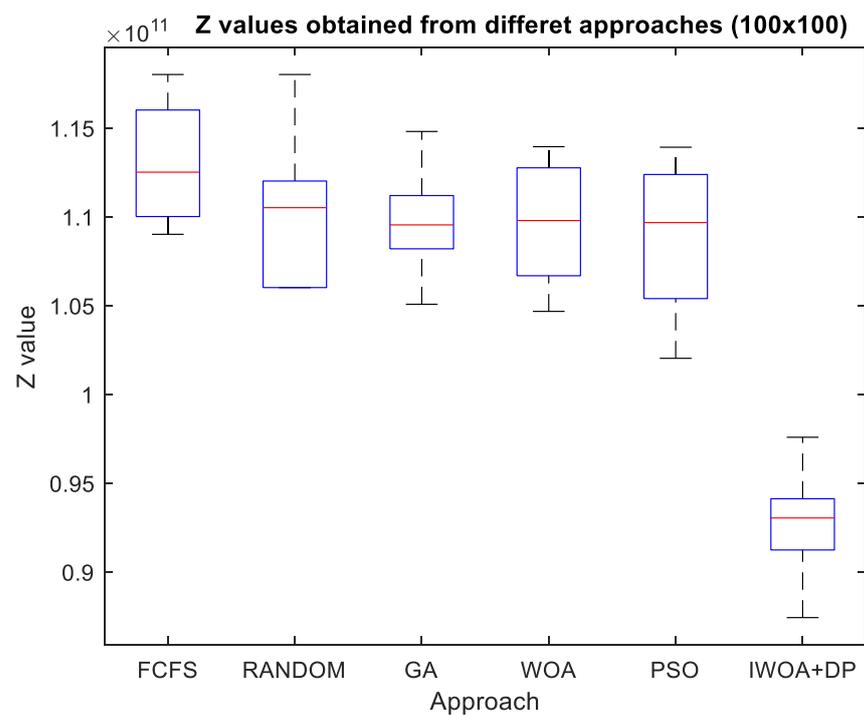


Figure 14. The box-whisker plot chart of Z values found for the problems (100 × 100).

- At the problem size 150×150 , the IWOA+DP has average edges of 21.43%, 22.78%, 21.92%, 21.98%, and 23.92% over the PSO, WOA, GA, RANDOM, and FCFS, respectively. Figure 15 shows the box-whisker plot of the Z values of the problem size 150×150 obtained using different approaches.

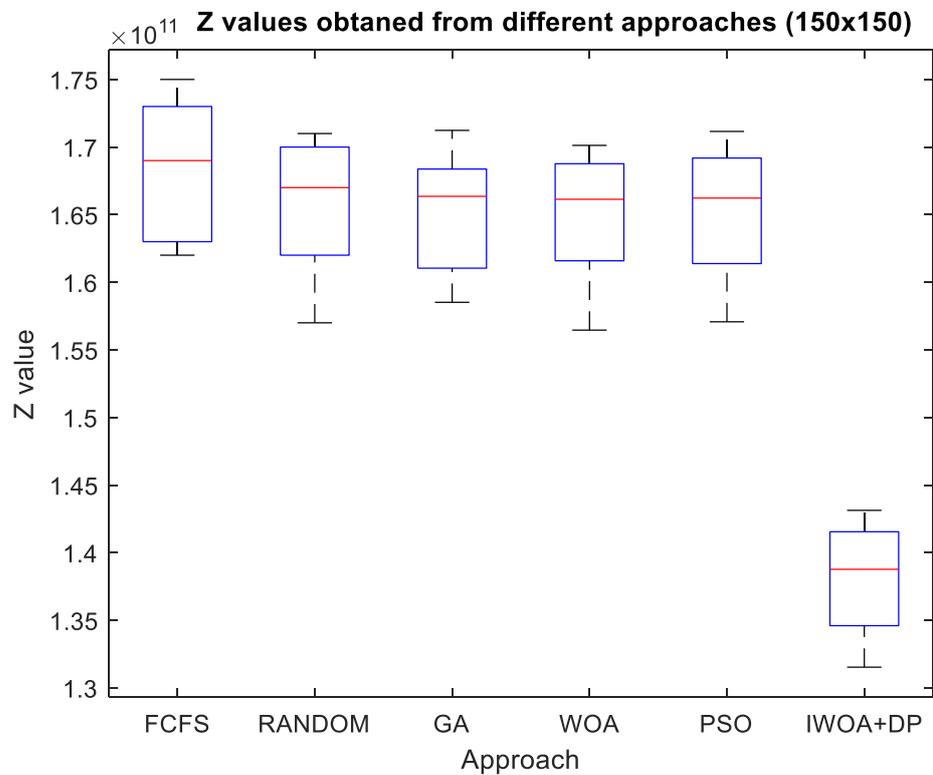


Figure 15. The box-whisker plot chart of Z values found for the problems (150×150).

- Experiments with different problem sizes consistently show that IWOA+DP outperforms the other approaches.

5.5. Statistical Tests

At the significance level of $\alpha = 0.05$, the statistical *t*-test is used to test Hypotheses H0 and H1, where the H0 assumes that the average Z values of comparing approaches have no significant difference; H1 assumes that these average Z values are different.

Table 6 shows the *t*-test results of each approach compared to the IWOA. The symbol “+” indicates that the IWOA is better; the symbol “-” indicates that the IWOA is inferior; the symbol “N” indicates no difference. The *p*-values of these comparisons are found to be less than 0.005. For example, at the problem size $N = 100$, the *p*-values of RANDOM, FCFS, the GA, PSO, and the WOA are 2.13695E-11, 1.48414E-09, 2.07195E-11, 5.69295E-09, and 2.08357E-11, respectively. The results lead to the rejection of H0 and the acceptance of H1. Thus, IWOA is confirmed to outperform the others in terms of the statistical *t*-test.

Table 6. Results of *t*-test for different approaches at different problem sizes.

Problem Size	FCFS		RANDOM		GA		WOA		PSO		IWOA Avg. Z
	Avg. Z	<i>t</i> -Test									
$N = 50$	5.79E+10	+	5.55E+10	+	5.53E+10	+	5.52E+10	+	5.51E+10	+	4.80E+10
$N = 100$	1.13E+11	+	1.10E+11	+	1.10E+11	+	1.10E+11	+	1.09E+11	+	9.27E+10
$N = 150$	1.69E+11	+	1.66E+11	+	1.65E+11	+	1.65E+11	+	1.65E+11	+	1.38E+11

5.6. Analysis and Discussion

1. The experiments demonstrate the feasibility of using closely coupled simulation-based optimization approaches to deal with the dual-command crane scheduling problem in a unit-load and double-deep AS/RS. The use of dual commands to operate an AS/RS is not difficult, especially in a busy AS/RS environment. For a high-workload AS/RS, we can set a higher value for N . For a low-workload AS/RS, we can lower the value for the N . The use of dual-command operation is advantageous for a crane.
2. On average, metaheuristics have a better performance than simple heuristics (Random and FCFS) in terms of solution quality due to the use of a guided search, instead of a random search.
3. In general, the greater the N , the higher the advantage for the IWOA+DP.
4. Though other metaheuristics such as GAs, PSO, and FAs are also population- and evolution-based metaheuristics, they cannot outperform the IWOA+DP.
5. Metaheuristics in general can achieve a better solution than the heuristics (RANDOM and FCFS) due to the use of an informed search. In contrast, RANDOM uses a kind of random search.
6. In this research, the RANDOM outperforms FCFS due to being empowered with an iterative mechanism that gives RANDOM more chances to find a better solution. FCFS is one special case of the RANDOM search. However, the two heuristics cannot outperform the other metaheuristics.
7. Though the standard WOA provides an exploration mechanism when $|A| \geq 1$, this exploration scope is still affected by the whales in the swarm. In the IWOA, a global mutation capability is additionally provided by the *mutation()* procedure. In addition, the IWOA uses an adaptive movement, in addition to working with DP.
8. In this research, the GA, WOA, and PSO generate positions for a population with real numbers, and then the ROV technique is used to transform the positions into solutions in the discrete domain of sequence numbers. The transformation procedure is used by these approaches during the whole solution process. However, the IWOA is different. After transforming the initial positions of the whales from the continuous domain to the discrete domain by using the ROV technique, the following evolutions of solutions are based on the discrete operator, as stated in Equations (26)–(28). The use of discrete operators appears to result in a better result as the crane scheduling problem is in essence in the discrete domain. Though the ROV transformation procedure is necessary to develop a feasible solution, the use of discrete operators to evolve solutions in a discrete domain appears to lead to a better result.
9. Szczepanski et al. [44] highlighted two common ways of handling constraints. One is the inclusion of the penalty function into the objective function and another is the separation of the objective function and constraints. The former decreases the fitness of an infeasible solution so as to find a feasible solution. However, in this research, we use the ROV technique, and this technique can always return feasible solutions to our focused scheduling problem. Thus, there is no need to use the penalty function for constraints in the objective function. The solutions transformed by the ROV techniques are always subject to the constraints described in Section 3.2.
10. Several intelligent algorithms have been proposed by researchers to deal with multiple objective optimization problems (MOPs), such as in [45–47]. Fountas et al. [45] proposed a virus-evolutionary genetic algorithm (GA) to optimize the tool path parameters in the sculptured surface. This algorithm is a variant of the standard GA. Rahman et al. [46] proposed a multi-objective learner performance-based behavior (MOLPB) algorithm for dealing with real-world engineering optimization problems. Zhang et al. [47] proposed an SDNSGA-III algorithm to deal with the dynamic multi-objective optimization problems (DMOPs) that have dynamic objectives and constraints in a changing environment. Our preliminary survey shows that there is a possibility of comparing the three algorithms with the IWOA proposed in this research if the ROV technique is applied to the three algorithms. This can be treated as a future research

direction. In addition, these studies show that a multi-objective function can be used as a performance measurement of a crane in future research.

6. Conclusions

The crane scheduling problem affects the performance of an AS/RS considerably. This research first defines the dual-command crane scheduling problem in a unit-load double-deep AS/RS. Following this, a MILP is formulated for this problem, with the objective of minimizing energy consumption. To couple the simulation and optimization closely, we have proposed a framework for developing simulation-based optimization approaches in which heuristics/metaheuristics, including FCFS, RANDOM, an IWOA, an WOA, a GA, and PSO, have been used as alternative sequencing methods. Experiments have been conducted to investigate their effectiveness. The experimental results show that the IWOA outperforms other heuristics/metaheuristics.

The contributions of this research are summarized as follows: (1) a MILP is formulated for this problem; (2) a framework is proposed to couple the simulation and optimization closely; (3) an improved WOA (IWOA) with features of adaptive movement and mutation is developed for solving discrete problems using a discrete operator; (4) experiments were conducted to investigate the effectiveness of different simulation-based optimization approaches; (5) the successful application of the proposed approaches to solve the dual-command crane scheduling problem in a unit-load double-deep AS/RS.

Future research can focus on the following directions. First, the assumption of the same number of storage and retrieval requests made in this research can be relaxed. Second, other metaheuristics, such as those proposed in [45–47], can be compared with multi-objective functions. Third, the allocation policies can be incorporated. Fourth, this approach can be applied to other kinds of AS/RSs. Lee [36] highlighted that AS/RSs are only used as an alternative to traditional systems but are also a part of the manufacturing system to support manufacturing.

Author Contributions: Conceptualization, H.-P.H.; Data curation, H.-P.H.; Formal analysis, C.-N.W.; Funding acquisition, H.-P.H.; Investigation, C.-N.W.; Methodology, H.-P.H.; Project administration, C.-N.W.; Software, H.-P.H.; Validation, T.-T.D.; Writing—original draft, H.-P.H.; Writing—review and editing, T.-T.D. All authors have read and agreed to the published version of the manuscript.

Funding: The authors received no specific funding for this study.

Data Availability Statement: Not applicable.

Acknowledgments: The authors appreciate the support from the National Kaohsiung University of Science and Technology, Taiwan; and Hong Bang International University, Vietnam.

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

Appendix A

Algorithm A1. The pseudo-code of the WOA

Define the problem // for a minimize problem
 Define the objective function $F(X)$
 Set parameter T (total number of iterations) P , and b
 Initialize the a , A , C , and l
 Initialize the positions X_i of whales i ($i = 1, 2, 3, \dots, P$)
 Calculate the $F(X_i)$ for each whale i
 Find the X^*
REPEAT1 while ($t \leq T$) or (stop criterion)
 Update a , \vec{A} , \vec{C} , l , and p
REPEAT2 while ($i < P$)
IF ($p < 0.5$) // shrinking circle movement
IF ($|A| < 1$) // **exploitation**
 Update the X_i using Equations (19) and (21)
ELSE ($|A| \geq 1$) // **exploration**
 Select a random X_R from the whale group
 Update the X_i using Equations (24) and (25)
ENE IF
ELSE // **spiral-shaped movement**
 Update the X_i using Equation (23)
END IF
END REPEAT2 (until $I > P$)
 Check if any whale is out of the search space and amend it
 Evaluate $F(X_i)$ for each whale i
 Update the $X_{w^*}(t)$ if better
END REPEAT1 (until $t > T$)
 Output X^*

Appendix B

Algorithm A2. The pseudo-code of the *mutation()*

Set the mutation rate ω
 Set the mutation rate θ // threshold
 Generate the random variable $R1$ and $R2$
IF $R1 < \omega$ // perform mutation
IF $SD < \theta$
 Use IM
ELSE
 Use SM
ELSE
 No mutation
END IF

Appendix C

Algorithm A3. The pseudo-code of the IWOA

```

1   Define the objective function  $F(X)$  of the problem
2   Set parameter  $T$  (the total number of iterations);  $a$ ,  $A$ ,  $C$ ,  $l$ , and  $p$ 
3   Initialize the positions  $X_i$  of whales ( $i = 1, 2, 3, \dots, N$ ) using Sobol sequence generator
4   Calculate the  $F(X_i)$  for each whale  $i$ ; and find the  $\vec{X}_{w'}(t)$   $w' \in ES$ 
5   REPEAT1 while ( $t \leq T$ ) or (stop criterion)
6   Update parameters  $a$ ,  $\left| \vec{A} \right|$ ,  $\vec{C}$ ,  $l$ , and  $p$ 
7   REPEAT2 while ( $i \leq N$ )
8   REPEAT3 while ( $j \leq NE$ )
9   IF ( $p < 0.5$ )
10  IF ( $\left| A \right| < 1$ ) // Exploitation
11  Update the  $X_i$  using Equations (26)–(32)
12  ELSE ( $\left| A \right| \geq 1$ ) // Exploration
13  Select a random whale  $X_R$  from the whale group
14  Update the  $X_i$  using Equations (26)–(32)
15  ENE IF
16  ELSE
17  Update the  $X_i$  using Equations (26)–(32)
18  END IF
19  END REPEAT3 (until  $j > NE$ )
20  END REPEAT2 (until  $i > N$ )
21  Check and amend the whales out of the search space
22  Evaluate  $F(X_i)$  for each whale  $i$ ;
23  Update the  $\vec{X}_{w^*}(t)$  if better
24  Mutation ( )
25  END REPEAT1 (until  $t > T$ )
26  Output  $X^*$ 

```

References

- Eynan, A.; Rosenblatt, M.J. Establishing Zones In Single-Command Class-Based Rectangular As/Rs. *IIE Trans.* **1994**, *26*, 38–46. [\[CrossRef\]](#)
- Rosenblatt, M.J.; Roll, Y.; Vered Zyser, D. A Combined Optimization And Simulation Approach For Designing Automated Storage/Retrieval Systems. *IIE Trans.* **1993**, *25*, 40–50. [\[CrossRef\]](#)
- Hu, Y.-H.; Huang, S.Y.; Chen, C.; Hsu, W.-J.; Toh, A.C.; Loh, C.K.; Song, T. Travel Time Analysis of a New Automated Storage and Retrieval System. *Comput. Oper. Res.* **2005**, *32*, 1515–1544. [\[CrossRef\]](#)
- Boysen, N.; Stephan, K. A Survey on Single Crane Scheduling in Automated Storage/Retrieval Systems. *Eur. J. Oper. Res.* **2016**, *254*, 691–704. [\[CrossRef\]](#)
- Graves, S.C.; Hausman, W.H.; Schwarz, L.B. Storage-Retrieval Interleaving in Automatic Warehousing Systems. *Manag. Sci.* **1977**, *23*, 935–945. [\[CrossRef\]](#)
- Han, M.-H.; McGinnis, L.F.; Shieh, J.S.; White, J.A. On Sequencing Retrievals In An Automated Storage/Retrieval System. *IIE Trans.* **1987**, *19*, 56–66. [\[CrossRef\]](#)
- Bozer, Y.A.; White, J.A. Design and Performance Models for End-of-Aisle Order Picking Systems. *Manag. Sci.* **1990**, *36*, 852–866. [\[CrossRef\]](#)
- Karasawa, Y.; Nakayama, H.; Dohi, S. Trade-off Analysis for Optimal Design of Automated Warehouses. *Int. J. Syst. Sci.* **1980**, *11*, 567–576. [\[CrossRef\]](#)
- Zollinger, H.A. *Planning, February, Evaluating, and Estimating Storage Systems*; 1982.
- Ashayeri, J.; Gelders, L.; van Wassenhove, L. A Microcomputer-Based Optimization Model for the Design of Automated Warehouses. *Int. J. Prod. Res.* **1985**, *23*, 825–839. [\[CrossRef\]](#)
- Chang, D.-T.; Wen, U.-P. The Impact on Rack Configuration on the Speed Profile of the Storage and Retrieval Machine. *IIE Trans.* **1997**, *29*, 525–531. [\[CrossRef\]](#)
- Hwang, H.; Ko, C.S. A Study on Multi-Aisle System Served by a Single Storage/Retrieval Machine. *Int. J. Prod. Res.* **1988**, *26*, 1727–1737. [\[CrossRef\]](#)
- Lee, Y.H.; Hwan Lee, M.; Hur, S. Optimal Design of Rack Structure with Modular Cell in AS/RS. *Int. J. Prod. Econ.* **2005**, *98*, 172–178. [\[CrossRef\]](#)

14. Bozer, Y.A.; White, J.A. A Generalized Design and Performance Analysis Model for End-of-Aisle Order-Picking Systems. *IIE Trans.* **1996**, *28*, 271–280. [[CrossRef](#)]
15. Park, B.C. Optimal Dwell Point Policies for Automated Storage/Retrieval Systems with Dedicated Storage. *IIE Trans.* **1999**, *31*, 1011–1113. [[CrossRef](#)]
16. Koh, S.-G.; Kwon, H.-M.; Kim, Y.-J. An Analysis of the End-of-Aisle Order Picking System: Multi-Aisle Served by a Single Order Picker. *Int. J. Prod. Econ.* **2005**, *98*, 162–171. [[CrossRef](#)]
17. van Oudheusden, D.L.; Zhu, W. Storage Layout of AS/RS Racks Based on Recurrent Orders. *Eur. J. Oper. Res.* **1992**, *58*, 48–56. [[CrossRef](#)]
18. Sarker, B.R.; Mann, L.; Leal Dos Santos, J.R.G. Evaluation of a Class-Based Storage Scheduling Technique Applied to Dual-Shuttle Automated Storage and Retrieval Systems. *Prod. Plan. Control* **1994**, *5*, 442–449. [[CrossRef](#)]
19. Mahajan, S.; Rao, B.V.; Peters, B.A. A Retrieval Sequencing Heuristic for Miniload End-of-Aisle Automated Storage/Retrieval Systems. *Int. J. Prod. Res.* **1998**, *36*, 1715–1731. [[CrossRef](#)]
20. van Oudheusden, D.L.; Tzen, Y.-J.J.; Ko, H.-T. Improving Storage and Order Picking in a Person-on-Board as/r System: A Case Study. *Eng. Costs Prod. Econ.* **1988**, *13*, 273–283. [[CrossRef](#)]
21. Goetschalckx, M.; Donald Ratliff, H. Optimal Lane Depths for Single and Multiple Products in Block Stacking Storage Systems. *IIE Trans.* **1991**, *23*, 245–258. [[CrossRef](#)]
22. Hwang, H.; Song, J.Y. Sequencing Picking Operation and Travel Time Models for Man-on-Board Storage and Retrieval Warehousing System. *Int. J. Prod. Econ.* **1993**, *29*, 75–88. [[CrossRef](#)]
23. Yang, P.; Miao, L.-X.; Qi, M. Slotting Optimization in a Multi-Shuttle Automated Storage and Retrieval System. *Comput. Integr. Manuf. Syst.* **2011**, *17*, 1050–1055.
24. Wu, K.-Y.; Xu, S.S.-D.; Wu, T.-C. Optimal Scheduling for Retrieval Jobs in Double-Deep AS/RS by Evolutionary Algorithms. *Abstr. Appl. Anal.* **2013**, *2013*, 634812. [[CrossRef](#)]
25. Popović, D.; Vidović, M.; Bjelić, N. Application of Genetic Algorithms for Sequencing of AS/RS with a Triple-Shuttle Module in Class-Based Storage. *Flex. Serv. Manuf. J.* **2014**, *26*, 432–453. [[CrossRef](#)]
26. Brezovnik, S.; Gotlih, J.; Balič, J.; Gotlih, K.; Brezočnik, M. Optimization of an Automated Storage and Retrieval Systems by Swarm Intelligence. *Procedia Eng.* **2015**, *100*, 1309–1318. [[CrossRef](#)]
27. Roozbeh Nia, A.; Haleh, H.; Saghaei, A. Dual Command Cycle Dynamic Sequencing Method to Consider GHG Efficiency in Unit-Load Multiple-Rack Automated Storage and Retrieval Systems. *Comput. Ind. Eng.* **2017**, *111*, 89–108. [[CrossRef](#)]
28. Chen, F.; Xu, G.; Wei, Y. An Integrated Metaheuristic Routing Method for Multiple-Block Warehouses with Ultranarrow Aisles and Access Restriction. *Complexity* **2019**, *2019*, 1–14. [[CrossRef](#)]
29. Cunkas, M.; Ozer, O. Optimization of Location Assignment for Unit-Load AS/RS with a DualShuttle. *Int. J. Intell. Syst. Appl. Eng.* **2019**, *7*, 66–71. [[CrossRef](#)]
30. Habibi Tostani, H.; Haleh, H.; Hadji Molana, S.M.; Sobhani, F.M. 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. Clean. Prod.* **2020**, *257*, 120409. [[CrossRef](#)]
31. Hojaghania, L.; Nematian, J.; Shojaiea, A.A.; Javadi, M. Metaheuristics for A New MINLP Model with Reduced Response Time for On-Line Order Batching. *Sci. Iran.* **2019**, *28*, 2789–2811. [[CrossRef](#)]
32. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
33. Ashayeri, J.; Gelders, L.F. Warehouse Design Optimization. *Eur. J. Oper. Res.* **1985**, *21*, 285–294. [[CrossRef](#)]
34. Azzi, A.; Battini, D.; Faccio, M.; Persona, A.; Sgarbossa, F. Innovative Travel Time Model for Dual-Shuttle Automated Storage/Retrieval Systems. *Comput. Ind. Eng.* **2011**, *61*, 600–607. [[CrossRef](#)]
35. Randhawa, S.U.; Shroff, R. Simulation-Based Design Evaluation of Unit Load Automated Storage/Retrieval Systems. *Comput. Ind. Eng.* **1995**, *28*, 71–79. [[CrossRef](#)]
36. Lee, H.F. Performance Analysis for Automated Storage and Retrieval Systems. *IIE Trans.* **1997**, *29*, 15–28. [[CrossRef](#)]
37. Hachemi, K.; Sari, Z.; Ghouali, N. A Step-by-Step Dual Cycle Sequencing Method for Unit-Load Automated Storage and Retrieval Systems. *Comput. Ind. Eng.* **2012**, *63*, 980–984. [[CrossRef](#)]
38. Yang, P.; Miao, L.; Xue, Z.; Ye, B. 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 Logist. Transp. Rev.* **2015**, *79*, 164–177. [[CrossRef](#)]
39. Roodbergen, K.J.; Vis, I.F.A. A Survey of Literature on Automated Storage and Retrieval Systems. *Eur. J. Oper. Res.* **2009**, *194*, 343–362. [[CrossRef](#)]
40. Rajasekhar, A.; Lynn, N.; Das, S.; Suganthan, P.N. Computing with the Collective Intelligence of Honey Bees—A Survey. *Swarm. Evol. Comput.* **2017**, *32*, 25–48. [[CrossRef](#)]
41. Hsu, H.-P.; Yang, S.-W. Optimization of Component Sequencing and Feeder Assignment for a Chip Shooter Machine Using Shuffled Frog-Leaping Algorithm. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 56–71. [[CrossRef](#)]
42. Hsu, H.-P. A Fuzzy Knowledge-Based Disassembly Process Planning System Based on Fuzzy Attributed and Timed Predicate/Transition Net. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 1800–1813. [[CrossRef](#)]
43. Hsu, H.-P.; Wang, C.-N.; Chou, C.-C.; Lee, Y.; Wen, Y.-F. Modeling and Solving the Three Seaside Operational Problems Using an Object-Oriented and Timed Predicate/Transition Net. *Appl. Sci.* **2017**, *7*, 218. [[CrossRef](#)]

44. Szczepanski, R.; Tarczewski, T.; Erwinski, K.; Grzesiak, L.M. Comparison of Constraint-handling Techniques Used in Artificial B Auto-Tuning of State Feedback Speed Controller for PMSM. In Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2018), Porto, Portugal, 29–31 July 2018; Volume 1, pp. 269–276.
45. Fountas, N.A.; Benhadj-Djilali, R.; Stergiou, C.I.; Vaxevanidis, N.M. An integrated framework for optimizing sculptured surface CNC tool paths based on direct software object evaluation and viral intelligence. *J. Intell. Manuf.* **2019**, *30*, 1581–1599. [[CrossRef](#)]
46. Rahman, C.M.; Rashid, T.A.; Ahmed, A.M.; Mirjalili, S. Multi-objective learner performance-based behavior algorithm with five multi-objective real-world engineering problems. *Neural Comput. Appl.* **2022**, *34*, 6307–6329. [[CrossRef](#)]
47. Zhang, H.; Wang, G.-G.; Dong, J.; Gandomi, A.H. Improved NSGA-III with Second-Order Difference Random Strategy for Dynamic Multi-Objective Optimization. *Processes* **2021**, *9*, 911. [[CrossRef](#)]