

Article

Energy-Efficient Collaborative Scheduling of Dual-Trolley Quay Cranes and Automated Guided Vehicles in Automated Container Terminals

Shichang Xiao ¹, Shuaishuai Deng ¹, Shaohua Yu ^{2,*} , Peng Zheng ¹ and Zigao Wu ³ 

¹ Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China; scxiao@shmtu.edu.cn (S.X.); 202330210309@stu.shmtu.edu.cn (S.D.); zhengpeng@shmtu.edu.cn (P.Z.)

² School of Intelligence Science and Technology, Nanjing University of Science and Technology, Nanjing 210014, China

³ Department of Mechanical Engineering, North China Electric Power University, Baoding 071003, China; zgwu@ncepu.edu.cn

* Correspondence: shaohua.yu@njust.edu.cn

Abstract

This paper investigates the energy-efficient collaborative scheduling of dual-trolley quay cranes (DTQCs) and automated guided vehicles (AGVs) in automated container terminals (ACTs). Considering operational constraints such as mixed bidirectional flows, limited buffers, precedence constraints, and deadlocks, this complex logistical system is formally characterized as a blocking hybrid flow shop scheduling problem (BHFSSP-BFLB). To systematically minimize the total energy consumption, a mathematical framework grounded in a mixed-integer programming model is developed. To solve the model efficiently, an improved genetic algorithm (IGA) is proposed featuring a two-layer encoding approach to respect precedence and mitigate deadlocks. Furthermore, an active scheduling strategy based on machine idle time insertion is incorporated during decoding to shorten the makespan without increasing energy consumption. Numerical experiments demonstrate that the IGA can significantly decrease the makespan while reducing total energy consumption: compared with a standard genetic algorithm (GA) without active scheduling, the proposed IGA reduces the makespan by 32.35% on average. In addition, the makespan under energy minimization is within 1.5% of that under makespan minimization, indicating that energy optimization yields an almost minimal makespan. Sensitivity analysis further evaluates the effects of DTQC-AGV configurations and buffer capacities, offering practical insights for decision-makers.

Keywords: automated container terminals; collaborative scheduling; energy efficient; improved genetic algorithm; active scheduling strategy



Academic Editor: Zhiguang Feng

Received: 4 January 2026

Revised: 25 January 2026

Accepted: 28 January 2026

Published: 29 January 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Ports are critical nodes in global logistics networks, handling more than 80% of international cargo by volume [1]. The rapid growth of containerized trade has placed increasing pressure on terminals to operate more efficiently, while the International Maritime Organization (IMO) and leading port operators are promoting a transition toward low-carbon and intelligent port systems [2]. As major energy consumers, container terminals face the dual challenge of improving productivity and minimizing environmental impact [3]. Consequently, enhancing energy efficiency through collaborative operations has become a key pathway for the green transformation of ports.

In recent years, automated container terminals (ACTs) have emerged as a cornerstone for achieving both sustainability and high operational efficiency in port logistics. Driven by automation, intelligent planning, and advanced control systems, ACTs can significantly improve productivity while reducing labor costs [2]. However, uncoordinated interactions among the components of integrated automated handling systems often lead to idle waiting, redundant movements, and unnecessary energy consumption [4]. As a result, enhancing efficiency in ACTs transcends mere throughput maximization; it fundamentally necessitates a rigorous approach to waste reduction, minimizing energy losses caused by ineffective scheduling. Motivated by this observation, this study focuses on minimizing total energy consumption (TEC) as its primary optimization objective, while also examining how intelligent scheduling strategies can maintain or even enhance operational efficiency. Accordingly, the main challenge in modern ACTs is therefore no longer the implementation of automation itself, but rather the coordination of heterogeneous equipment.

Figure 1 delineates the layout of an ACT, dividing it into three functional zones: the quayside interface, the horizontal transit area, and the yard storage blocks [5]. Among various ACT subsystems, dual-trolley quay cranes (DTQCs) play a pivotal role in linking seaside and landside operations [6]. Each DTQC integrates a main trolley, a transfer platform, and a portal trolley, and requires precise synchronization with AGVs to ensure a continuous flow of containers. The DTQC-AGV collaboration, however, remains a critical challenge due to the timing mismatches that result in idle waiting and redundant movements, ultimately affecting operational efficiency and energy consumption [7,8]. Optimizing the collaborative scheduling of DTQCs and AGVs to reduce inter-equipment waiting time remains a significant research challenge and a hot topic in the field of ACT operations.

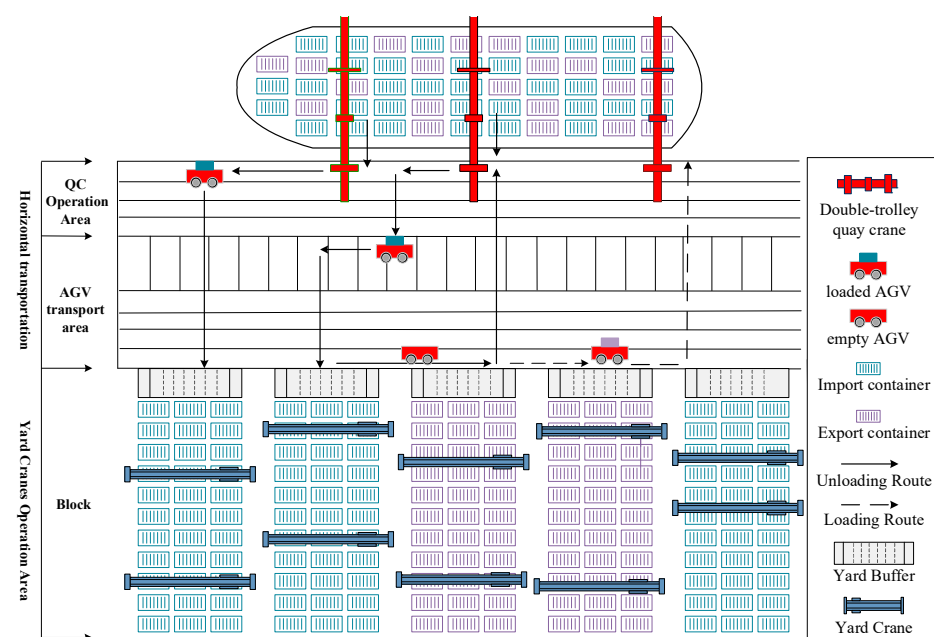


Figure 1. The layout of an automated container terminal.

Recent studies on equipment scheduling in ACTs provide valuable insights into operational optimization. However, the majority of existing studies make simplifying assumptions about several practical constraints present in ACT operations. Specifically, the existing literature often refers to constraints such as dual cycling, buffer limits, or blocking; these factors are often examined in isolation rather than as part of an interacting system [7,9,10]. In contrast, this paper jointly accounts for mixed loading and unloading bidirectional flows, the limited buffers of DTQCs, container precedence constraints, and blocking and

deadlocks observed in real ACTs. Meanwhile, numerous studies have demonstrated that reducing empty travel, idle waiting, and blocking not only significantly decreases energy consumption but also simultaneously enhances operational efficiency; these two objectives are not conflicting but strongly and positively correlated [8,11,12]. On this basis, we formulate a mixed-integer programming model with the objective of minimizing total energy consumption and propose an improved genetic algorithm (IGA) that incorporates an active scheduling strategy based on machine idle time insertion during decoding. This strategy helps shorten the makespan without increasing energy consumption, thereby achieving energy-efficient scheduling.

To address the aforementioned challenges, this paper investigates the collaborative scheduling problem of DTQCs and AGVs in ACT by explicitly considering the operational characteristics, with the aim of optimizing energy consumption while maintaining operational efficiency. The main contributions are summarized as follows:

1. We analyze the energy consumption of equipment in three operational states: loaded, waiting, and empty. The objective is to minimize total energy consumption while considering actual operational constraints, including mixed loading and unloading bidirectional flows, the limited buffers of DTQCs, machine eligibility constraints, container precedence constraints, and issues related to blocking and deadlocks. By adopting a hybrid flow shop scheduling framework, we map the collaborative scheduling problem of DTQCs and AGVs into a blocking hybrid flow shop scheduling problem with bidirectional flows and limited buffers (BHFSSP-BFLB) and develop a corresponding MIP model.
2. We propose an IGA that can effectively solve the BHFSSP-BFLB model. A two-layer encoding method incorporating container precedence constraints is designed to ensure compliance with the prioritized processing sequence of containers in ACTs while effectively mitigating system deadlocks. During decoding, an active scheduling strategy based on the insertion of machine idle time is introduced to minimize the makespan.
3. The effectiveness of the proposed BHFSSP-BFLB model and the IGA in solving this collaborative scheduling problem is validated through numerical experiments. The results demonstrate that the scheduling solutions generated by the active scheduling strategy can significantly reduce total energy consumption while simultaneously decreasing the makespan. Moreover, the numerical experiments reveal the interaction between makespan and energy consumption optimization, providing a useful theoretical basis and decision-making reference for energy-efficient scheduling in ACTs.

2. Literature Review

2.1. Collaborative Scheduling in Automated Container Terminals

In recent decades, operations research in the maritime sector has increasingly focused on optimization in terminal logistics, particularly concerning resource allocation. To keep models computationally tractable, many studies restrict integrated scheduling to only one or two equipment types [13]. In this context, Stahlbock and Voß [14] provided a comprehensive survey of operations research methodologies applicable to this specific domain. Building on these foundations, this section briefly reviews key findings in equipment scheduling at container terminals.

Quay cranes (QCs) serve as the primary interface for vessel cargo operations, handling the loading and discharging of containers, and their efficiency strongly influences overall terminal productivity [15]. Wei et al. [16] addressed the Quay Crane Scheduling Problem (QCSP) via a linear integer model that rigorously enforces time-window adherence. Abou Kasm and Diabat [17] introduced an exact solution methodology that integrates a refined partition heuristic and a branch pricing algorithm for QCSP, considering non-crossing and

safe spacing constraints under single-ship operation to reduce the movement of quay crane repositioning. Chang et al. [18] developed a multi-vessel QCSP framework employing a dynamic rolling horizon approach to address container handling under dynamic ship arrival time. Functioning as the critical logistic link between the quay and the yard, AGVs are essential for fluid cargo transfer, and their efficiency plays an important role in determining the performance of the entire system. Cao et al. [19] proposed a two-layer differential evolution algorithm to optimize the two-way transportation mode in the AGV scheduling problem. Hu et al. [20] devised an IP formulation to specifically resolve conflicts arising from head-on encounters and node occupation. Focused on AGV navigation, Zhong et al. [21] developed an MIP model designed to ensure collision-free trajectories based on path optimization, integrated scheduling and conflict deadlock.

Despite the extensive literature on isolated equipment optimization, the strong operational interdependence among different resources in ACTs indicates that single-equipment optimization without coordination may be insufficient for achieving system-level optimality. Therefore, this paper investigates coordinated scheduling of automated handling units across operational stages to mitigate inter-equipment interference and enhance overall terminal performance.

For the collaborative scheduling problem of multiple pieces of equipment, Yue et al. [6] proposed a two-stage dual-objective mixed integer programming model and an improved non-dominated sorting genetic algorithm for the DTQCs and AGVs configuration and scheduling optimization problem. Liang et al. [22] utilized GA to manage the dynamic scheduling variables of quay cranes and AGVs grouping work surface scheduling formula for the joint scheduling problem. Hop et al. [23] framed the integrated scheduling task as an MIP problem, subsequently applying adaptive particle swarm optimization to solve this model. Duan et al. [24] presented a model that explicitly factors in land-side buffer constraints during integrated scheduling and optimized it using the improved NSGA-II.

To address integrated scheduling in ACTs, some studies model the problem as a hybrid flow shop (HFS) scheduling problem and draw on corresponding solution methods. Qin et al. [25] conceptualized the integrated scheduling workflow as a three-stage HFS problem and proposed a solution method combining mixed integer and constraint programming. Furthermore, Qin and Liang [26] built a flexible flow shop model that strictly incorporates buffer limitations for the integrated scheduling problem; the NEH heuristic algorithm and GA were designed to solve the problem. Jonker et al. [10] constructed a mixed-flow shop scheduling model considering characteristics including simultaneous bidirectional traffic for loading and unloading, job pairing, blocking, and machine-specific constraints, and designed a simulated annealing algorithm for solving it. Similarly, Zhuang et al. [7] aligned integrated scheduling with the Blocked HFS model, accounting for buffer constraints, and designed an adaptive large neighborhood search algorithm to solve it.

While numerous algorithmic approaches exist for multi-equipment collaboration, a gap still exists in effectively integrating the complex characteristics of ACT, such as mixed loading and unloading bidirectional flows, limited buffers of DTQCs, machine eligibility constraints, container precedence constraints, and issues related to blocking and deadlocks.

2.2. The ACT Scheduling Considering Energy Consumption

More recently, the energy consumption of container terminals has emerged as a key area of scholarly focus, particularly with the rapid development of ACTs.

Regarding the scheduling of individual machinery units, Tan et al. [27] established an MIP model for QCSP and quantitatively analyzed the relationship between operation efficiency and energy consumption. Li et al. [28] aimed to simultaneously minimize vessel turnaround duration and the energy burn rate of quay cranes. This resulted in a

bi-objective MIP model, solved using branch-and-bound methodologies. Yang et al. [29] created a model specifically tailored for battery-operated AGVs, factoring in charging needs, considering recycling demand and battery swapping station capacity constraints. Xiao et al. [30] investigated the minimization of the sum of energy cost and delay cost for AGV scheduling in hybrid mode of charging and switching, built an MIP planning structure and solved it via an adaptive large neighborhood search algorithm.

For multi-equipment collaborative scheduling, Zhao et al. [31] constructed an MIP model for energy consumption considering the capacity limitation of the transfer platform and suggested a two-stage tabu search algorithm as the primary solver. Yue et al. [32] devised a two-stage MIP model centered on minimizing aggregate energy usage of DTQCs and AGVs, respectively, and proposed a local enumeration strategy and genetic algorithm. Fan et al. [33] explored scheduling protocols within the context of single-ship operations, taking into account factors such as the capacity limits of transfer platforms and yard buffer racks, as well as AGV endurance time. Fontes and Homayouni [9] examined the coordination between quay cranes and variable-speed transport vehicles under a double-cycle strategy, developed a mixed integer linear programming model and proposed a bi-objective multi-population biased stochastic key genetic algorithm. Luo et al. [34] studied the integrated scheduling problem of the quay cranes and IGVs under loading operation mode, formulated an MIP model and applied a dimensional mutation sparrow search algorithm. Sun et al. [11] constructed a multi-equipment integrated scheduling model with the objective of minimizing the maximum completion time of quay crane and the energy consumption of AGV transportation and solved it by combining simulated annealing and genetic algorithm. Xing et al. [8] focused on the tri-partite scheduling involving QCs, YCs, and AGVs with speed optimization, constructed an MIP model and designed a three-stage method to solve this problem. Under the layout background of U-shaped ACT, Niu et al. [35] developed an MIP model for integrated scheduling that accommodates multi-point transfer operations of quay cranes, IGVs, double-cantilever yard cranes and outer trucks in multi-point loading and unloading mode and designed a decision tree learning algorithm based on a heuristic Monte Carlo tree search algorithm. Considering the uncertain arrival time of the external trucks, Xu et al. [36] investigated the joint scheduling dynamics between DTQCs, AGVs, double-cantilever yard cranes and external trucks. To simultaneously curb carbon emissions and reduce the makespan of loading and unloading equipment, an integrated scheduling model was constructed, and an improved hybrid genetic cuckoo algorithm was proposed.

Despite the attention paid to energy consumption optimization in ACT collaborative scheduling, most existing studies focus solely on the single optimization of energy consumption or efficiency, without fully considering the relationship between the two objectives.

2.3. Research Gap

As shown in Table 1, the research on collaborative scheduling of container terminal equipment has made some progress.

However, there are still the following problems that need to be studied: (1) Most of the studies only consider a single unloading or loading operation, but in actual operation, given that loading and unloading frequently occur in tandem, operational complexity is heightened; (2) For the ACT scheduling process of limited capacity buffers, container loading and unloading precedence constraints, blocking, deadlocks and other terminal operation characteristics are usually considered separately, and current frameworks fail to fully capture the multifaceted constraints of collaborative scheduling of DTQCs and AGVs for ACT; (3) Most existing studies consider only energy consumption or only operational

efficiency when addressing collaborative or few studies explore methods to curtail energy use in multi-equipment ACTs without compromising operational efficiency.

Table 1. Summary of the literature.

Literature	Equipment				Objective		Operational Characteristic			Solution Method
	QC	DTQC	AGV	YC	Energy	Efficiency	MLUBF *	Limited Buffer	Blocking/Deadlock	
Yue et al. [6]		✓	✓			✓		✓		EA + IGA
Zhuang et al. [7]		✓	✓	✓		✓	✓	✓	✓	ALNS
Xing et al. [8]	✓		✓	✓	✓	✓			✓	Three-phase
Fontes et al. [9]	✓		✓		✓		✓			mp-BRKGA
Jonker et al. [10]	✓		✓	✓		✓	✓		✓	TSA
Sun et al. [11]		✓	✓	✓	✓	✓	✓			SA-GA
Qin et al. [25]	✓		✓	✓		✓			✓	MIP + CP
Luo et al. [34]	✓		✓		✓					DMSSA
Bish et al. [37]	✓		✓	✓		✓			✓	TLS
This paper		✓	✓		✓	✓	✓	✓	✓	IGA

* MLUBF means mixed loading and unloading bidirectional; presence of a checkmark indicates included and its absence indicates not included.

To address the above challenges, this paper develops a BHFSSP-BFLB model by considering practical operational constraints, such as mixed bidirectional flows, limited buffers, precedence constraints, and deadlocks. Meanwhile, an improved genetic algorithm, combined with an active scheduling strategy based on idle time insertion, is then proposed. This approach effectively minimizes energy consumption while preserving operational efficiency, thereby achieving high energy-efficient scheduling.

3. Problem Description and Mathematical Modeling

3.1. Problem Description

ACT workflows can be divided into two directional flows: import (unloading) and export (loading) logistics. In import operations, the cycle begins when the DTQC's main trolley hoists a container from the vessel to the transfer platform. The portal trolley moves the container from the platform onto a waiting AGV, which completes the cycle by delivering the container to its assigned block. Conversely, export operations follow the reverse flow, starting with an AGV retrieving a container from the yard, delivering it to the DTQC, transferring it via the portal trolley to the transfer platform, and finally loading it onto the vessel by the main trolley. In ACTs, both loading and unloading operations are executed concurrently, thereby increasing the complexity of equipment collaboration and scheduling.

In the collaborative scheduling problem of DTQCs and AGVs, import containers sequentially pass through four stages: the main trolley, the transfer platform, the portal trolley, and the AGV, while export containers follow the sequence of the AGV, the portal trolley, the transfer platform, and the main trolley. At each stage, only one single piece of equipment is responsible for handling operations. In flow shop scheduling, a job must pass through multiple processing stages sequentially from start to finish. Thus, the joint scheduling of DTQCs and AGVs naturally conforms to the BHFSSP-BFLB problem structure with bidirectional flows and limited buffers. The corresponding mapping between the two is illustrated in Figure 2.

Key characteristics defining this specific collaborative scheduling problem include: (1) Mixed loading and unloading bidirectional flows: The simultaneous processing of inbound and outbound flows. (2) Limited buffers of DTQCs: The transfer platform serves as a finite buffer zone during the operation process, alleviating deadlocks caused by the mixed loading and unloading bidirectional flows to some extent. However, the transfer

platform has a capacity limit, and once the capacity is reached, new containers can no longer be placed. (3) Container precedence constraints: In actual ACT operations, upper-layer containers must be unloaded before lower-layer containers, and lower-layer containers must be loaded before upper-layer containers. (4) Equipment Eligibility: A specific DTQC's trolley and platform must exclusively handle assigned containers. (5) Blocking: The operation processes of ACT equipment are interconnected and tightly dependent. Upstream equipment can only release completed jobs and proceed to the next task when downstream equipment becomes available. (6) Deadlock: Simultaneous bidirectional flows may trigger operational stalemates if the scheduling scheme lacks collaboration. For example, when an AGV delivering an export container arrives under the portal trolley while the portal trolley is ready to unload an import container onto the same AGV, both the portal trolley and AGV are unable to process the current container or receive new containers, resulting in a local deadlock.

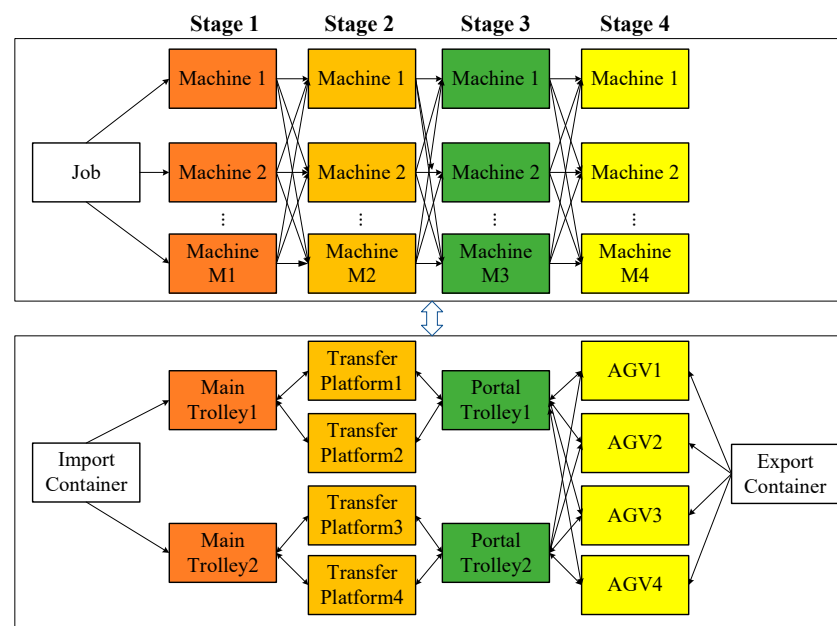


Figure 2. Schematic of HFS applied to the collaborative scheduling problem.

3.2. Mathematical Modeling

3.2.1. Assumptions and Notations

The following assumptions are adopted in this study to render the problem mathematically tractable for modeling:

1. All containers and their handling types are known, regardless of new arrivals.
2. All AGVs can be shared by all DTQCs.
3. There is no mutual interference between DTQCs.
4. The release and extraction time of containers at the AGV or transfer platform are known.
5. The storage locations of the containers on the ship and in the container yard are known.
6. Once a handling operation has started, it must not be interrupted.
7. Each capacity location on the transfer platform is modeled as a single piece of equipment.
8. The yard buffer is set to infinite capacity, and there is no waiting time for AGVs to hand over containers.
9. The operation of the transfer platform does not involve significant energy consumption and is only used as a temporary storage point for containers.

The notations used in the mathematical formulation are shown in Table 2.

Table 2. Definition of parameters and decision variables.

Notations	Description
Parameters:	
$i, k :$	Index of container job
$0 :$	Index of initial virtual job
$\theta :$	Index of final virtual job
$j :$	Index of stage
$m :$	Index of equipment
$U :$	Set of import container job
$L :$	Set of export container job
$N :$	Set of container job, $N = U \cup L$
$N_0 :$	Set of initial virtual job
$N_\theta :$	Set of final virtual job
$H :$	A sufficiently large positive real number
$p_{ij} :$	Processing time of job i at stage j
$M_{ij} :$	Equipment set of job i at stage j
$M_j(m) :$	Set of equipment available at stage j determined by equipment m
$s_{ikj} :$	Adjustment time of switching from job i to job k at stage j
$P :$	Set of container job pairs with priority constraints, when $(i, k) \in P$, job i must precede job k
$C_j^1 :$	Unit-loaded energy consumption of equipment at each stage, $j = 1, 3, 4$, respectively, corresponds to the main trolley, portal trolley and AGV
$C_j^2 :$	Unit empty energy consumption of equipment at each stage, $j = 1, 3, 4$, respectively, corresponds to the main trolley, portal trolley and AGV
$C_j^3 :$	Unit waiting energy consumption of equipment at each stage, $j = 1, 3, 4$, respectively, corresponds to the main trolley, portal trolley and AGV
Decision variable	
$x_{ijm} :$	Binary variable that equals one if job i is processed by equipment m at stage j , and zero otherwise
$y_{ikj} :$	Binary variable that equals one if job i is processed before job k at stage j , and zero otherwise
$z_{ikjm} :$	Binary variable that equals one if job i is processed immediately before job k by equipment m at stage j , and zero otherwise
$u_{ijm} :$	Waiting time before processing job i by equipment m at stage j
$t_{ij} :$	Start time of job i processing at stage j
$r_{ij} :$	Release time after processing of job i at stage j

3.2.2. Mathematical Model

The core objective of our mathematical formulation is to minimize the total energy consumption of the AGV and DTQC fleets. Operational energy expenditure is calculated across three states: loaded, empty, and waiting.

The loaded energy consumption primarily depends on the storage location of the container on the vessel and within the yard block. It is calculated as the product of the equipment's operating time and its unit-loaded energy consumption, as shown in Equation (1):

$$TEC_1 = \sum_{i \in N} \sum_{j \in \{1, 3, 4\}} \sum_{m \in M_{ij}} C_j^1 \times x_{ijm} \times p_{ij} \quad (1)$$

The empty energy consumption is primarily influenced by the terminal scheduling scheme. It is determined by the transfer time of the equipment between containers. Specifically, it is defined as the product of the empty load time and the empty energy consumption rate per unit time. This energy consumption occurs during the interval between completing

the current container and moving to the starting point of the next container, as shown in Equation (2):

$$TEC_2 = \sum_{i \in N} \sum_{k \in N} \sum_{j \in \{1,3,4\}} \sum_{m \in M_{ij}} C_j^2 \times z_{ikjm} \times s_{ikj} \quad (2)$$

The waiting energy consumption occurs when the current-stage equipment has completed processing a container but must wait because the next-stage equipment is still occupied or the previous-stage equipment has not yet finished processing. Both situations can be generalized as follows: before processing the next container, the equipment must complete the current job and then pass through the empty load time and waiting time (which can be zero). The waiting energy consumption is calculated as the product of the waiting time before the equipment processes the next container job and the unit waiting energy consumption value, as shown in Equation (3):

$$TEC_3 = \sum_{i \in N} \sum_{j \in \{1,3,4\}} \sum_{m \in M_{ij}} C_j^3 \times u_{ijm} \quad (3)$$

The BHFSSP-BFLB model is designed to minimize the total energy consumption of DTQCs and AGVs during operation with the following objective function and constraints:

$$\min TEC = TEC_1 + TEC_2 + TEC_3 \quad (4)$$

Subject to

$$t_{ij} + p_{ij} \leq t_{i(j+1)}, \quad \forall i \in U, \forall j \in \{1,2,3\} \quad (5)$$

$$t_{ij} + p_{ij} \leq t_{i(j-1)}, \quad \forall i \in L, \forall j \in \{2,3,4\} \quad (6)$$

$$r_{ij} = t_{i(j+1)}, \quad \forall i \in U, \forall j \in \{1,2,3\} \quad (7)$$

$$r_{ij} = t_{ij} + p_{ij}, \quad \forall i \in U, \forall j \in \{4\} \quad (8)$$

$$r_{ij} = t_{i(j-1)}, \quad \forall i \in L, \forall j \in \{2,3,4\} \quad (9)$$

$$r_{ij} = t_{i(j-1)}, \quad \forall i \in L, \forall j \in \{2,3,4\} \quad (10)$$

The objective function (4) minimizes the total energy consumption. Constraint (5) specifies that for any import container, the start time of the next stage cannot be earlier than the sum of the start time and processing time of the current stage. Constraint (6) indicates that for any export container, the start time of the next stage cannot be earlier than the sum of the start time and processing time of the current stage. Constraints (7) to (10) are used to calculate the release time of the container at any stage. Specifically, Equation (7) defines the release time of the import container at stage $j = 1, 2, 3$ as the start time of the next stage. Equation (8) defines the release time of the import containers at stage $j = 4$ as the completion time of the current stage. Equation (9) defines the release time of the export container at stage $j = 2, 3, 4$ as the start time of the next stage. Equation (10) defines the release time of the export container at stage $j = 1$ as the completion time of the current stage.

$$t_{ij} - (r_{kj} + s_{kij}) + H(2 + y_{ikj} - x_{ijm} - x_{kjm}) \geq 0, \quad \forall i \in N, k \in N, \forall j \in \{1,2,3,4\}, \forall m \in M_{ij} \quad (11)$$

$$t_{kj} - (r_{ij} + s_{ikj}) + H(3 - y_{ikj} - x_{ijm} - x_{kjm}) \geq 0, \quad \forall k \in N, i \in N, \forall j \in \{1,2,3,4\}, \forall m \in M_{ij} \quad (12)$$

Equation (11) represents the blocking constraint by equipment m at any stage when any container job i is later than any container job k . Specifically, when $x_{ijm} = 1, x_{kjm} = 1$ and $y_{ikj} = 0$, constraint (11) becomes $t_{ij} - (r_{kj} + s_{kij}) \geq 0$, which indicates that when job i and

job k are processing by equipment m at stage j , and job i is later than job k , the start time t_{ij} of job i can not be earlier than the sum of the release time r_{kj} of job k at the current stage and the adjustment time s_{kij} of switching from job k to job i at current stage. Equation (12) represents the blocking constraint by equipment m at any stage when any container job i precedes any container job k . Specifically, when $x_{ijm} = 1$, $x_{kjm} = 1$ and $y_{ikj} = 0$, constraint (12) becomes $t_{kj} - (r_{ij} + s_{ikj}) \geq 0$, which indicates that when job i and job k are processing by equipment m at stage j , and job i precedes job k , the start time t_{kj} of job k can not be earlier than the sum of the release time r_{ij} of job i at the current stage and the adjustment time s_{ikj} of switching from job i to job k at current stage.

$$t_{ij} \leq t_{kj}, \quad \forall i, k \in P, \forall j \in \{1, 2, 3, 4\} \quad (13)$$

$$\sum_{m \in M_{ij}} x_{ijm} = 1, \quad \forall i \in N, \forall j \in \{1, 2, 3, 4\} \quad (14)$$

$$\sum_{m' \in M_j(m)} x_{ijm'} = x_{i1m}, \quad \forall i \in N, \forall j \in \{2, 3\}, \forall m \in M_{i1} \quad (15)$$

$$\sum_{m' \in M_j(m)} x_{ijm'} = x_{i1m}, \quad \forall i \in N, \forall j \in \{2, 3\}, \forall m \in M_{i1} \quad (16)$$

$$\sum_{i \in N \cup N_0} z_{kijm} = x_{kjm}, \quad \forall k \in N, \forall j \in \{1, 2, 3, 4\}, \forall m \in M_{kj} \quad (17)$$

$$\sum_{i \in N \cup N_0} z_{kijm} = x_{kjm}, \quad \forall k \in N, \forall j \in \{1, 2, 3, 4\}, \forall m \in M_{kj} \quad (18)$$

$$\sum_{i \in N_0 \cup N} z_{i0jm} = 1, \quad \forall j \in \{1, 2, 3, 4\}, \forall m \in M_{ij} \quad (19)$$

$$\sum_{i \in N \cup N_0} z_{kijm} - \sum_{i \in N_0 \cup N} z_{ikjm} = 0, \quad \forall k \in N, \forall j \in \{1, 2, 3, 4\}, \forall m \in M_{kj} \quad (20)$$

$$z_{iijm} = 0, \quad \forall i \in N, \forall j \in \{1, 2, 3, 4\}, \forall m \in M_{ij} \quad (21)$$

$$t_{kj} - (t_{ij} + p_{ij} + s_{ikj}) + H \left(1 - \sum_{m \in M_{ij}} z_{ikjm} \right) \geq 0, \quad \forall k \in N, i \in N_0 \cup N, \forall j \in \{1, 2, 3, 4\} \quad (22)$$

Constraint (13) is a priority sequence constraint that indicates that the job i processes before the job k . Constraint (14) indicates that any container can only be processed by one piece of the equipment at any stage. Constraint (15) ensures that the main trolley, transfer platform, and portal trolley selected for any container must belong to the same DTQC. Constraint (16) indicates that any container job k has only one immediately preceding task at any stage; Constraint (17) indicates that any container job k has only one immediately following job at any stage. Constraints (18) and (19) ensure that each piece of equipment starts from executing the virtual start job until the completion of the virtual end job. Constraint (20) represents the flow balance constraint, ensuring that the execution of virtual and container jobs by any equipment at any stage must satisfy the flow balance constraint. Constraint (21) guarantees that the container job cannot be self-connected. Constraint (22) indicates that at stage j the start time t_{kj} of the job k must be greater than or equal to the sum of the end time $(t_{ij} + p_{ij})$ of the immediately preceding container job i and the adjustment time s_{ikj} of switching from job i to job k .

$$u_{kjm} \geq t_{kj} - (t_{ij} + p_{ij} + s_{ikj}) - H * (1 - z_{ikjm}), \quad \forall i \in N_0 \cup N, k \in N, \forall j \in \{1, 2, 3, 4\}, \forall m \in M_{ij} \quad (23)$$

$$u_{kjm} \leq t_{kj} - (t_{ij} + p_{ij} + s_{ikj}) + H * (1 - z_{ikjm}), \quad \forall i \in N_0 \cup N, k \in N, \forall j \in \{1, 2, 3, 4\}, \forall m \in M_{ij} \quad (24)$$

$$u_{\theta jm} \geq r_{ij} - (t_{ij} + p_{ij}) - H * (1 - z_{i\theta jm}), \forall i \in N, \forall j \in \{1, 2, 3, 4\}, \forall m \in M_{ij} \quad (25)$$

$$u_{\theta jm} \leq r_{ij} - (t_{ij} + p_{ij}) + H * (1 - z_{i\theta jm}), \forall i \in N, \forall j \in \{1, 2, 3, 4\}, \forall m \in M_{ij} \quad (26)$$

$$x_{ijm}, y_{ikj}, z_{ikjm} \in \{0, 1\}, \forall i, k \in N_0 \cup N \cup N_\theta, \forall j \in \{1, 2, 3, 4\}, \forall m \in M_{ij} \quad (27)$$

Equations (23) and (24) calculate the waiting time of equipment m before processing job k at stage j . Specifically, when $z_{ikjm} = 1$, constraints (23) and (24) become $u_{kjm} = t_{kj} - (t_{ij} + p_{ij} + s_{ikj})$, indicating that when job i is immediately preceded by job k at stage j , the waiting time u_{kjm} before equipment m execute job k is the start time t_{kj} of job k at stage j , minus the end time $(t_{ij} + p_{ij})$ of job i at stage j , and minus the setup time s_{ikj} for switching from job i to job k at stage j . Equations (25) and (26) calculate the waiting time of equipment m for processing the last job i at stage j . That is, when $z_{i\theta jm} = 1$, constraints (25) and (26) become $u_{\theta jm} = r_{ij} - (t_{ij} + p_{ij})$, indicating that when job i is immediately preceded by the virtual end job θ , that is, job i is the last job, its waiting time is the release time r_{ij} of this job at stage j minus the end time $(t_{ij} + p_{ij})$ of this job at this stage. Constraint (27) defines the decision variables x_{ijm} , y_{ikj} and z_{ikjm} as binary variables of 0 or 1.

4. Solution Method

The collaborative scheduling of DTQCs and AGVs has been proven to be NP-hard [37]. As the scale of the dataset increases, relying on standard commercial solvers to find exact solutions within feasible timeframes becomes increasingly impractical. Therefore, an improved genetic algorithm (IGA) based on an active scheduling strategy is proposed, specifically designed to accommodate the BHFSSP-BFLB model.

The proposed IGA is designed to address the characteristics of the BHFSSP-BFLB model. Specifically, it adopts a two-layer encoding scheme based on task allocation, which jointly optimizes the sequence of container operations and AGV assignments while ensuring the prescribed container precedence order. To better align with the actual terminal operations, an active scheduling decoding strategy based on machine idle time is proposed, drawing on concepts from flow shop scheduling methods. This strategy significantly improves scheduling efficiency by reducing equipment idle time.

Figure 3 presents the algorithmic flowchart of the IGA, illustrating its main steps. The subsequent sections describe in detail the chromosome encoding, decoding method, fitness function design, and genetic operations.

4.1. Dual-Layer Chromosome Encoding Based on Task Allocation

Based on the problem characteristics, a two-layer encoding scheme is designed for task assignment. Figure 4 depicts the proposed encoding strategy, where the primary chromosome layer dictates the execution order of containers while ensuring compliance with container precedence constraints. The secondary chromosome layers determine specific AGV assignments. Additionally, the capacity of the transfer platform at stage 2 is determined by its availability.

Suppose an ACT needs to handle eight containers, including four import containers (1–4) and four export containers (5–8). It is equipped with two DTQCs and four AGVs. The transfer platform capacity of each DTQC is two, and the yard is divided into an import container area and an export container area. According to the ship load allocation plan and the container storage plan, import containers 1–2 and export containers 5–6 are handled by quay crane 1, while the remaining containers are handled by quay crane 2. Import containers are stored in block 1, and export containers are stored in block 2. Import

containers 1 and 3 are handled before import containers 2 and 4, while export containers 5 and 7 are handled before export containers 6 and 8. The first layer of chromosomes is randomly generated while ensuring compliance with container precedence constraints and equipment task assignments. Simultaneously, a second layer of chromosomes is generated by randomly assigning an AGV to each container.

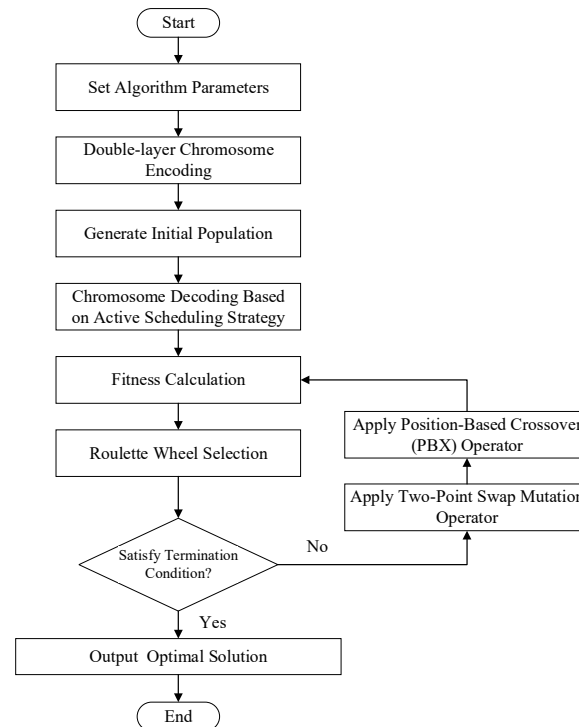


Figure 3. The flow chart of IGA.

Container	1	5	2	6	3	7	4	8
AGV	1	2	1	2	3	3	4	4
Quay crane	1	1	1	1	2	2	2	2

Figure 4. Schematic diagram of chromosome encoding.

4.2. Chromosome Decoding Based on Active Scheduling Strategy

By decoding the chromosomes in Figure 4, in which the operation sequence is derived directly from the vessel's stowage manifest, the operation sequence of AGV 1 is 1-2, and that of DTQC 1 is 1-5-2-6, with similar sequences obtained for other equipment. However, decoding a chromosome is not merely the inverse process of encoding. For example, in the container operation sequence of DTQC 1, the main trolley and portal trolley first unload import container 1 and then export container 5. During this process, the quay crane unloads the import container before loading the export container. The main trolley remains idle while waiting for the portal trolley to process the container, resulting in inefficient resource utilization.

To better reflect real-world operational dynamics, we adopt an active decoding scheme for terminal equipment scheduling to improve overall efficiency. Specifically, we employ an idle-time-based active scheduling strategy for decoding, drawing on decoding methods commonly used in flow shop scheduling [38].

As shown in Figure 5, J_1 and J_2 represent two jobs, while M_1, M_2, M_3 represent three pieces of equipment. In the initial schedule, the job sequence of the jobs strictly follows the order J_1 - J_2 , and J_2 must wait for J_1 to finish processing on equipment M_1 before it can start processing, resulting in a maximum completion time of 9. However, by adopting an active scheduling strategy based on machine idle time, processing can take place during the 4 units of idle time before equipment M_2 processes job J_1 to process job J_2 , reducing the maximum completion time to 7, thereby decreasing the overall job time by 2 units and improving the utilization rate of equipment resources.

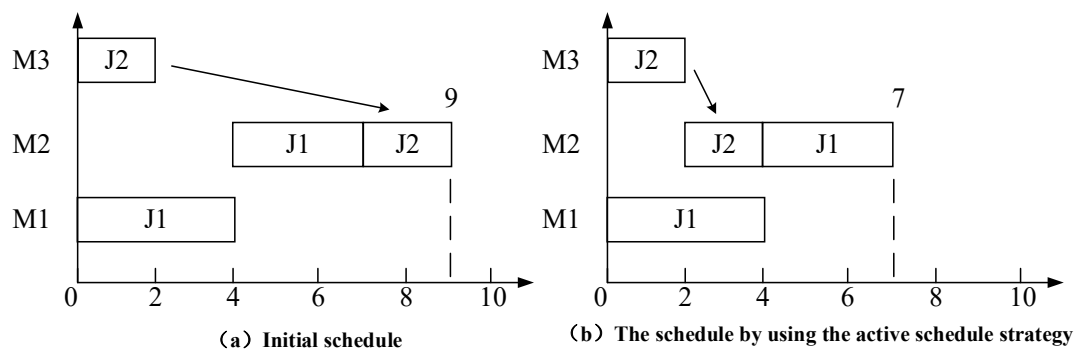


Figure 5. An example of active scheduling strategy.

The above example demonstrates that applying the active scheduling strategy allows effective utilization of machine idle time for pending jobs, thereby reducing the maximum completion time. Consequently, based on the active scheduling strategy, IGA employs an operation insertion method based on machine idle time to decode chromosomes. The decoding procedure unfolds in the following steps:

Step 1: Iterate through the ordered list of pending container tasks, the stages of the container operations, and the allocation of equipment on this chromosome.

Step 2: Translate the first-stage operations into actionable schedules, where the stage indices j for import containers and export containers are 1 and 4, respectively.

(1) Determine whether container k is the first container to be processed by the equipment. If so, the start time of the processing of container k is $t_{kj} = 0$;

(2) Determine whether container k is the last container in the container pair with precedence constraints. If so, its start processing time t_{kj} is $\max\{LM_m + s_{zjk}, t_{ij}\}$, where LM_m indicates the time of the last container completed by the current equipment, s_{zjk} indicates the adjustment time for the equipment to switch from the previous container z to container k at stage j , and t_{ij} indicates the start processing time of the previous container i in the container pair; otherwise, the start processing time of container $t_{kj} = LM_m + s_{zjk}$;

(3) Compute the completion timestamp $c_{kj} = t_{kj} + p_{kj}$ of container k and log any resultant idle intervals for the equipment at the current stage, denoted as $[TS_m, TE_m]$, where TS_m indicates the start time of the interval idle time period and TE_m indicates the end time of the interval idle time period;

(4) If container k is an import container, a random capacity position on the transfer platform will be assigned at the next stage (the transfer platform stage).

Step 3: The decoding operations of the containers at other stages.

(1) Determine whether container k is the first container processed by the equipment at this stage. If so, the start of processing time $t_{kj} = c_{k(j-1)}$ or $c_{k(j+1)}$ for container k indicates the completion time of processing for import container k in the previous stage $j - 1$ or export container k in the previous stage $j + 1$; otherwise, enter (4).

(2) Compute the completion timestamp and log any resultant idle intervals for the equipment $c_{kj} = t_{kj} + p_{kj}$ of container k and record the idle time period $[TS_m, TE_m]$ on the current equipment.

(3) For export containers at the second stage k of the export container (the portal trolley stage), the container will be randomly assigned a capacity position on the transfer platform at next stage (the transfer platform stage).

(4) If container k is not the first container to be processed by the equipment, traverse all the idle time periods $[TS_m, TE_m]$ on the current equipment in turn.

① If $TS_m = 0$, it means that if container k is inserted into the current idle time period, container k is updated to be the first container for this equipment, and the earliest start time $t_{earliest}$ of the container on this equipment is calculated according to Equation (28).

$$t_{earliest} = \max\{c_{k(j-1)} \text{ or } c_{k(j+1)}, TS_m\} \quad (28)$$

② If $TS_m \neq 0$, it means that if container k is inserted into the current idle time period, and container k is located between two containers, the earliest start time $t_{earliest}$ of the container on this equipment is calculated according to Equation (29).

$$t_{earliest} = \max\{c_{k(j-1)} \text{ or } c_{k(j+1)}, TS_m + s_{zkj}\} \quad (29)$$

③ Determine whether container k meets the requirements for insertion into the current idle time period in accordance with Equation (30), where container i indicates the container immediately following the idle time period. If it meets the requirements, insert it into the current idle time period and update the idle time period of the current equipment.

$$t_{earliest} + p_{kj} + s_{kij} \leq TE_m \quad (30)$$

④ If container k is an export container and the current processing stage is stage 3 (the portal trolley stage), before inserting the current idle time period, a check is required to ensure the finish time $t_{earliest} + p_{kj}$ of the existing container on the transfer platform. Otherwise, container k cannot be inserted into the current idle time period.

⑤ If container k cannot be inserted into any of the available idle time periods, schedule container k at the stage's conclusion. The start processing time t_{kj} and processing completion time $c_{kj} = t_{kj} + p_{kj}$ of container k are calculated according to Equation (31), and the interval idle time periods $[TS_m, TE_m]$ on the current equipment are updated.

$$t_{kj} = \max\{c_{k(j-1)} \text{ or } c_{k(j+1)}, LM_m + s_{ikj}\} \quad (31)$$

(5) Update the sequence of container operations on the equipment.

Step 4: Evaluate the final completion time against the baseline schedule. If the new completion time exceeds the original threshold, discard the update; otherwise, accept it and proceed until all containers have been traversed.

The flowchart for chromosome decoding based on active scheduling strategy is presented in Figure 6.

4.3. Population Initialization and Fitness Function

Based on the above two-layer chromosome representation constructed under the task-allocation framework, the initial population is randomly generated by creating T chromosomes that encode both the prioritized container sequence and the AGV assignment sequence. First, we generate a random container sequence that satisfies the precedence constraints, in which import and export containers are ordered according to their priority

rules. Next, we randomly generate the corresponding AGV assignment sequence so that each container is assigned to an available AGV. Each container sequence is paired with its AGV assignment sequence to form a complete chromosome. By repeating this procedure T times, we obtain an initial population of T chromosomes.

Since the research problem involves a minimization objective, an individual's fitness score is calculated as the inverse of its associated objective function cost. Given the objective function value corresponding to the chromosome be TEC_i , and its corresponding fitness value is defined as Equation (32):

$$f_i = \frac{1}{TEC_i} \quad (32)$$

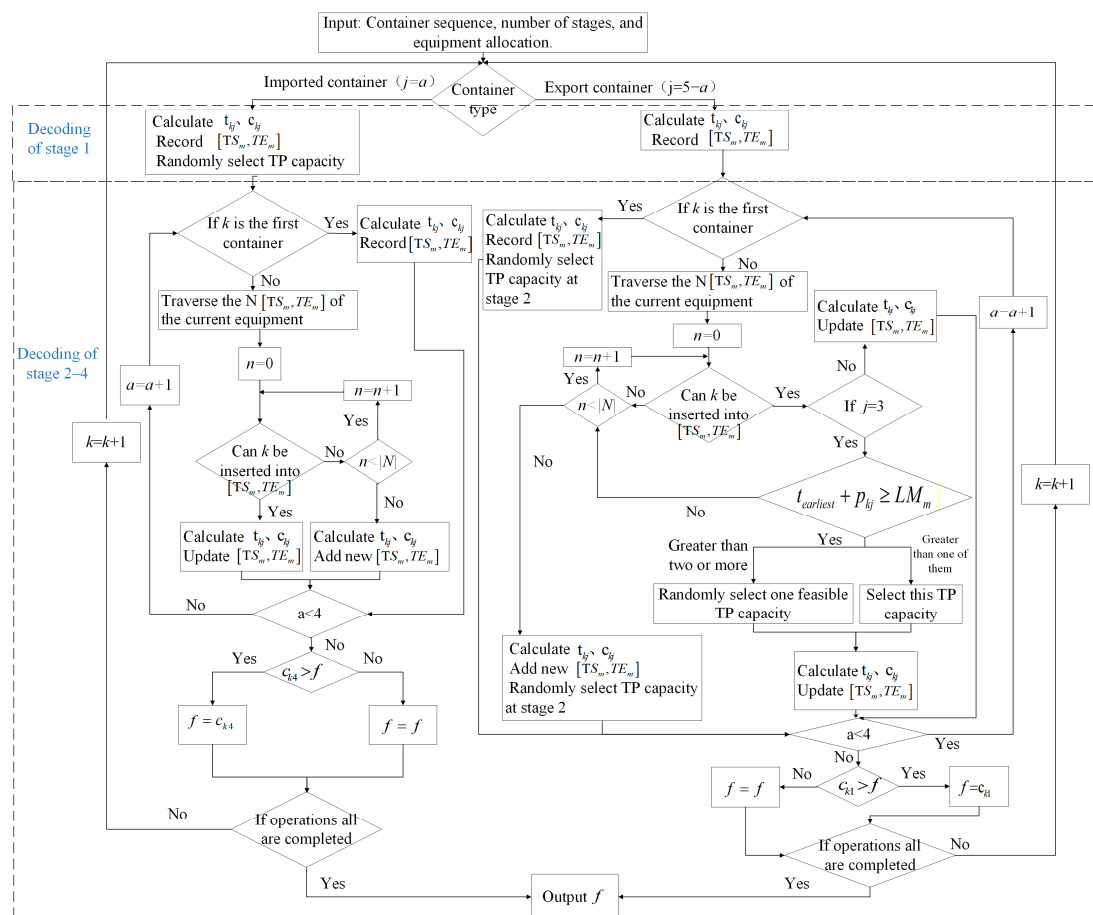


Figure 6. Flowchart of chromosome decoding based on active scheduling strategy.

4.4. Genetic Manipulation

We employ a roulette-wheel selection operator, whereby chromosomes with higher fitness values have a higher probability of being selected. This mechanism is adopted to maintain population diversity and reduce the risk of premature convergence in constrained combinatorial scheduling. In particular, the BHFSSP-BFLB involves multiple interacting constraints, which often leads to a complex and multimodal search landscape; preserving diversity therefore supports continued exploration of alternative feasible schedules.

The crossover operator adopts a position-based crossover (PBX) technique. Two parent chromosomes are selected according to the crossover probability. First, a random set of positions is chosen from the first (container-sequence) gene layer, and the genes at these positions are copied directly to the corresponding positions of the offspring. Next, the remaining genes are filled into the unassigned positions in the order they appear in

the parent, thereby preserving the relative order of the sequence. After crossover, each offspring chromosome is checked against the container precedence constraints. If any violation occurs, a swap-based repair procedure is applied to restore feasibility.

The mutation operator adopts a two-point exchange mutation method. Specifically, a parent chromosome is selected according to the mutation probability, and the values of two randomly chosen gene positions are swapped to generate the offspring chromosome. The mutated offspring generation needs to be checked for compliance with the container precedence constraints; if not, a swap repair operation needs to be applied.

5. Numerical Experiments

To validate the proposed BHFSSP-BFLB model and assess the performance of the IGA, numerical experiments were conducted on both small- and large-scale instances. For a comprehensive comparison, two widely used heuristic algorithms, particle swarm optimization (PSO) and simulated annealing (SA) were also implemented. For small-scale instances, solutions obtained by CPLEX were used as benchmarks to evaluate the solution quality of the three algorithms. All experiments were performed on a computer with Windows 11 operating system, an AMD Ryzen 7 6800H CPU (3.20 GHz) and 16 GB RAM. The mathematical model was solved using IBM ILOG CPLEX 12.9, while IGA, PSO, and SA were implemented in Python 3.12 using NumPy (version 1.19.5) and Pandas (version 1.1.5) for numerical computation and data processing.

5.1. Experimental Setting

The test instances are constructed based on data from an ACT, in which the total length of the quay line is 2350 m, the width of the operating zone in front of the quay is 120 m, and the distance between the blocks is 35 m. The average cycle time for the main trolley is 120 s per TEU, while the portal trolley requires 60 s per TEU. The speed of the AGV with a heavy load is 210 m per minute, and the speed of the empty load is 350 m per minute. The relevant energy consumption parameters of the DTQCs and AGV selected based on the research of Fan et al. [33] are presented in Table 3. The settings of the relevant parameters have been further calibrated from the design values of the relevant equipment at this ACT.

Table 3. Relevant energy consumption parameters.

Parameter	Description	Value (kW·h/(h·unit))
C_1^1	loaded energy consumption of main trolley	91.24
C_3^1	loaded energy consumption of portal trolley	91.24
C_4^1	loaded energy consumption of AGV	21
C_1^2	empty energy consumption of main trolley	70.18
C_3^2	empty energy consumption of portal trolley	70.18
C_4^2	empty energy consumption of AGV	14
C_1^3	waiting energy consumption of main trolley	49.6
C_3^3	waiting energy consumption of portal trolley	49.6
C_4^3	waiting energy consumption of AGV	9

To balance computational time and solution performance, the IGA parameters are determined through a combination of literature reference [26] and problem-specific orthogonal experiments. Given that parameter settings significantly affect IGA performance and optimal combinations vary by problem type, especially for the complex BHFSSP-BFLB with multiple constraints. We designed an orthogonal experiment with crossover probability P_c at 0.6, 0.8, 0.9 and mutation probability P_m at 0.1, 0.2, 0.25, evaluating performance via convergence speed and computational efficiency as shown in Table 4, where OBJ indicates

the objective function value and T is the average solving time. Results show $P_c = 0.9$ and $P_m = 0.2$ yield the best comprehensive performance. Thus, the final IGA parameters are set as population size = 100, number of iterations = 500, crossover probability = 0.9, and mutation probability = 0.2.

Table 4. Algorithm parameters of IGA.

Algorithm Parameters	$P_m = 0.1$		$P_m = 0.2$		$P_m = 0.25$	
	OBJ	T (s)	OBJ	T (s)	OBJ	T (s)
$P_c = 0.6$	49.66	47.65	49.82	46.02	48.79	46.64
$P_c = 0.8$	48.72	48.05	47.65	45.79	48.45	47.28
$P_c = 0.9$	46.53	48.32	45.98	43.56	46.28	45.12

5.2. Analysis of Experimental Results

To comprehensively evaluate the performance of the proposed BHFSSP-BFLB model and the IGA, numerical experiments were conducted on two sets of problem instances with different scales. Specifically, small-scale instances (S1–S10) were used to evaluate solution quality by comparing IGA with the exact solutions obtained by CPLEX. Large-scale instances (L1–L10) were used to assess scalability and robustness by benchmarking IGA against two widely used heuristic algorithms, PSO and SA, under increasing problem sizes.

5.2.1. Results of Small- and Large-Scale Numerical Experiments

For small-scale instances, CPLEX was used to obtain optimal solutions when possible or the best feasible solutions within the 7200 s limit, and the corresponding objective values were treated as benchmarks for algorithm comparison. Therefore, both IGA and CPLEX were tasked with solving the BHFSSP-BFLB instances. The solution results are shown in Table 5, where T_{real} is the average solution time of the CPLEX, T_{IGA} is the average solution time of the IGA, OBJ indicates the objective function value, and GAP represents the percentage of the difference.

Table 5. Calculation results of CPLEX and IGA.

ID	Containers	DTQCs/AGVs/Blocks	CPLEX		IGA			GAP1 *
			T_{real} (s)	OBJ ₁	T_{IGA} (s)	OBJ ₂	AVG	
S1	8	2/4/2	19.70	45.44	43.05	45.97	46.36	1.17%
S2	8	2/6/2	18.34	45.44	42.99	46.03	46.38	1.30%
S3	12	2/4/2	7200	69.22	65.62	70.80	71.57	2.28%
S4	12	2/6/2	7200	69.82	65.45	70.95	71.87	1.62%
S5	16	2/4/2	7200	92.99	90.53	95.54	98.01	2.74%
S6	16	2/6/2	7200	93.53	90.38	96.23	97.92	2.89%
S7	20	2/4/2	--	--	111.38	120.75	124.78	--
S8	20	2/6/2	--	--	120.74	121.28	124.11	--
S9	24	2/4/2	--	--	148.11	146.19	152.10	--
S10	24	2/6/2	--	--	147.64	147.71	152.47	--
AVG								2.00%

* $\text{GAP1} = (\text{OBJ}_2 - \text{OBJ}_1) / \text{OBJ}_1 \times 100\%$.

As shown in Table 5, IGA achieves solutions close to the CPLEX benchmarks, with an average gap of 2.00% for instances where a CPLEX reference value is available. This indicates that IGA can solve the proposed model effectively and deliver competitive solution quality. When the instance size increases to 12 containers, CPLEX becomes less time-efficient, whereas IGA attains solutions within 3% of the CPLEX benchmark in substantially shorter time. For larger cases, CPLEX fails to find a feasible solution within

7200 s, while IGA consistently produces high-quality feasible schedules within acceptable runtimes, demonstrating its robustness as the problem scale grows.

Figure 7 presents the Gantt chart, for instance S9, and the scheduling sequence of each piece of equipment. The horizontal axis denotes the processing timeline, while the vertical axis lists the equipment identifiers. Process x-y denotes the yth operation of task x. Specifically, Machines 1 and 2 correspond to the main trolleys of the quay cranes, Machines 3 to 6 to the transfer platforms, Machines 7 and 8 to the portal trolleys, and Machines 9 to 12 to the AGVs. Tasks 1 to 10 represent import containers, whereas Tasks 11 to 20 represent export containers.

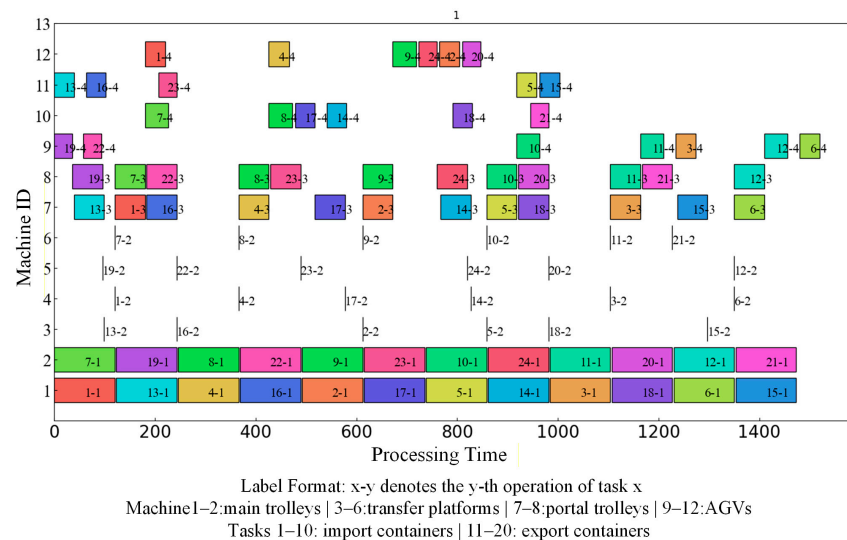


Figure 7. The Gantt chart of instance S9.

For large-scale instances (L1–L10), the problem size increases significantly. Due to the inherent computational complexity of the problem [37], the computing time of CPLEX escalates exponentially, failing to obtain feasible solutions within the specified time limit of 7200 s. Consequently, only the three heuristic algorithms, namely the IGA, PSO, and SA, were utilized for these instances. To evaluate and compare the performance and stability of the three algorithms across different scales, each problem instance was executed 20 times independently, and key performance metrics were recorded, including the average solution time (T), optimal objective function value (OBJ) and average objective function value (AVG). The experimental results are summarized in Table 6.

Table 6. Results of computational experiments in large-size instances.

ID	Containers	DTQCs/AGVs/ Blocks	PSO		SA		IGA		GAP	
			T _{PSO} (s)	OBJ _{PSO}	T _{SA} (s)	OBJ _{SA}	T _{IGA} (s)	OBJ _{IGA}	GAP2 ^{*1}	GAP3 ^{*2}
L1	40	4/12/4	150.89	248.75	231.86	260.47	242.06	245.86	1.2%	5.9%
L2	40	4/14/4	147.64	246.78	226.38	259.89	242.92	244.59	0.9%	6.3%
L3	60	5/14/4	246.31	409.85	345.26	411.36	396.75	380.7	7.7%	8.1%
L4	60	5/16/4	240.18	415.62	338.27	410.28	392.57	381.11	9.1%	7.7%
L5	80	6/16/4	559.24	529.65	498.65	569.45	543.08	526.43	0.6%	8.2%
L6	80	6/18/4	540.34	530.98	504.37	565.74	536.3	527.1	0.7%	7.3%
L7	100	7/18/4	594.36	697.34	654.39	716.91	660.53	680.81	2.4%	5.3%
L8	100	7/20/4	599.48	696.41	672.87	712.39	691.56	677.27	2.8%	5.2%
L9	120	8/20/4	874.26	872.66	890.71	905.26	840.54	850.86	2.6%	6.4%
L10	120	8/22/4	865.34	874.28	864.49	907.48	850.69	851.46	2.7%	6.6%
AVG			481.804		522.7		539.7		3.1%	6.7%

^{*1} GAP2 = (OBJ_{PSO} − OBJ_{IGA})/OBJ_{IGA} × 100%. ^{*2} GAP3 = (OBJ_{SA} − OBJ_{IGA})/OBJ_{IGA} × 100%.

The experimental results of large-scale instances in Table 6 further validate the superiority and stability of the proposed IGA in solving proposed scheduling problem. As the problem scale expands, the solution time of all three algorithms increases, and IGA takes slightly longer runtime but achieves better optimization performance, with average GAP2 and GAP3 of 3.1% and 6.7%, respectively, compared with PSO and SA. In addition, IGA maintains stable performance in different configurations, demonstrating strong adaptability to large-scale and complex scheduling scenarios.

5.2.2. Effectiveness of the Active Scheduling Strategy

The active scheduling method based on the idle time can decode chromosomes into active scheduling schemes, thereby compressing the critical path and efficiently diminishing the makespan. To verify the effectiveness of this method, we solve the BHFSSP-BFLB model under a makespan-minimization objective using two algorithms: a standard GA without the active scheduling decoding and the proposed IGA with this decoding mechanism. For several representative small-scale instances, we record the resulting makespan values (denoted as C_{max1} for GA and C_{max2} for IGA), as summarized in Table 7.

Table 7. The results of GA and IGA.

ID	GA	IGA	GAP4 *
	C_{max1}	C_{max2}	
S1	729	489	32.92%
S3	1095	735	32.88%
S5	1467	981	33.13%
S7	1876	1278	31.88%
S9	2207	1524	30.95%
AVG	1474.8	1001.4	32.35%

$$* \text{GAP4} = (C_{max1} - C_{max2}) / C_{max2} \times 100\%.$$

Table 7 confirms that schedules without the active strategy consistently result in longer completion times compared to those utilizing the active scheduling method, with an average GAP of 32.35%. This clearly indicates that the active scheduling decoding effectively improves schedule compactness in the tested instances and improves overall ACT efficiency.

An examination of the resulting schedules provides additional insight into this improvement. When the active scheduling strategy is not applied, idle time between operations is not utilized for inserting pending jobs, which results in larger idle gaps and additional waiting for both the main and portal trolleys. By contrast, with the active scheduling strategy, such idle gaps can be used to accommodate pending container-handling tasks. This better utilization of idle time reduces unnecessary waiting and contributes to a shorter overall completion time.

A closer inspection of the resulting schedules further explains this improvement. Without the active scheduling strategy, equipment idle time is not effectively exploited, leading to long idle waiting periods for both the main trolleys and the portal trolleys. In contrast, when the active scheduling strategy is applied, these idle periods are systematically used to process pending container jobs. By making full use of available idle time, the active scheduling strategy produces smoother equipment utilization and a substantial reduction in the overall completion time.

5.2.3. Analysis of the Correlation Between Energy Consumption and C_{max}

Following the experimental setup of Zhang et al. [12], we examine the relationship between total energy consumption and makespan. The linear association is quantified using

the Pearson correlation coefficient, which is defined as the covariance of the two variables divided by the product of their standard deviations. The corresponding formulation is given in Equation (33):

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (33)$$

We compute the Pearson correlation coefficient to quantify the linear relationship between makespan and total energy consumption across the solutions obtained for several instances. The averaged results are reported in Table 8.

Table 8. Pearson correlation coefficient of some instances.

ID	Mean	Maximum	Minimum
S1	0.91	0.94	0.88
S3	0.92	0.94	0.87
S5	0.92	0.94	0.89
S7	0.91	0.94	0.88
S9	0.92	0.93	0.90
AVG	0.92	0.94	0.89

As reported in Table 8, the Pearson correlation coefficient is above 0.88 for all tested instances, with an average value of 0.92, indicating a strong positive association between total energy consumption and makespan. This observation suggests that, under the studied settings, improvements in energy performance tend to coincide with reductions in makespan, and the two objectives are largely aligned. Consequently, optimizing total energy consumption in the proposed BHFSSP-BFLB framework can also lead to competitive makespan performance, supporting energy-efficient scheduling without a substantial sacrifice in operational efficiency.

Further analysis of the results from different scale problem instances compares the maximum completion times of the two models: (1) the maximum completion time (set to C_{max3}) of the DTQCs and AGVs collaborative scheduling model with the goal of energy consumption optimization, where the total energy consumption has the optimal value. (2) The maximum completion time C_{max} obtained with the objective of minimizing makespan (set to C_{max4}). For instances where exact CPLEX solutions are attainable, the maximum completion time solved by CPLEX is used. For the instances that CPLEX cannot solve, the maximum completion time solved by IGA is used. The results are shown in Table 9.

From Table 9, the GAP between makespan values derived from the perspective of minimizing energy consumption and those obtained under the objective of minimizing makespan is only 1.5%. This indicates that when IGA with the active scheduling strategy is used to solve this problem, the makespan for the overall operations under minimal energy consumption remains at a relatively low level. In other words, the proposed approach can reduce total energy consumption while maintaining operation efficiency, enabling container tasks to be completed within a short time horizon and supporting energy-efficient scheduling.

Table 9. Comparison of makespan obtained by using two comparison objective functions.

ID	C_{max3}	C_{max4}	GAP5 *
S1	489	489	0.00%
S2	489	489	0.00%
S3	735	735	0.00%
S4	735	735	0.00%
S5	981	981	0.00%

Table 9. Cont.

ID	C_{max3}	C_{max4}	GAP5 *
S6	1032	981	5.20%
S7	1278	1227	4.16%
S8	1227	1227	0.00%
S9	1524	1473	3.46%
S10	1473	1473	0.00%
L1	1227	1227	0.00%
L2	1227	1227	0.00%
L3	1694	1667	1.62%
L4	1667	1667	0.00%
L5	1913	1913	0.00%
L6	1918	1913	0.26%
L7	2343	2172	7.87%
L8	2176	2151	1.16%
L9	2497	2365	5.59%
L10	2378	2362	0.68%
AVG			1.50%

* GAP5 = $(C_{max3} - C_{max4}) / C_{max4} \times 100\%$.

5.2.4. Analysis of the Configuration Ratio Between DTQCs and AGVs

To systematically evaluate how DTQC and AGV configurations affect energy consumption, we fixed the number of container tasks at 100 and DTQCs at 7. We then analyzed how varying AGV numbers influence: total energy consumption, DTQC waiting energy, and AGV waiting energy. The corresponding changes in energy consumption are shown in Figure 8.

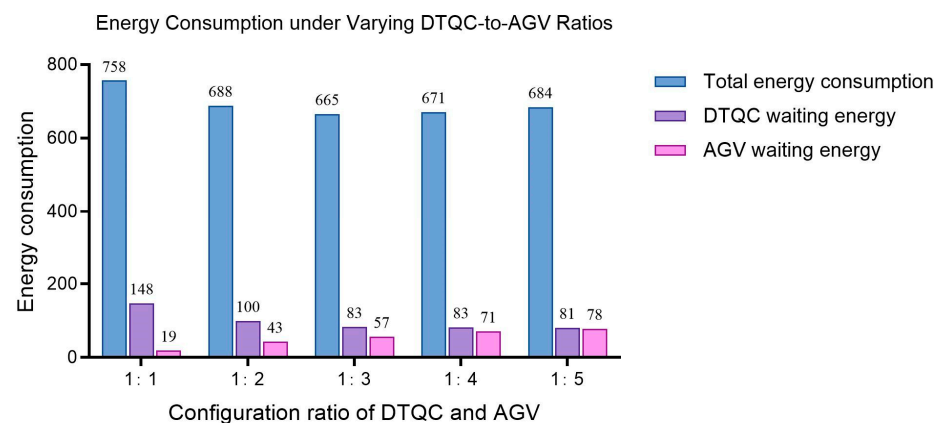


Figure 8. Energy consumption under varying DTQC-to-AGV configuration ratios.

Figure 8 suggests that whenever the AGV fleet size remains below a 3:1 ratio with DTQCs, both total energy consumption and DTQC waiting energy consumption decrease as the number of AGVs increases. However, the waiting energy consumption of the AGVs increases. When the number of AGVs exceeds this threshold, the total energy consumption of the system increases. This phenomenon is likely due to higher idle energy costs stemming from excess of AGVs, implying that the optimal DTQC-to-AGV ratio is approximately 1:3.

5.2.5. Analysis Transfer Platform Capacity

After determining the configuration ratio of DTQCs and AGVs, this section further analyzes the impact of the intermediate platform capacity on total energy consumption, DTQC and AGV waiting energy consumption. The corresponding changes in energy consumption are shown in Figure 9.

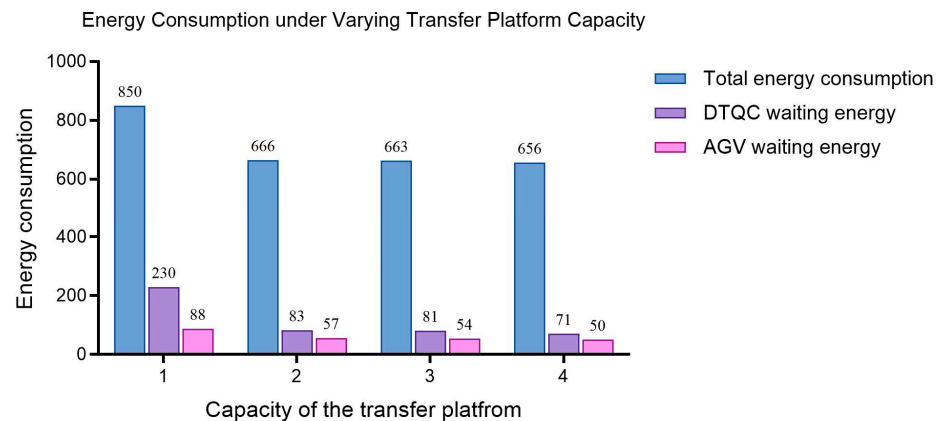


Figure 9. Energy consumption under varying transfer platform capacity.

According to the data in Figure 9, as the capacity of the transfer platform increases, the total energy consumption and the waiting energy consumption of the DTQCs and AGVs generally decrease. This indicates that increasing the capacity of the transfer platform can effectively reduce the waiting energy consumption between the equipment, thereby lowering the overall energy consumption. However, in the operation of the ACTs, since infinite expansion is impossible, considering the capacity limits is crucial in the transfer platform in the collaborative scheduling problem of DTQCs and AGVs in ACTs. This helps ensure the optimization of system energy consumption under limited resource conditions and facilitates energy-efficient scheduling.

6. Conclusions

This work tackles the challenge of synchronizing DTQCs and AGVs in ACTs with a focus on energy efficiency. The characteristics of mixed loading and unloading bidirectional flows, limited buffers of DTQCs, machine eligibility constraints, container precedence constraints, blocking, and deadlocks in ACTs are comprehensively considered. The problem is formulated as a blocking hybrid flow shop scheduling problem with bidirectional flows and limited buffers (BHFSSP-BFLB), and an MIP model is established with the objective of minimizing total energy consumption. In addition, an IGA featuring a two-layer encoding approach and an active scheduling decoding method is proposed.

Numerical experiments demonstrate that the proposed model can accurately depict actual operational characteristics and significantly reduce the total energy consumption, while the IGA exhibits excellent optimization performance across different problem scales: in small-scale instances, its solutions have an average gap of only 2.00% with CPLEX's exact solutions, and in large-scale instances, it outperforms PSO and SA with average gaps of 3.1% and 6.7%, respectively. Additionally, the embedded active scheduling strategy effectively shortens the makespan. As compared with the standard GA without this strategy, the proposed IGA reduces the makespan by an average of 32.35%, which verifies the strategy's role in enhancing operational efficiency. Moreover, the Pearson correlation coefficient between total energy consumption and makespan is as high as 0.92, confirming that optimizing energy consumption can simultaneously promote the reduction in makespan. Notably, the GAP between makespan values derived from energy-minimization and makespan-minimization objectives is only 1.5%, indicating that when the energy consumption objective is optimized, the resulting makespan is already sufficiently low. The study provides managerial insights showing that an approximately 1:3 DTQC-to-AGV configuration and appropriately sized transfer platform buffers can significantly reduce total and waiting energy consumption, thereby improving resource utilization and supporting greener, more sustainable ACT operations.

Future research on the proposed collaborative scheduling problem of DTQCs and AGVs will focus on the following areas: incorporating yard side operational coupling and resource constraints to characterize the operational coupling between the quay and the yard and the associated congestion effects; considering processing-time variability and other disturbance factors to improve the adaptability and robustness of scheduling decisions; and modeling inter-crane interference (e.g., safety clearance and non-crossing constraints) together with more flexible bay-assignment and coordination strategies to better reflect practical terminal operations.

Author Contributions: Conceptualization, S.X.; methodology, S.X. and P.Z.; software, S.D. and S.Y.; validation, S.D. and Z.W.; formal analysis, S.X. and S.Y.; investigation, P.Z.; writing—original draft preparation, S.X. and S.D.; writing—review and editing, S.Y. and Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: The research is supported by National Natural Science Foundation of China [grant no. 72301137].

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Statista Research Department. Container Shipping—Statistics and Facts. Available online: <https://www.statista.com/topics/1367/container-shipping> (accessed on 10 January 2024).
2. Hsu, H.P.; Wang, C.N.; Nguyen, T.T.T.; Dang, T.T.; Pan, Y.J. Hybridizing WOA with PSO for Coordinating Material Handling Equipment in an Automated Container Terminal Considering Energy Consumption. *Adv. Eng. Inform.* **2024**, *60*, 102410. [\[CrossRef\]](#)
3. Lombardi, S.; Dillio, G.; Tribioli, L.; Jannelli, E. Optimal Design of an Adaptive Energy Management Strategy for a Fuel Cell Tractor Operating in Ports. *Appl. Energy* **2023**, *352*, 121917. [\[CrossRef\]](#)
4. Tang, D.; Tang, H.; Yuan, C.; Dong, M.; Diaz-Londono, C.; Agundis-Tinajero, G.D. Economic and Resilience-Oriented Operation of Coupled Hydrogen–Electricity Energy Systems at Ports. *Appl. Energy* **2025**, *390*, 125825. [\[CrossRef\]](#)
5. Zhang, X.; Zeng, Q.; Sheu, J.B. Modeling the Productivity and Stability of a Terminal Operation System with Quay Crane Double Cycling. *Transp. Res. Part E Logist. Transp. Rev.* **2019**, *122*, 181–197. [\[CrossRef\]](#)
6. Yue, L.; Fan, H.; Ma, M. Optimizing Configuration and Scheduling of Double 40 ft Dual-Trolley Quay Cranes and AGVs for Improving Container Terminal Services. *J. Clean. Prod.* **2021**, *292*, 126019. [\[CrossRef\]](#)
7. Zhuang, Z.; Zhang, Z.; Teng, H.; Qin, W.; Fang, H. Optimization for Integrated Scheduling of Intelligent Handling Equipment with Bidirectional Flows and Limited Buffers at Automated Container Terminals. *Comput. Oper. Res.* **2022**, *145*, 105863. [\[CrossRef\]](#)
8. Xing, Z.; Liu, H.; Wang, T.; Chew, E.P.; Lee, L.H.; Tan, K.C. Integrated Automated Guided Vehicle Dispatching and Equipment Scheduling with Speed Optimization. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *169*, 102993. [\[CrossRef\]](#)
9. Fontes, D.B.M.M.; Homayouni, S.M. A Bi-Objective Multi-Population Biased Random Key Genetic Algorithm for Joint Scheduling Quay Cranes and Speed Adjustable Vehicles in Container Terminals. *Flex. Serv. Manuf. J.* **2023**, *35*, 241–268. [\[CrossRef\]](#)
10. Jonker, T.; Duinkerken, M.B.; Yorke-Smith, N.; de Waal, A.; Negenborn, R.R. Coordinated Optimization of Equipment Operations in a Container Terminal. *Flex. Serv. Manuf. J.* **2021**, *33*, 281–311. [\[CrossRef\]](#)
11. Sun, B.; Zhai, G.; Li, S.; Pei, B. Multi-Resource Collaborative Scheduling Problem of Automated Terminal Considering the AGV Charging Effect Under COVID-19. *Ocean Coast. Manag.* **2023**, *232*, 106422. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Zhang, Y.; Tang, K.; Xu, Y.; Xu, S. Integrated Scheduling of Handling Operations in Container Terminal Considering Energy Saving. *Comput. Integr. Manuf. Syst.* **2024**, *30*, 2608–2620. [\[CrossRef\]](#)
13. Vis, I.F.A.; de Koster, R. Transshipment of Containers at a Container Terminal: An Overview. *Eur. J. Oper. Res.* **2003**, *147*, 1–16. [\[CrossRef\]](#)
14. Stahlbock, R.; Voß, S. Operations Research at Container Terminals: A Literature Update. *OR Spectr.* **2008**, *30*, 1–52. [\[CrossRef\]](#)
15. Msakni, M.K.; Diabat, A.; Rabadi, G.; Al-Salem, M.; Kotachi, M. Exact Methods for the Quay Crane Scheduling Problem When Tasks Are Modeled at the Single Container Level. *Comput. Oper. Res.* **2018**, *99*, 218–233. [\[CrossRef\]](#)
16. Wei, M.; He, J.; Tan, C.; Yue, J.; Yu, H. Quay Crane Scheduling with Time Windows Constraints for Automated Container Port. *Ocean Coast. Manag.* **2023**, *231*, 106401. [\[CrossRef\]](#)

17. Abou Kasm, O.; Diabat, A. The Quay Crane Scheduling Problem with Non-Crossing and Safety Clearance Constraints: An Exact Solution Approach. *Comput. Oper. Res.* **2019**, *107*, 189–199. [[CrossRef](#)]
18. Chang, D.; Fang, T.; Fan, Y. Dynamic Rolling Strategy for Multi-Vessel Quay Crane Scheduling. *Adv. Eng. Inform.* **2017**, *34*, 60–69. [[CrossRef](#)]
19. Cao, Y.; Yang, A.; Liu, Y.; Zeng, Q.; Chen, Q. AGV Dispatching and Bidirectional Conflict-Free Routing Problem in Automated Container Terminal. *Comput. Ind. Eng.* **2023**, *184*, 109611. [[CrossRef](#)]
20. Hu, H.; Yang, X.; Xiao, S.; Wang, F. Anti-Conflict AGV Path Planning in Automated Container Terminals Based on Multi-Agent Reinforcement Learning. *Int. J. Prod. Res.* **2023**, *61*, 65–80. [[CrossRef](#)]
21. Zhong, M.; Yang, Y.; Dessouky, Y.; Postolache, O. Multi-AGV Scheduling for Conflict-Free Path Planning in Automated Container Terminals. *Comput. Ind. Eng.* **2020**, *142*, 106371. [[CrossRef](#)]
22. Liang, C.J.; Shen, Z.; Zhang, Y. Quay Crane and AGV Joint Scheduling with Grouping Work Surface under Interference Constraints. *Comput. Eng. Appl.* **2020**, *56*, 254–261. [[CrossRef](#)]
23. Hop, D.C.; Van Hop, N.; Anh, T.T.M. Adaptive Particle Swarm Optimization for Integrated Quay Crane and Yard Truck Scheduling Problem. *Comput. Ind. Eng.* **2021**, *153*, 107075. [[CrossRef](#)]
24. Duan, J.; Li, L.; Zhang, Q.; Qin, J.; Zhou, Y. Integrated Scheduling of Automatic Guided Vehicles and Automatic Stacking Cranes in Automated Container Terminals Considering Landside Buffer Zone. *Transp. Res. Rec.* **2023**, *2677*, 502–528. [[CrossRef](#)]
25. Qin, T.; Du, Y.; Chen, J.; Sha, M. Combining Mixed Integer Programming and Constraint Programming to Solve the Integrated Scheduling Problem of Container Handling Operations of a Single Vessel. *Eur. J. Oper. Res.* **2020**, *285*, 884–901. [[CrossRef](#)]
26. Qin, Q.; Liang, C. Optimization of Equipment Coordination Scheduling Considering Buffer Space in Automated Container Terminal. *Comput. Eng. Appl.* **2020**, *56*, 262–270. [[CrossRef](#)]
27. Tan, C.; Yan, W.; Yue, J. Quay Crane Scheduling in Automated Container Terminal for the Trade-Off between Operation Efficiency and Energy Consumption. *Adv. Eng. Inform.* **2021**, *48*, 101285. [[CrossRef](#)]
28. Li, H.; Li, X. A Branch-and-Bound Algorithm for the Bi-Objective Quay Crane Scheduling Problem Based on Efficiency and Energy. *Mathematics* **2022**, *10*, 4705. [[CrossRef](#)]
29. Yang, X.; Hu, H.; Jin, J. Battery-Powered Automated Guided Vehicles Scheduling Problem in Automated Container Terminals for Minimizing Energy Consumption. *Ocean Coast. Manag.* **2023**, *246*, 106873. [[CrossRef](#)]
30. Xiao, S.; Huang, J.; Hu, H.; Gu, Y. Automatic Guided Vehicle Scheduling in Automated Container Terminals Based on a Hybrid Mode of Battery Swapping and Charging. *J. Mar. Sci. Eng.* **2024**, *12*, 305. [[CrossRef](#)]
31. Zhao, Q.; Ji, S.; Guo, D.; Du, X.; Wang, H. Research on Cooperative Scheduling of Automated Quayside Cranes and Automatic Guided Vehicles in Automated Container Terminal. *Math. Probl. Eng.* **2019**, *2019*, 6574582. [[CrossRef](#)]
32. Yue, L.; Fan, H.; Zhai, C. Joint Configuration and Scheduling Optimization of a Dual-Trolley Quay Crane and Automatic Guided Vehicles with Consideration of Vessel Stability. *Sustainability* **2019**, *12*, 24. [[CrossRef](#)]
33. Fan, H.; Guo, Z.; Yue, L.; Ma, M. Joint Configuration and Scheduling Optimization of Dual-Trolley Quay Crane and AGV for Container Terminal Considering Energy Saving. *Acta Autom. Sin.* **2021**, *47*, 2412–2426. [[CrossRef](#)]
34. Luo, Y.; Liang, X.; Zhang, Y.; Tang, K.; Li, W. Energy-Aware Integrated Scheduling for Quay Crane and IGV in Automated Container Terminal. *J. Mar. Sci. Eng.* **2024**, *12*, 376. [[CrossRef](#)]
35. Niu, Y.; Yu, F.; Yao, H.; Yang, Y. Multi-Equipment Coordinated Scheduling Strategy of U-Shaped Automated Container Terminal Considering Energy Consumption. *Comput. Ind. Eng.* **2022**, *174*, 108804. [[CrossRef](#)]
36. Xu, B.; Jie, D.; Li, J.; Yang, Y. Green Integrated Scheduling of U-Shaped Automated Terminals under Uncertainty. *Comput. Integr. Manuf. Syst.* **2022**, *28*, 4057–4068. [[CrossRef](#)]
37. Bish, E.K. A Multiple-Crane-Constrained Scheduling Problem in a Container Terminal. *Eur. J. Oper. Res.* **2003**, *144*, 83–107. [[CrossRef](#)]
38. Gao, L.; Zhang, G.H.; Wang, X.J. *Intelligent Algorithms for Scheduling in Flexible Job Shops and Their Applications*; Huazhong University of Science and Technology Press: Wuhan, China, 2012.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.