Mixed-Integer Nonlinear Programming for Energy-Efficient Container Handling Formulation and Customized Genetic Algorithm

Xin, Jianbin; Meng, Chuang; D'Ariano, Andrea; Wang, Dongshu; Negenborn, Rudy R.

DOI
10.1109/TITS.2021.3094815

Publication date
2022

Document Version
Final published version

Published in
IEEE Transactions on Intelligent Transportation Systems

Mixed-Integer Nonlinear Programming for Energy-Efficient Container Handling: Formulation and Customized Genetic Algorithm

Jianbin Xin, *Member, IEEE*, Chuang Meng, Andrea D'Ariano, Dongshu Wang, and Rudy R. Negenborn

***Abstract*— Energy consumption is expected to be reduced while maintaining high productivity for container handling. This paper investigates a new energy-efficient scheduling problem of automated container terminals, in which quay cranes (QCs) and lift automated guided vehicles (AGVs) cooperate to handle inbound and outbound containers. In our scheduling problem, operation times and task sequences are both to be determined. The underlying optimization problem is mixed-integer nonlinear programming (MINLP). To deal with its computational intractability, a customized and efficient genetic algorithm (GA) is developed to solve the studied MINLP problem, and lexicographic and weighted-sum strategies are further considered. An ϵ-constraint algorithm is also developed to analyze the Pareto frontiers. Comprehensive experiments are tested on a container handling benchmark system, and the results show the effectiveness of the proposed lexicographic GA, compared to results obtained with two commonly-used metaheuristics, a commercial MINLP solver, and two state-of-the-art methods.**

***Index Terms*— Automated container terminals, energy efficiency, mixed-integer nonlinear programming, genetic algorithm.**

I. Introduction

OVER 60% of worldwide deep-sea general cargo is transported by containers [1], and global container market demand is projected to increase by around 3% in 2020 and 2021 [2]. As a freight transport hub, a container terminal serves as the interface between different transport modalities and provides flexibility and scalability to cover various geographical areas [3], [4]. As a result, the performance of container terminals is crucial to freight transport.

During container terminal operations, several challenges need to be addressed by operators both for customers and society. The first challenge is the terminal facing incremental arrival and departure containers with container ships. The volume of the container ship has reached 23,000 TEU in 2019, and this number is predicted to grow further in the near future [4], [5]. If no proper measures are taken, the turnaround time of a container vessel could increase considerably. In such circumstances, the handling capacity must be maximized to reduce the turnaround time of a container vessel.

Meanwhile, the container terminal industry is under significant pressure to meet not only economic but also environmental criteria [6], [7]. CO_2 emissions resulting from the energy consumption at containers terminals are required to be reduced. In addition to using electric machines [8], an effective way is to reduce the great amount of energy consumed in the terminal. It is noticed that the yearly electricity consumption of a typical container terminal can be up to 45,000 MWh with a yearly throughput of 4,260,000 TEU [9]. Ideally, the terminal operators expect to improve energy efficiency of the operations, *i.e., using less energy to achieve the level of service*. Therefore, reducing energy consumption without deteriorating the handling capacity of container terminals becomes a great challenge for researchers in the domain of transportation and logistics.

Motivated by the challenge from the industry, we investigate a new energy-aware scheduling problem of automated container terminals at the operational level. We aim to minimize the operational energy consumption as much as possible, while keeping a high handling capacity. The energy reduction also further decreases the terminal operation costs, which is beneficial to terminal operators. For this research problem, we consider the completion time (makespan) and the kinetic energy consumption of transporting containers as the objectives and present a new mathematical formulation to minimize the energy consumption straightforwardly. The underlying optimization problem is computationally intractable, and an efficient algorithm is also developed to handle complex instances.

The remainder of this paper is structured as follows: In Section II, the related literature is reviewed and the contributions are highlighted, Section III presents the problem statement and formulates the mathematical optimization problem of terminal operations. In Section IV, a dedicated metaheuristic algorithm is proposed for solving the formulated optimization problem. Section V presents the case studies to

Manuscript received 6 November 2020; revised 19 May 2021; accepted 30 June 2021. Date of publication 14 July 2021; date of current version 9 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61703372, Grant 61903304, and Grant 61603345; and in part by the Outstanding Foreign Scientist Project in Henan Province under Grant GZS2019008. The Associate Editor for this article was F. Chu. *(Corresponding author: Dongshu Wang.)*

Jianbin Xin, Chuang Meng, and Dongshu Wang are with the School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China (e-mail: j.xin@zzu.edu.cn; zzumengchuang@163.com; wangdongshu@zzu.edu.cn).

Andrea D'Ariano is with the Department of Engineering, Roma Tre University, 00146 Roma, Italy (e-mail: a.dariano@dia.uniroma3.it).

Rudy R. Negenborn is with the Department of Marine and Transport Technology, Delft University of Technology, 2628 Delft, The Netherlands (e-mail: r.r.negenborn@tudelft.nl).

Digital Object Identifier 10.1109/TITS.2021.3094815

test the proposed approach and analyzes the compared results. Section VI concludes this paper and gives future research directions.

II. Literature Review

A. Related Work

A container terminal represents a transport hub, exchanging containers between different transport modalities (vessel, barge, train, and truck) [10]. As automation could noticeably increase throughput and decrease the costs of container terminals [11], the research on automated container terminals has become a hot topic in recent years. For automated container terminals, three types of planning problems are categorized: strategic, tactical, and operational [12]. Strategic planning problems study the terminal layout and the machine selection [13], [14]. Tactical problems concentrate on the capacity level of machines, and the necessary machine number for completing operations efficiently is investigated [15]. Regarding the operational problems, the detailed operations for transporting containers by the pieces of machines are determined.

The operational level relates to the most complex processes of the terminal, covering from the quayside area connecting the waterway to the landside area linking the hinterland. Since container vessels are the major customers of the terminal, much attention has been paid in quayside-related operations [12], including long-term decision planning (e.g., berth allocation [16]) and short-term decision planning (e.g., quay crane scheduling [17]).

For the short-term planning problem of container terminals, the quayside operations of Quay Cranes (QCs) and related machines are scheduled to reduce the turnaround time of each vessel staying at the terminal. In automated container terminals, Automated Guided Vehicles (AGVs) are commonly used for transporting containers between the quayside and the landside [18], [19]. As the AGV requires close cooperation with QCs and Yard Cranes (YCs) when unloading and loading containers, the scheduling problems of QCs are typically integrated with the operations of AGVs and YCs [20], [21]. In [20], the integrated scheduling problem of QCs, AGVs, and YTs is studied for an export terminal to minimize the makespan of the total loading operations. The work [21] includes the path planning of AGVs into the integrated scheduling problem and formulates the overall problem as a bi-level programming. For non-automated terminals, the AGV is replaced by a Yard Truck (YT). The operations of the yard truck are similar to the AGVs, while the scheduling problem of QCs, and YTs, and YCs is also integrated to minimize the makespan of the container handling process [22].

Recently, newly developed lift AGVs are introduced in newly-built automated container terminals (e.g., APM terminal MV2, 2015). Unlike conventional AGVs, a lift AGV has two active lifting platforms, which enable the vehicle to lift and place containers independently on transfer racks in an interchange zone in front of the stacking cranes [23]. Using the lift AGVs, the interactive operation between each AGV and each YC can be decoupled for improving the operation efficiency [11]. It is noted that the handshake between the lift AGVs and the QCs is needed, and this operation is different from a straddle carrier, which moves a container independently between the corresponding quayside and stack [24]. In this paper, we focus on the integrated operations by using QCs and lift AGVs.

For scheduling container terminals, the objective in [20]–[22] focuses on the completion time indices, while energy consumption needs to be better studied. The importance of energy saving is highlighted in the works proposed by [6], [25] for more energy-efficient container operations and saving the operation cost further. To improve energy efficiency, several works [26]–[28] have been investigated. In [26], the energy consumption of all tasks and the total operation delays are unified into one objective function to be optimized when scheduling the integrated operations of QCs, YTs, and YCs; the influence of the route choices on the energy consumption is addressed. The energy consumption is also considered for scheduling the YTs in [27], [28]. A weighted bi-objective problem is studied in [27] to balance the efficiency and energy consumption. In [28], a single objective scheduling problem of yard crane is investigated for the minimal energy consumption by deciding the crane distribution and movements between different yard blocks.

Energy consumption optimization does not only depend on the vehicle routes [26], [28], but also relies on operation times of machines [9], [29]. In [9], [29], the shortest makespan is achieved at reduced kinetic energy consumption by optimizing the operation times in a simplified way. The nonlinear energy consumption is minimized without considering the detailed representation of energy consumption and the influence of task orders. Meanwhile, the optimized schedule in [9], [29] is limited to the case of a single QC and the transitions of transporting containers between different QCs are not considered. To obtain more energy-efficient operations for the case of multiple QCs, an integrated mathematical formulation and an efficient algorithm are required to minimize both the detailed energy consumption and the makespan.

B. Contributions of This Work

This paper investigates an operational energy-aware scheduling problem for automated container terminals. The contributions of this paper are given as follows:

- A new methodology is proposed to minimize energy consumption while maintaining a competitive makespan for completing operations at automated container terminals. These operations are performed by multiple quay cranes and multiple vehicles cooperatively. Task sequences and operation times are considered as decision variables and the optimization problem is formulated as an MINLP. To the best of our knowledge, this research problem has not been investigated yet in the literature.
- A customized and efficient genetic algorithm (GA) is developed for solving the considered MINLP, for which typical solution methods suffer from computational intractability. A new encoding scheme and customized algorithm procedures are proposed. This metaheuristic also builds up the key elements of the algorithm

TABLE I
RECENT REPRESENTATIVE SCHEDULING METHODS FOR AUTOMATED CONTAINER TERMINALS AT THE OPERATIONAL LEVEL

Publication	Areas		Energy-aware	Kinetic energy minimization		Model type
	single	integrated		indirect	direct	
[9]		√	√	√		MILP
[20]		√				MIP
[21]		√				MILP
[22]		√				MIP and CP
[26]		√	√	√		MILP
[27]	√		√	√		MILP
[28]	√		√	√		IP
[29]		√	√	√		MILP
Proposed		√	√		√	MINLP

framework to obtain the Pareto frontier by using the ϵ-constraint method. The proposed GA is comprehensively tested and compared with several approaches on a new set of instances extended from a given benchmark of the container terminal system.

As this paper builds on our previous work [9], [29], we need to clarify the potential improvements. First of all, this paper models energy efficiency accurately to better reduce energy consumption, while the previous works use a simplified way to maximize the transport time. Secondly, this paper considers multiple QCs, including loading and unloading operations simultaneously, while the earlier works consider the case of a single QC with unloading operations only. Finally, the optimization problem formulated here is MINLP, which is more challenging to be solved than the MIP, as provided by the models in [9], [29].

The distinctive features of this paper are given in Table I in comparison with the existing scheduling methods of container terminals in the literature (CP: Constraint Programming; IP: Integer Programming). Our approach minimizes the kinetic energy straightforwardly when scheduling integrated pieces of machines together.

III. HYBRID FLOW SHOP REPRESENTATION

This section introduces the research problem to be investigated for optimizing the makespan and energy consumption at the same time. The first part defines the studied research problem. Afterward, the mathematical model is described for the integrated operations in the terminal and the objective function is discussed.

A. Problem Statement

This paper focuses on the integrated operations of QCs and lift AGVs. As the handshake between the QC and the lift AGV is needed [11], the quayside operations can be described as a hybrid flow shop. The hybrid flow shop has a number of stages, and each task has to pass through these stages [30]. Each stage involves several identical machines for processing a part of a task in parallel. Each task is being processed by the same sequence of machines and handled for a certain processing time in each stage. As two types of machines are used for the quayside operations in each stage, the scheduling problem is described as a two-stage hybrid flow shop.

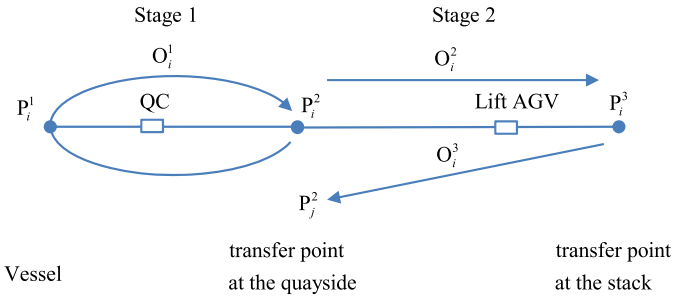

Fig. 1. Sequence of moving containers by using QCs and lift AGVs.

Important assumptions concerning the investigated scheduling problem are as follows:

- Inbound and outbound containers are both considered. An outbound container is available when an inbound one is unloaded by the lift AGV at a particular stack.
- Each task is performed only by one machine (QC or AGV) in each stage.
- The size of the containers is considered to be the same, which is 20 feet.
- The destinations of all inbound containers in the stack are known in advance.
- The position constraints of containers are ignored when loading or unloading them in the vessel, as the scope of applications is restricted to vessels with a small number of containers.
- The service time for transferring a container at each transfer point is ignored.
- The locations of inbound containers and outbound containers handled by the same QC and the stack are assumed to be very close, and the transition time between handling an outbound container and an inbound container is neglected.

B. Model of Operations

In our two-stage flow shop formulation, a task is defined as a complete process of moving a specific inbound container from the vessel to its unloading point in the stack and then moving a specific outbound container back to the vessel from the same stack. A task is performed by two types of equipment (QC and lift AGV). We define i and j as the indexes of inbound containers, while we use k to mark outbound containers.

The considered container transport operations for task i are defined in Fig. 1. P_i^1 is described as the location of the inbound

TABLE II
LIST OF VARIABLES AND PARAMETERS FOR MODELING THE TWO-STAGE FLOW SHOP

Sets	Definition
Φ	Set of all processed tasks for inbound containers ($\|\Phi\| = n$);
Φ_1	Union set of Φ and dummy task 0 ($\|\Phi\| = n+1$);
Φ_2	Union set of Φ and dummy task $n+1$ ($\|\Phi\| = n+1$);
Ψ	Set of all outbound containers ($\|\Psi\| = n$) ;
Parameters	**Definition**
n	Number of tasks to be processed;
m_1	Number of QCs available in the terminal;
m_2	Number of lift AGVs available in the terminal;
t_{ij}	Minimal transport time for transition from location P_i^3 to P_j^2;
t_i^{qc}	Processing time of operation O_i^1 carried out by the QC ;
R	a large positive number.
Variables	**Definition**
x_{ij}	$x_{ij} = 1$ if task j is processed directly after task i in stage 1, otherwise $x_{ij} = 0$;
y_{ij}	$y_{ij} = 1$ if task j is processed directly after task i in stage 2, otherwise $y_{ij} = 0$;
z_{ik}	$z_{ik} = 1$ if outbound container k is assigned to task i for each AGV, otherwise $z_{ik} = 0$;
t_i^{in}	Processing time of operation O_i^2 carried out by the AGV for task i ;
t_i^{out}	Processing time of operation O_i^3 carried out by the AGV for task i;
a_i	Start time of O_i^1, i.e., the time at which the QC handling task i leaves P_i^2;
b_i	Start time of O_i^2, at which the lift AGV handling task i leaves P_i^2;
c_i	Departure time of O_i^3 at which the lift AGV handling task i leaves P_i^3.
c	makespan (the completion time of all tasks)

container i in the vessel. P_i^2 is defined as the transfer point, where the inbound container i is moved from a particular QC to a particular lift AGV (P_i^2 applies for the inbound container j). P_i^3 is defined as the transfer point where the inbound container i is unloaded from a particular lift AGV to the buffer area in the stack. Fig. 1 also shows that in the proposed paradigm one task involves three operations, denoted by O_i^1, O_i^2 and O_i^3. These three operations are described in detail below.

In Stage 1, the operation O_i^1 contains the QC motion from P_i^2 to P_i^1 and the QC motion from P_i^1 back to P_i^2. For the QC motion from P_i^2 to P_i^1, a specific outbound container is loaded after that each QC starts unloading inbound containers. For each QC, before unloading the first inbound container, there is no outbound container to be processed. For the QC motion from P_i^2 to P_i^1, the inbound container i is unloaded. As we focus on energy-efficient scheduling of lift AGVs, for the sake of simplicity, these two successive motions for each task are included in one operation (O_i^1). The processing time of O_i^1 (defined as t_i^{qc}) mainly depends on the location of inbound container i, as each outbound container does not require a fixed location in the vessel.

Stage 2 comprises operations O_i^2 and O_i^3, which respectively denote the lift AGV's motion from P_i^2 to P_i^3 with inbound container i and the AGV motion back from P_i^3 to P_j^2 with outbound container k, after unloading inbound container i. P_i^3 and P_j^2 are the origin and destination of the outbound container k, if task j is processed after task i by a particular lift AGV. We assume that the stacking schedule is optimized for locating a particular container in the stack area.

Let there be n tasks of moving an inbound container from the vessel to the stack and then moving an outbound container from the stack back to the vessel. Let Φ be the set of tasks ($|\Phi| = n$). The hybrid flow shop has time relationships among the processes of each task by every machine in each stage. For a machine at a certain stage, this machine has a time constraint on every two successive tasks. For a certain task processed in successive stages, the task also has time constraints for guaranteeing the stage sequence. Two dummy tasks 0 and $n+1$ are introduced to model the initial task and the final task. We then define $\Phi_1 = \Phi \cup \{0\}$ and $\Phi_2 = \Phi \cup \{n+1\}$. These time constraints can be described as follows:

$$a_i + R(1 - x_{0j}) \geq 0 \quad \forall j \in \Phi \tag{1}$$

$$a_j + R(1 - x_{ij}) \geq b_i \quad \forall i \in \Phi, \forall j \in \Phi \tag{2}$$

$$a_i + t_i^{\mathrm{qc}} \leq b_i \quad \forall i \in \Phi, \tag{3}$$

$$b_j + R(1 - y_{ij}) \geq c_i + t_i^{\mathrm{out}} \quad \forall i \in \Phi, \forall j \in \Phi \tag{4}$$

$$b_i + t_i^{\mathrm{in}} \leq c_i \quad \forall i \in \Phi, \tag{5}$$

$$t_i^{\mathrm{out}} + R(1 - y_{ij}) \geq t_{ij} \quad \forall i \in \Phi, \forall j \in \Phi_2 \tag{6}$$

$$c \geq c_i + t_i^{\mathrm{out}} \quad \forall i \in \Phi, \tag{7}$$

where Inequality (1) initializes the first task handled by the QC. Inequality (2) gives the relation among tasks i and j handled by a particular QC. Inequality (3) ensures that task i is processed by an AGV after a QC. Inequality (4) links tasks i and j handled by a particular AGV. Inequality (6) ensures that O_i^3 is handled after O_i^2 by a particular AGV. Inequality (6) gives the constraint of operation time t_i^{unload} for the transition between task i and task j. Inequality (7) defines the makespan c to finish all operations. A summary definition of all relevant variables and parameters is given in Table II.

For each machine in every stage, it should be guaranteed that there is exactly one preceding task and one succeeding task. For this reason, decision variables x_{ij} and y_{ij} have additional equality constraints. For the first task j ($j \in \Phi$) in each stage, x_{ij} and y_{ij} ($i \in \Phi, j \in \Phi, i \neq j$) must be zero; for the last task i ($i \in \Phi$) in each stage, x_{ij} and y_{ij} ($i \in \Phi, j \in \Phi, i \neq j$) must be zero. Each outbound container is assigned to a special task. With these sets Φ_1 and Φ_2 we define:

$$\sum_{j \in \Phi_2} x_{ij} = 1, \quad \forall i \in \Phi \tag{8}$$

$$\sum_{i\in\Phi_1} x_{ij} = 1, \quad \forall j \in \Phi \tag{9}$$

$$\sum_{j\in\Phi} x_{0j} = m_1, \tag{10}$$

$$\sum_{i\in\Phi} x_{i(n+1)} = m_1, \tag{11}$$

$$\sum_{j\in\Phi_2} y_{ij} = 1, \quad \forall i \in \Phi \tag{12}$$

$$\sum_{i\in\Phi_1} y_{ij} = 1, \quad \forall j \in \Phi \tag{13}$$

$$\sum_{j\in\Phi} y_{0j} = m_2, \tag{14}$$

$$\sum_{i\in\Phi} y_{i(n+1)} = m_2, \tag{15}$$

$$\sum_{k\in\Psi} z_{ik} = 1, \quad \forall i \in \Phi \tag{16}$$

$$\sum_{i\in\Phi} z_{ik} = 1, \quad \forall k \in \Psi, \tag{17}$$

where Equalities (8) and (9) guarantee that, for each task in Stage 1, there is exactly one preceding task and one succeeding task assigned. Equalities (10) and (11) guarantee that m_1 QCs are employed. Equalities (12) and (13) enforce that, for each task in Stage 2, there is just one preceding task and one succeeding task assigned. Equalities (14) and (15) ensure that in total m_2 lift AGVs are used. Equalities (16) and (17) certify the matching of task i and outbound container k.

With constraints (1)-(17), the integrated operations involving QCs and lift AGVs are modeled as a two-stage hybrid flow shop. In this hybrid flow shop, the completion time of task i in each stage and the task sequences by every machine in each stage are decision variables. The makespan c is defined within those decision variables. The optimization problem will be extended by considering the energy consumption of the lift AGVs, as discussed in the next subsection.

C. Objective Function and Problem Formulation

This section formulates the bi-objective optimization problem in the lexicographical form and weighted-sum form. The objective of makespan c is represented by J_{C}. Another objective defined as J_{E} represents the kinematic energy consumption of AGVs. Here we show how J_{E} is calculated in details.

In general, for a certain distance s at a constant speed v without a slope, the vehicle energy consumption E is computed approximately as follows:

$$E \approx \frac{1}{2}mv^2 + mgC_r s, \tag{18}$$

where m is the total vehicle weight (kg), g is the gravity coefficient, and C_r is the coefficient of rolling resistance. The two parts represent the energy consumed for the acceleration and the rolling resistance of the vehicle. It is noted that, since the lift AGV is a heavy vehicle (at least 30,000 kg), the energy of air drag is significantly small and this part is neglected when computing E.

Based on equation (19) for a single AGV operation, the energy consumption calculation can be extended for multiple operations. Regarding the inbound AGV operations, their total energy (defined as $J_{\mathrm{E}}^{\mathrm{in}}$), is given as follows:

$$J_{\mathrm{E}}^{\mathrm{in}} \approx \sum_{i=1}^{n} \frac{1}{2} m_i^{\mathrm{in}} \left(\frac{s_i}{t_i^{\mathrm{in}}}\right)^2 + \sum_{i=1}^{n} m_i^{\mathrm{in}} g C_r s_i, \tag{19}$$

where s_i is the distance of the lift AGV when handling inbound container i. m_i^{in} is the weight of inbound container i. Here, the distance-over-time ratios s_i/t_i^{in} represent the average velocity of each AGV during operation O_i^2.

The outbound AGV operations are more complex than the inbound operations, because each inbound container has a particular destination in the stack, while every outbound one has an unknown destination in the quayside. Furthermore, a particular outbound container is assigned for inbound container i arriving in the stack. Considering these above constraints, the related total energy of the outbound AGV operations, which is defined as $J_{\mathrm{E}}^{\mathrm{out}}$, is formulated as follows:

$$J_{\mathrm{E}}^{\mathrm{out}} \approx \sum_{i=1}^{n} \frac{1}{2} \sum_{k\in\Psi} z_{ik} m_k^{\mathrm{out}} \left(\frac{\sum_{j\in\Phi_2} s_{ij} y_{ij}}{t_i^{\mathrm{out}}}\right)^2 + \sum_{i=1}^{n} \sum_{k\in\Psi} z_{ik} m_k^{\mathrm{out}} g C_r \sum_{j\in\Phi_2} s_{ij} y_{ij}, \tag{20}$$

where s_{ij} is the distance of outbound container k between inbound container i and inbound container j for each AGV. The function $\sum_{j\in\Phi_2} s_{ij} y_{ij}$ gives the distance for outbound container k, since there is one outbound container assigned to task i. The term $\sum_{k\in\Psi} z_{ik} m_k^{\mathrm{out}}$ represents the chosen m_k^{out} for task i.

With $J_{\mathrm{E}}^{\mathrm{in}}$ and $J_{\mathrm{E}}^{\mathrm{out}}$, the total consumption for operating AGVs J_{E} is computed by summing up these two functions as follows:

$$J_{\mathrm{E}} = J_{\mathrm{E}}^{\mathrm{in}} + J_{\mathrm{E}}^{\mathrm{out}}. \tag{21}$$

From (19), we can see that $J_{\mathrm{E}}^{\mathrm{in}}$ is a convex term. However, it is observed from (20) that $J_{\mathrm{E}}^{\mathrm{out}}$ is a non-convex nonlinear function, because the Hessian matrices of $z_{ik} y_{ij}^2 (t_i^{\mathrm{out}})^{-2}$ and $z_{ik} y_{ij}$ are both not positive definite.

For the energy-efficient scheduling, we consider two strategies (lexicographical and weighted-sum) to optimize the two objectives J_{C} and J_{E}. For the lexicographical strategy, the optimization problem is formulated as follows:

$$\min_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{t},\mathbf{a},\mathbf{b},\mathbf{c}} J_{\mathrm{E}} \tag{22}$$

$$\text{subject to } J_{\mathrm{C}} \le J_{\mathrm{C}}^*, \tag{23}$$

$$t_i^{\mathrm{in}} \ge t_{i,\min}^{\mathrm{in}}, \tag{24}$$

and subject to the equalities and inequalities (1)-(17),

where $t_{i,\min}^{\mathrm{in}}$ is the minimal operation time for O_i^2, $\mathbf{a}^{\mathrm{T}} = [a_1, a_2, \cdots, a_n]$, $\mathbf{b}^{\mathrm{T}} = [b_1, b_2, \cdots, b_n]$, $\mathbf{c}^{\mathrm{T}} = [c_1, c_2, \cdots, c_n]$, $\mathbf{x}^{\mathrm{T}}$ is a vector comprised of all the elements in the set of indices $\{x_{ij}\}_{i\in\Phi_1, j\in\Phi_2, i\neq j}$, $\mathbf{y}^{\mathrm{T}}$ is a vector comprised of all the elements in the set of indices $\{y_{ij}\}_{i\in\Phi_1, j\in\Phi_2, i\neq j}$, $\mathbf{z}^{\mathrm{T}}$ is a vector comprised of all the elements in the set of indices

$\{z_{ik}\}_{i\in\Phi,k\in\Psi}$, and $\mathbf{t}^{\mathrm{T}}$ is the vector of $\{t_i^{\mathrm{in}}\}$ and $\{t_i^{\mathrm{out}}\}$. J_{C}^* denotes the minimal makespan subject to constraints (1)-(17) and (24).

For the weighted-sum strategy, the detailed optimization problem is given as follows:

$$\min_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{t},\mathbf{a},\mathbf{b},\mathbf{c}} J_{\mathrm{C}} + \lambda J_{\mathrm{E}} \tag{25}$$

subject to constraints (1)-(17) and (24). λ is the weighting factor in which the chosen unit can be used for normalization. To highlight the importance of J_{C}, a small positive value of λ is adopted in our experiments.

The above two problems are both non-convex MINLP, which is considered difficult to be solved. In the next section, we propose a dedicated algorithm to address this computational challenge.

IV. Customized Genetic Algorithm

In this section, a customized genetic algorithm is developed to efficiently solve the non-convex MINLP problem formulated in Section III. The non-convex MINLP is known to be NP-hard [31], and commercial MINLP solvers, like Baron [32], cannot provide high-quality solutions in a reasonable time. We propose to solve the MINLP using an advanced genetic algorithm.

Genetic algorithms are considered as an effective metaheuristic approach for solving constrained optimization problems. Compared to other metaheuristics (like variable neighborhood search and tabu search), GA's efficiency has been verified by the large number of works related to scheduling problems of container terminal [20] and works dealing with MINLP problems [33]. The GA has a relatively simple algorithmic structure, but this metaheuristic has a good ability to diversify the search in the feasible region of the search space. Therefore, the GA is regarded as our core method to design a highly customized algorithm for efficiently solving the considered MINLP problem.

For container terminals, GAs have been developed for addressing the energy-aware scheduling problem, and the related optimization problem is limited to Mixed Integer Programming (MIP) [26], [27]. As we model the detailed vehicle energy-saving operation, the resulting MINLP problem is more challenging; so far, no dedicated algorithm has been developed. The following parts give the encoding scheme and the customized algorithmic procedures used by the proposed GA for solving the MINLP problem modeled as a hybrid flow shop formulation.

A. Encoding

In GAs, a population (a set of possible solutions) needs to be initialized and further optimized. The encoding scheme for each solution is highly relevant for the solution quality. For energy-efficient scheduling problems of container terminals, the encoding scheme of exiting GAs is designed for the solution to MIPs only [26], [27], and this encoding scheme cannot be used for MINLPs. In this part, we develop a new encoding scheme for constructing the solutions suitable for the considered MINLP.

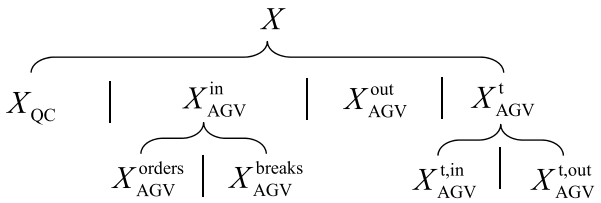

Fig. 2. Composition of the solution X.

TABLE III
Encoding of the Composed X in Fig. 2

Notation	Explanations
X_{QC}	Listed task orders performed by each QC
$X_{\mathrm{AGV}}^{\mathrm{in}}$	Listed assigned tasks in order for each AGV
$X_{\mathrm{AGV}}^{\mathrm{out}}$	Listed assigned outbound container for each task
$X_{\mathrm{AGV}}^{\mathrm{orders}}$	Listed task orders without separation for all AGVs
$X_{\mathrm{AGV}}^{\mathrm{breaks}}$	Listed positions of breaking tasks for each AGV
$X_{\mathrm{AGV}}^{\mathrm{t}}$	Listed operation time for inbound and outbound containers
$X_{\mathrm{AGV}}^{\mathrm{t,in}}$	Listed operation time for each inbound container
$X_{\mathrm{AGV}}^{\mathrm{t,out}}$	Listed operation time for each outbound container

In the developed encoding scheme, we consider a mixed sequence-time encoding scheme, which includes task sequences completed by different types of machines and operation times of AGVs for each task.

The overall encoded solution is defined as X, and the composition is illustrated in Fig. 2. The solution X contains four parts: X_{QC}, $X_{\mathrm{AGV}}^{\mathrm{in}}$, $X_{\mathrm{AGV}}^{\mathrm{out}}$, and $X_{\mathrm{AGV}}^{\mathrm{t}}$. The first three parts (X_{QC}, $X_{\mathrm{AGV}}^{\mathrm{in}}$, and $X_{\mathrm{AGV}}^{\mathrm{out}}$) correspond to task sequences (binary decision variables in Table II) and the last part ($X_{\mathrm{AGV}}^{\mathrm{t}}$) relates to operation times of inbound and outbound tasks performed by the AGVs (integer decision variables in Table II).

X_{QC} lists the task orders performed by each QC. $X_{\mathrm{AGV}}^{\mathrm{in}}$, which consists of $X_{\mathrm{AGV}}^{\mathrm{orders}}$ and $X_{\mathrm{AGV}}^{\mathrm{breaks}}$ gives the sequenced tasks for each AGV. $X_{\mathrm{AGV}}^{\mathrm{in}}$ lists the order between all the tasks without separation, while $X_{\mathrm{AGV}}^{\mathrm{breaks}}$ provides the positions of breaking tasks to terminate a task order for a particular AGV. The number of elements in $X_{\mathrm{AGV}}^{\mathrm{breaks}}$ is $m_2 - 1$. $X_{\mathrm{AGV}}^{\mathrm{out}}$ gives the assigned ordered outbound tasks for each task. $X_{\mathrm{AGV}}^{\mathrm{t}}$ contains the operation times of inbound and outbound tasks; the parts of inbound and outbound tasks are denoted by $X_{\mathrm{AGV}}^{\mathrm{t,in}}$ and $X_{\mathrm{AGV}}^{\mathrm{t,out}}$. $X_{\mathrm{AGV}}^{\mathrm{t,in}}$ includes the set of decision variables t_i^{in}, while $X_{\mathrm{AGV}}^{\mathrm{t,out}}$ contains the set of decision variables t_i^{out}. The elements of $X_{\mathrm{AGV}}^{\mathrm{t}}$ are all integers.

We use a sample solution in Table IV to exemplify the constructed encoding used by the algorithm. This solution gives the schedule of 2 QCs and 2 AGVs for processing 4 inbound containers (tasks 1-4) and 4 outbound containers (tasks 5-8) at two stacks. The task sequences are as follows: QC1: 1→2, QC2: 3→4, AGV1: 1→3 (the one listed in $X_{\mathrm{AGV}}^{\mathrm{breaks}}$), AGV2: 2→4. The outbound containers 6, 7, 5, and 8 are operated after the inbound containers 1-4, respectively. Implicitly, the overall container sequence are: AGV1: 1→6→3→5, AGV2: 2→7→4→8. The operation times for the inbound containers are as follows: $t_1^{\mathrm{in}} = 25$, $t_2^{\mathrm{in}} = 25$, $t_3^{\mathrm{in}} = 28$, and $t_4^{\mathrm{in}} = 26$. The operation times for the outbound containers are as follows: $t_1^{\mathrm{out}} = 30$, $t_2^{\mathrm{out}} = 24$, $t_3^{\mathrm{out}} = 32$, and $t_4^{\mathrm{out}} = 27$.

TABLE IV
ILLUSTRATIVE EXAMPLE OF THE ENCODED SOLUTION X

Notation	Details
X_{QC}	1-2 3-4
$X_{\mathrm{AGV}}^{\mathrm{orders}}$	1-3-2-4
$X_{\mathrm{AGV}}^{\mathrm{breaks}}$	2
$X_{\mathrm{AGV}}^{\mathrm{out}}$	6 7 5 8
$X_{\mathrm{AGV}}^{\mathrm{t,in}}$	25 25 28 26
$X_{\mathrm{AGV}}^{\mathrm{t,out}}$	30 24 32 27

B. Energy-Efficient Algorithm

1) Choices of Operators: In this part, we detail the operators of the developed GA for solving the MINLP based on the two-stage flow shop representation presented in Section III. This flow shop scheduling problem can be decomposed as two multiple traveling salesman problems (mTSPs). For each mTSP in each stage, we construct the strategies of selection and mutation of GAs. The crossover strategy is not considered in the algorithm as this strategy results in longer computation and may deteriorate the solution quality [34].

Regarding the selection, the top-ranking method is employed. At each iteration, the best 1/8 solutions are selected from the entire population as the elites and retained until the next iteration.

For the mutation used in this paper, three operators, namely, *flip, swap*, and *slide*, are considered in the proposed algorithm. The flip mutation works by randomly choosing two positions in the chromosome and reversing the order in which the values appear between those positions. The swap operator randomly swaps the values of two positions in the solution chromosome. For the slide operator, two positions in the chromosome are randomly selected, and the contents between these two positions move one position to the left. These three operators can be described as follows [34]:

$$flip(\pi, p_1, p_2) \triangleq \pi'(p_1 : p_2) = \pi(p_1, -1, p_2), \tag{26}$$

$$swap(\pi, p_1, p_2) \triangleq \pi'(p_1) = \pi(p_2), \pi'(p_2) = \pi(p_1), \tag{27}$$

$$slide(\pi, p_1, p_2) \triangleq \pi'(p_1 : p_2) = [\pi(p_1 + 1 : p_2), \pi(p_1)], \tag{28}$$

where π is a segment of the chromosome, and π is a permutation. For this paper, π can be X_{QC}, $X_{AGV}^{\mathrm{orders}}$, $X_{\mathrm{AGV}}^{\mathrm{out}}$ and $X_{\mathrm{AGV}}^{\mathrm{t}}$. p_1 and p_2 represent two positions of the segment.

2) Main Procedures: Algorithm 1 presents the pseudocode of the lexicographic GA developed for solving the MINLP. In Algorithm 1, $iter$, $iter_1$ and $iter_{\max}$ denote the iteration index, the maximum number of iterations to minimize the makespan, and the total maximum number of iterations, respectively. $F(X)$ is the fitness function of solution X. The designed algorithm is expected to provide high-quality solutions while satisfying the constraints of the considered MINLP problem.

In the designed lexicographic GA, for minimizing the single objective J_{C}, the fitness function $F(X)$ is calculated as $F(X) = J_{\mathrm{C}} + p(X)$. $p(X)$ is the penalty function to deal with the speed constraint and is defined as follows:

$$p(X) = \begin{cases} 0, & \upsilon_{\min} \le \dfrac{s_i}{t_i^{\mathrm{in}}} \le \upsilon_{\max}, \\ & \text{and } \upsilon_{\min} \le \dfrac{\sum_{j \in \Phi_2} s_{ij} y_{ij}}{t_i^{\mathrm{out}}} \le \upsilon_{\max} \\ R, & \text{otherwise.} \end{cases} \tag{29}$$

where R is the same parameter as given in Table III. When selecting the best solutions as elites, for the next iteration, the solutions that are not satisfied with the speed constraints will be abandoned due to its high penalty.

Algorithm 1 The Lexicographic GA for the Considered MINLP

1: $iter = 0$
2: initialize $P(iter)$
3: **while** $iter \le iter_{\max}$ **do**
4: **if** $iter \le iter_1$ **then**
5: **for** $p = 1$ to N_{p} **do**
6: evaluate the fitness $F(X) = J_{\mathrm{c}} + p(X)$
7: **end for**
8: record the minimal c as c^*
9: **else**
10: **for** $p = 1$ to N_{p} **do**
11: evaluate $F(X) = J_{\mathrm{e}} + p(X)$ under $c \le c^*$
12: **end for**
13: **end if**
14: select 1/8 of $P(iter)$ with the lowest fitness as $P_1(iter)$
15: **for** $p = 1$ to $N_{\mathrm{p}}/8$ **do**
16: flip $P_{1,\mathrm{QC}}(iter)$ or $P_{1,AGV}^{\mathrm{orders}}(iter)$, flip $P_{1,\mathrm{AGV}}^{\mathrm{out}}(iter)$ and $P_{1,\mathrm{AGV}}^{\mathrm{t}}(iter)$, and keep $P_{1,\mathrm{AGV}}^{\mathrm{breaks}}(iter)$, to construct $P_2(iter)$
17: swap $P_{1,\mathrm{QC}}(iter)$ or $P_{1,AGV}^{\mathrm{orders}}(iter)$, swap $P_{1,\mathrm{AGV}}^{\mathrm{out}}(iter)$ and $P_{1,\mathrm{AGV}}^{\mathrm{t}}(iter)$, and keep $P_{1,\mathrm{AGV}}^{\mathrm{breaks}}(iter)$, to construct $P_3(iter)$
18: slide $P_{1,\mathrm{QC}}(iter)$ or $P_{1,AGV}^{\mathrm{orders}}(iter)$, slide $P_{1,\mathrm{AGV}}^{\mathrm{out}}(iter)$ and $P_{1,\mathrm{AGV}}^{\mathrm{t}}(iter)$, and keep $P_{1,\mathrm{AGV}}^{\mathrm{breaks}}(iter)$, to construct $P_4(iter)$
19: random $P_{1,AGV}^{\mathrm{breaks}}(iter)$ and keep the other parts of $P_1(iter)$ to construct $P_5(iter)$
20: flip $P_{1,\mathrm{QC}}(iter)$ or $P_{1,AGV}^{\mathrm{orders}}(iter)$, flip $P_{1,\mathrm{AGV}}^{\mathrm{out}}(iter)$ and $P_{1,\mathrm{AGV}}^{\mathrm{t}}(iter)$, and random $P_{1,\mathrm{AGV}}^{\mathrm{breaks}}(iter)$, to construct $P_6(iter)$
21: swap $P_{1,\mathrm{QC}}(iter)$ or $P_{1,AGV}^{\mathrm{orders}}(iter)$, swap $P_{1,\mathrm{AGV}}^{\mathrm{out}}(iter)$ and $P_{1,\mathrm{AGV}}^{\mathrm{t}}(iter)$, and random $P_{1,\mathrm{AGV}}^{\mathrm{breaks}}(iter)$, to construct $P_7(iter)$
22: slide $P_{1,\mathrm{QC}}(iter)$ or $P_{1,AGV}^{\mathrm{orders}}(iter)$, slide $P_{1,\mathrm{AGV}}^{\mathrm{out}}(iter)$ and $P_{1,\mathrm{AGV}}^{\mathrm{t}}(iter)$, and random $P_{1,\mathrm{AGV}}^{\mathrm{breaks}}(iter)$, to construct $P_8(iter)$
23: **end for**
24: $P(iter + 1) = P_1(iter) \cup P_2(iter) \cup P_3(iter) \cup P_4(iter) \cup P_5(iter) \cup P_6(iter) \cup P_7(iter) \cup P_8(iter)$
25: $iter = iter + 1$
26: **end while**

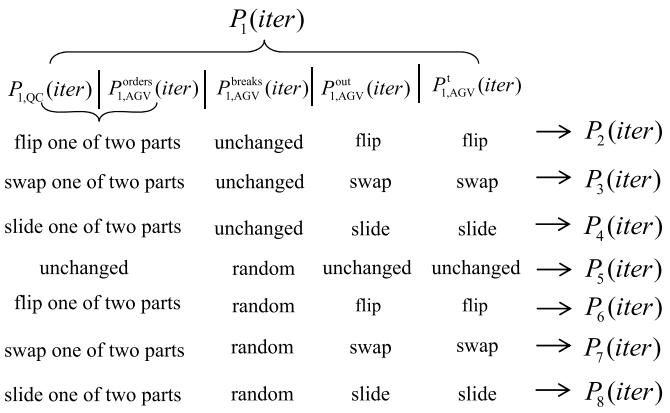

Fig. 3. Illustrating the composition of population P for each iteration.

Algorithm 2 The ϵ-Constraint Method

1: $iter = 0$
2: min J_C subject to constraints (1)-(17), and set $\beta_1 = J^*_C(iter)$, $\varphi_2 = J''_E(iter)$
3: min J_E subject to constraints (1)-(17), and set $\beta_2 = J^*_E(iter)$
4: insert the pair (β_1,φ_2) in the Pareto solution set Ψ
5: **while** $\beta_2 < \varphi_2$ **do**
6: $iter = iter + 1$
7: min J_C subject to constraints (1)-(17) plus the constraint: $J_E < \varphi_2$, and set $\beta_1 = J^*_C(iter)$, $\varphi_2 = J''_E(iter)$
8: insert the pair (β_1,φ_2) in the Pareto solution set Ψ
9: **end while**
10: return the Pareto solution set Ψ

For minimizing the single objective J_E, $F(X)$ is computed as $F(X) = J_C + p(X)$. In the proposed algorithm, J_C is computed by the constraints (1)-(7) and J_E is evaluated by the equations (19)-(21).

$P(iter)$ is the entire population at iteration $iter$. $P_1(iter)$– $P_4(iter)$ are the populations used for the elite, flip, swap, and slide operations, respectively, without changing the breaking tasks. $P_5(iter)$– $P_8(iter)$ are the populations used for the elite, flip, swap, and slide operations, respectively, when updating the breaking tasks. Their sizes are considered as one-eighth of the entire population. These eight parts are illustrated in Fig. 3. The algorithm stops when the maximal number of the iterations has been reached.

In addition to the lexicographic GA, we also consider a weighted-sum formulation, in which J_C and J_E are given in the weighted-sum formulation as the formula (25). The procedures of the weight-sum GA can be easily implemented by replacing lines 4-13 in Algorithm 1 with *evaluate the fitness* $F(X) = J_C + \lambda J_C + p(X)$. The weighted objective $J_C + \lambda J_C$ is jointly incorporated into the fitness $F(X)$. These two GA configurations will be evaluated in the next section.

C. Computation of the Pareto Frontier

This subsection proposes an algorithm for computing the Pareto frontier between the makespan and energy consumption via the ϵ-constraint method. The ϵ-constraint method can be efficiently used for computing non-dominated solutions [35]. Here, we propose an iterative procedure based on the ϵ-constraint method and the developed GA in the previous section. At each iteration, we solve a single-objective optimization problem via the proposed GA for the bi-objective optimization problem. In the bi-objective problem, one performance indicator is optimized directly in the objective function, while the other performance indicator is indirectly optimized by inserting an additional bound constraint in the single-objective formulation.

Algorithm 2 describes the main steps of the proposed ϵ-constraint method. At each iteration $iter$, the values of $J^*_C(iter)$ and $J^*_E(iter)$ are computed individually. The proposed ϵ-constraint method at first initializes the number of iterations $iter$ and finds the optimal values of the two single-objective optimization problems. The value of the primary single-optimization problem (β_1, φ_2) is inserted into the solution set Ψ. After the initialization, a new single optimization problem is iteratively solved by adding an additional constraint on the value of the secondary indicator J_E. For each iteration, a new solution pair (β_1, φ_2) is added into the solution set Ψ. When the value of $J''_E(iter)$ is equal to $J^*_E(i_{\text{iter}})$, the entire iterative process is terminated.

V. Case Studies

This section discusses the computational results obtained with the numerical experiments carried out to assess the proposed GA for solving the considered energy-efficient scheduling problem. The first part introduces the benchmark system of the container handling system and the experimental setting. Then, the proposed GA configurations are compared with two commonly-used metaheuristics, two state-of-the-art scheduling methods, and a commercial solver.

A. Instances Generation and Settings

1) Benchmark: To evaluate the performance of the proposed method for improving energy efficiency, we consider the benchmark system proposed in [36]. The benchmark system includes a small container vessel, a quayside transport area, and 8 stacking blocks with 5 QCs and 10 lift AGVs, as shown in Fig. 4. For setting up the problem, a company specialized in advice on the design and simulation of terminals has provided the container terminal layout and insightful information on the studied problem. The parameter settings have been determined considering the physical capabilities of each piece of equipment (e.g., weight, velocity). Additional fundamental features can be found in [36].

Key parameters are given as follows: for lift AGVs, $v_{\max}$ is set to be 6 m/s; m_i^{in} and m_i^{out} are randomly generated between [40,60] (tons) including a dead AGV weight 35 tons. These parameters are suggested by [23].

To compare the different methods above, we extend the set of benchmark instances with more complex instances than the ones tested in [36]. The configurations of the proposed instances are given in Table V. For each configuration, ten

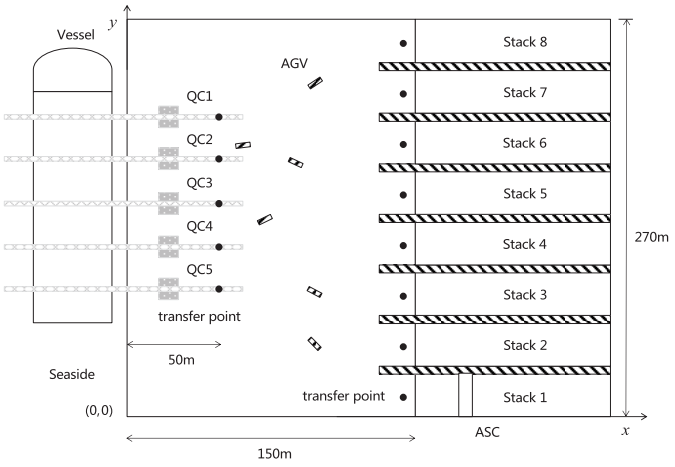

Fig. 4. Benchmark system of an automated container terminal [36].

TABLE V
SETTINGS OF THE NUMERICAL EXPERIMENTS EXTENDED FROM [36]

Case	Tasks	QCs	AGVs	Stacks
2QC-1	16	2	2	3
2QC-2	16	2	3	3
2QC-3	16	2	4	3
3QC-1	24	3	4	5
3QC-2	24	3	5	5
3QC-3	24	3	6	5
4QC-1	32	4	6	6
4QC-2	32	4	7	6
4QC-3	32	4	8	6
5QC-1	40	5	8	8
5QC-2	40	5	9	8
5QC-3	40	5	10	8
5QC-4	60	5	8	8
5QC-5	60	5	9	8
5QC-6	60	5	10	8
5QC-7	80	5	8	8
5QC-8	80	5	9	8
5QC-9	80	5	10	8
5QC-10	100	5	8	8
5QC-11	100	5	9	8
5QC-12	100	5	10	8

experiments are conducted to have a general comparison. In each experiment, the container locations in the vessel and the storage place are randomly generated.

The proposed lexicographic and weighted-sum GA are compared with two standard commonly-used metaheuristic algorithms, i.e., variable neighborhood search (VNS) and tabu search (TS), to select the most proper metaheuristic that suits the aim to reduce the energy consumption while keeping a competitive makespan. The tested VNS and TS are briefly described as follows:

- The implemented VNS in this paper is the basic version of VNS [37], which combines deterministic and random changes of neighborhoods. This algorithm consists of two phases: the shaking phase for the global search and the improvement phase for the local search. The encoding is the same as the proposed GA to construct the neighborhood. For the global search, one of the seven operations presented in Algorithm 1 (Lines 16-22) is randomly selected. For the local search, the seven operations in Algorithm 1 (same as above) are selected to modify the neighborhood solutions by following the sequencing strategy in Algorithm 1. The local search stops when the first improvement is achieved.
- Tabu search is a deterministic metaheuristic based on local search. The implemented TS follows the generic algorithmic procedures presented in [38]. The tabu list is used to escape from the local optimum and initially, the list is empty. The encoding is set the same as the proposed GA and the VNS. For generating the candidate list (similarly to the population of the GA), the neighborhood solutions of a candidate solution are modified by adopting one of the seven operations presented in Algorithm 1 (Lines 16-22) randomly chosen, and the probability for selecting each operation is equal. Regarding the aspiration criterion, if the tabu list contains all the seven operations, the operation with the best solution is removed from the tabu list. Here, no advanced intensification and diversification strategies are used.

In addition to the above metaheuristic algorithms, the following three state-of-the-art methods for solving the considered MINLP are also included for a further comparison:

- A two-phase method, which is a commonly-used heuristic to decompose the complexity of MINLPs [39]. The first phase solves an MILP formulation, in which task orders are optimized for the minimal makespan. The second phase solves a convex nonlinear programming (NLP), in which the operation times are optimized by following the obtained task orders of the first phase.
- Bi-objective MILP (B-MILP for short), in which the optimization problem is simplified as an MILP [9], [29]. The makespan and the sum of operation times are considered as two objectives. The sum of operation times is maximized to get a lower energy consumption, while the overall makespan is minimized.
- Commercial solver Baron, which is regarded as an efficient solver for MINLPs [32].

To compare the different methods above, multiple case studies (involving various system scales) are considered. The configurations of the case studies are given in Table V. For each configuration, ten experiments are conducted to have a general comparison. In each experiment, the container locations in the vessel and the storage place are randomly generated.

The following key performance indicators are used to evaluate the compared methods: the first one is *makespan* J_{C}, which is the completion time of all tasks; the second one is the *energy consumption* J_{E}, which is the sum of the consumed kinetic energy for completing all the operations of lift AGVs; the last one is the *computation time* required to compute the best-found solution of the studied scheduling problem.

2) Algorithm Settings: For the proposed GAs, the maximum number of iterations is set to 500, and its population is 200. For a fair comparison, we use the maximum fitness evaluations (MaxFEs), which is a common criterion to terminate different metaheuristic algorithms [40]. The same MaxFEs value is used when comparing the GA, VNS, and TS algorithms. The

TABLE VI
COMPUTATIONAL PERFORMANCE USING DIFFERENT METAHEURISTIC METHODS. (UNIT: J_C: SECONDS, J_E: Kwh)

	L-GA		W-GA		L-VNS		W-VNS		L-TS		W-TS	
Case	J_C	J_E	J_C	J_E	J_C	J_E	J_C	J_E	J_C	J_E	J_C	J_E
2QC-1	**490.5±2.3**	**5.51±0.05**	491.4±2.2	5.64±0.10	491.1±2.4	5.88±0.17	495.1±3.4	6.04±0.14	491.3±2.6	5.90±0.22	496±3.3	6.57±0.16
2QC-2	**478.9±2.3**	**4.9±0.10**	479.5±2.1	4.96±0.12	480.6±2.8	5.68±0.13	486.7±3.1	5.96±0.08	480.6±2.0	5.70±0.24	486±4.0	6.17±0.12
2QC-3	**467.4±1.4**	**4.28±0.15**	468.8±1.3	4.37±0.12	469.5±1.6	4.97±0.12	473.1±2.1	5.17±0.12	468.6±1.5	4.92±0.13	473±2.5	5.13±0.13
3QC-1	**496±3.3**	**7.35±0.24**	496±3.3	7.38±0.19	505.5±2.5	7.61±0.10	510.5±4.7	7.59±0.11	504.6±4.8	7.75±0.16	511.6±4.8	7.72±0.11
3QC-2	**484±2.2**	6.33±0.28	484.6±2.0	**6.29±0.22**	493.3±2.9	6.76±0.16	498.3±2.9	6.90±0.15	492.8±2.2	6.62±0.17	499.8±2.2	7.43±0.14
3QC-3	**471.6±1.3**	5.36±0.12	472.2±1.5	**5.31±0.06**	475.5±2.0	6.41±0.18	482.5±2.0	6.63±0.16	478.6±2.0	6.30±0.19	485.6±2.0	6.52±0.14
4QC-1	**496.2±2.9**	8.22±0.14	496.5±3.1	**8.18±0.10**	507.0±3.6	8.66±0.32	512±3.6	8.31±0.21	505.6±4.1	8.45±0.25	510.6±4.1	8.84±0.11
4QC-2	**483.4±2.4**	**7.26±0.19**	484.4±2.2	7.31±0.18	491.0±2.8	7.68±0.25	498±2.8	8.04±0.29	490.6±3.4	7.55±0.25	497.6±3.4	8.54±0.09
4QC-3	**471±1.8**	**6.41±0.12**	471.9±1.6	6.45±0.12	479±0.9	7.36±0.22	482.5±2.0	6.63±0.16	477.6±2.4	7.22±0.26	485.6±2.0	6.52±0.14
5QC-1	**505.9±2.8**	**10.07±0.19**	506±2.7	10.17±0.12	513.7±4.0	10.53±0.28	518.7±4.0	10.27±0.15	514.8±0.1	10.46±0.21	519.8±3.9	10.98±0.24
5QC-2	**492.9±2.4**	**8.89±0.21**	493.5±2.9	8.98±0.19	500.9±3.5	9.55±0.28	507.9±3.5	10.01±0.16	499.7±3.7	9.49±0.19	506.6±3.8	10.65±0.20
5QC-3	**480.9±1.4**	8.0±0.11	482.4±1.2	**7.95±0.11**	491.7±3.7	9.29±0.24	498.7±3.7	9.14±0.22	490.8±2.9	9.18±0.23	497.8±2.9	9.02±0.20
5QC-4	**734.4±3.5**	13.08±0.34	736.5±4.3	**12.93±0.43**	740.8±4.9	13.89±0.35	745.8±4.9	14.28±4.39	741.4±4.4	14.14±0.63	747.4±4.4	13.94±0.39
5QC-5	**719.6±3.3**	**11.32±0.28**	721.7±4.2	11.57±0.31	726.2±4.3	13.62±0.41	731.2±4.3	13.82±0.31	724.1±3.2	13.55±0.28	731.1±3.2	13.54±0.30
5QC-6	**703.3±3.5**	**10.12±0.27**	706.8±3.1	10.42±0.30	710.2±5.7	13.15±0.32	717.2±5.7	13.17±0.36	709.5±4.8	12.60±0.28	716.5±4.8	13.06±0.23
5QC-7	**990±4.4**	**17.41±0.43**	992.8±5.2	17.82±0.46	996±4.6	19.35±0.36	1002±4.6	19.86±0.34	996.1±5.3	19.96±0.36	1002.1±5.3	19.80±0.36
5QC-8	**976.4±3.1**	**16.55±0.25**	979.5±3.7	16.67±0.50	983.3±6.0	19.04±0.41	989.3±6.0	19.46±0.54	981.9±2.7	19.06±0.37	986.9±2.8	19.31±0.52
5QC-9	**960.5±3.4**	**15.48±0.33**	962.5±3.2	15.82±0.62	969.5±4.8	18.38±0.34	975.5±4.8	18.48±0.46	966.3±5.2	18.68±0.33	973.3±5.2	18.58±0.37
5QC-10	**1275.2±3.7**	**22.99±0.42**	1276.8±2.6	23.04±0.59	**1275.2±3.7**	24.19±0.39	1278.2±4.4	25.32±0.30	1275.6±3.7	24.09±0.38	1278.9±4.7	25.17±0.50
5QC-11	**1258±3.6**	21.45±0.35	1259.4±3.6	**20.84±0.35**	1258.5±3.2	23.19±0.39	1262.3±2.5	23.64±0.42	1259.2±3.1	23.21±0.37	1262.8±4.4	23.14±0.62
5QC-12	**1238.1±4.2**	**19.49±0.26**	1239±3.8	20.05±0.33	1242.4±9.0	22.85±0.42	1246.7±9.9	22.68±0.36	1243.3±7.3	22.64±0.41	1249.7±7.9	22.59±0.36
Average	**698.8±286.8**	**11.00±5.62**	702.4±294.0	11.05±5.63	704.8±285.4	12.34±6.27	710.2±285.0	12.47±6.37	704.4±285.5	12.27±6.31	710.4±285.1	12.57±6.20

number of iterations for the VNS is 500 × 200 (there is no population for the VNS). For the TS, the number of candidates is 200, and the maximum number of iterations is 500.

In the weighted-sum formulation of the studied metaheuristic algorithms (GA, VNS, and TS), the parameter λ is set to be 10^{-6} to prioritize the makespan. The maximum computation time for the metaheuristics (GA, VSN, and TS) is set to be 600 seconds. For the lexicographic strategy of the GA, VNS, and TS algorithms, the MaxFEs to individually compute the two objectives are all set as half of the weighted-sum value.

For the two-phase method and the B-MILP method, it is well-known that solving the makespan minimization problem is NP-hard [41], and we thus use a more compact GA, without encoding operation times of AGVs, to solve this single-objective MILP formulation, both for the B-MILP method and the two-phase method. The commercial solver Baron (version 19.3.24) is used for solving the considered MINLP. The maximum computation time of Baron is set to 1 hour, as the solver may not return good quality solutions (if any) in a short computation time.

The hardware for all experiments is an Intel i7-4200 processor (1.6GHz) with 4GB of memory. The optimization problems are modeled and solved in Matlab R2014.

B. Results of GA Against Other Metaheuristics

This part compares the experimental results of the proposed GA with two commonly-used metaheuristics (i.e., VNS and TS). These three metaheuristics are tested on all the instances of Table V, both for the lexicographic and weighted-sum formulations. For the sake of convenience, the lexicographic GA and the weighted-sum GA are abbreviated as L-GA and W-GA. These abbreviations also applied to VNS and TS. The makespan and energy consumption of the proposed GA configurations are compared against the VNS and TS configurations in Table VI. The computation times of these metaheuristics are presented in Fig. 5.

In Table VI, the makespan J_C and the energy consumption J_E are reported. It can be observed that for the three

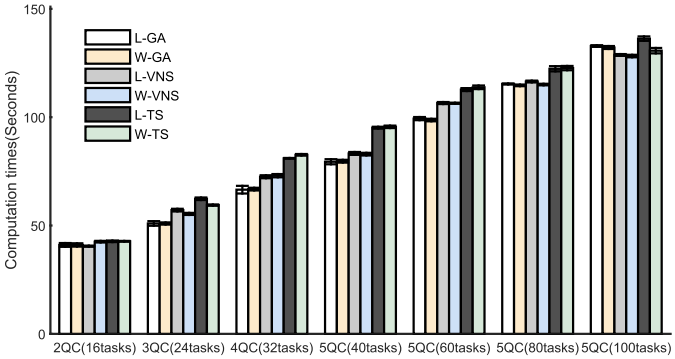

Fig. 5. Computation times of different metaheuristics for solving the formulated MINLP.

metaheuristics, the lexicographic strategy obtains shorter a makespan than the weighted-sum strategy. The studied two objectives both depending on these decision variables (task orders and AGV transport times) are closely correlated, and the weighted-sum strategy is thus less computationally efficient than the lexicographic one. This explains why the lexicographic strategy is better than the weighted-sum one. For either the lexicographic strategy or the weighted-sum strategy, the proposed GA outperforms the algorithms VNS and TS, both on the makespan and the energy consumption. Due to the combination of the designed encoding scheme and the popularity diversity, the proposed GA is more efficient than the tested VNS and TS, both in solving the MIP when minimizing the makespan and in solving the MINLP when minimizing the energy consumption.

Fig. 5 reports the computation times of three metaheuristics both for the lexicographic and weighted-sum formulations. Overall, these computation times are close to each other. The proposed GA and the VNS have slightly shorter computation times than algorithm TS, because TS may take extra steps to deal with updating the tabu list to obtain the best candidate solution.

TABLE VII
COMPUTATIONAL PERFORMANCES OF CASE STUDIES USING START-OF-THE-ART METHODS. (UNIT: J_C: SECONDS, J_E: Kwh)

	L-GA		Two-phase		B-MILP		Baron (1 hour)	
Case	J_C	J_E	J_C	J_E	J_C	J_E	J_C	J_E
2QC-1	490.5±2.3	**5.51±0.05**	490.5±2.3	6.48±0.21	490.5±2.3	7.24±0.36	612.4±88.9	7.61±0.24
2QC-2	478.9±2.3	**4.90±0.10**	478.9±2.3	5.44±0.25	478.9±2.3	6.2±0.42	495.2±29.8	7.27±0.26
2QC-3	467.4±1.4	**4.28±0.15**	467.4±1.4	4.53±0.15	467.4±1.4	5.83±0.40	472.8±4.2	6.68±0.27
3QC-1	496±3.3	**7.35±0.24**	496±3.3	7.92±0.15	496±3.3	8.60±0.28	786.3±72.0	10.04±0.55
3QC-2	484±2.2	**6.33±0.28**	484±2.2	7.06±0.15	484±2.2	7.72±0.22	697.6±54.5	9.54±0.32
3QC-3	471.6±1.3	**5.36±0.12**	471.6±1.3	5.95±0.19	471.6±1.3	7.06±0.23	584.4±52.4	7.60±0.28
4QC-1	496.2±2.9	**8.22±0.14**	496.2±2.9	9.30±0.24	496.2±2.9	10.17±0.37	1640.8±1058.0	11.48±0.44
4QC-2	483.4±2.4	**7.26±0.19**	483.4±2.4	8.21±0.30	483.4±2.4	8.84±0.34	1246.1±861.9	10.26±0.47
4QC-3	471±1.8	**6.41±0.12**	471±1.8	7.73±0.24	471±1.8	8.21±0.13	775.4±94.3	8.98±0.20
5QC-1	505.9±2.8	**10.07±0.19**	505.9±2.8	11.20±0.28	505.9±2.8	12.36±0.16	N.S	N.S
5QC-2	492.9±2.4	**8.89±0.21**	492.9±2.4	10.12±0.26	492.9±2.4	11.34±0.31	N.S	N.S
5QC-3	480.9±1.4	**8.00±0.11**	480.9±1.4	9.27±0.54	480.9±1.4	10.34±0.34	1120.6±197.9	11.61±0.76
5QC-4	734.4±3.5	**13.08±0.34**	734.4±3.5	14.78±0.43	734.4±3.5	18.07±0.42	N.S	N.S
5QC-5	719.6±3.3	**11.32±0.28**	719.6±3.3	13.83±0.50	719.6±3.3	16.42±0.32	N.S	N.S
5QC-6	703.3±3.5	**10.12±0.27**	703.3±3.5	13.02±0.60	703.3±3.5	15.60±0.40	N.S	N.S
5QC-7	990±4.4	**17.41±0.43**	990±4.4	20.63±0.62	990±4.4	24.79±0.66	N.S	N.S
5QC-8	976.4±3.1	**16.55±0.25**	976.4±3.1	19.57±0.45	976.4±3.1	23.79±0.48	N.S	N.S
5QC-9	960.5±3.4	**15.48±0.33**	960.5±3.4	18.15±1.39	960.5±3.4	22.24±1.63	N.S	N.S
5QC-10	1275.2±3.7	**22.99±0.42**	1275.2±3.7	25.53±0.53	1275.2±3.7	29.55±0.37	N.S	N.S
5QC-11	1258±3.6	**21.45±0.35**	1258±3.6	23.43±0.31	1258±3.6	28.66±0.31	N.S	N.S
5QC-12	1238.1±4.2	**19.49±0.26**	1238.1±4.2	22.48±0.27	1238.1±4.2	27.67±0.36	N.S	N.S
Average	698.8±286.8	**11.00±5.62**	698.8±286.8	12.62±6.41	698.8±286.8	14.82±7.97	N.S	N.S

In summary, from the obtained results, the proposed L-GA can be considered as the most promising metaheuristic, since it achieves the shortest makespan and reduces energy consumption considerably better than VNS and TS. In the next part, the proposed L-GA is compared with three state-of-the-art methods to further investigate its potential to reduce energy consumption when scheduling AGVs in container terminals.

C. Results of GA Against Other Methods

The proposed L-GA method is now compared against three state-of-the-art methods (the two-phase method, the B-MILP method, and the MINLP solver Baron) that can be used for computing energy-efficient terminal operations. The comparisons are performed for all the instances of Table V. The operational performance of these methods is reported in Table VII.

In general, the best solution provided by the solver Baron is computed in 1 hour, but the solution quality is the worst among the other approaches in Table VII. Within the given maximum computation time, Baron cannot provide solutions for all the instances (solutions cannot be obtained for the cases of 5QC), while the other listed methods return solutions for each instance. For the cases in which Baron always returns feasible schedules, the corresponding makespan is the longest and the energy consumption is the highest compared with the other methods. As a result, we will focus on discussing the results obtained by the proposed L-GA algorithm, the two-phase method, and the B-MILP method.

In Table VII, the L-GA, the two-phase method, and the B-MILP method provide energy-efficient solutions for all the instances. The makespan computed by these approaches is identical, because they use the same GA as the algorithm core. However, the L-GA reports a better energy minimization performance, than the two-phase and B-MILP methods, for each instance. On average, the proposed L-GA reduces energy consumption by 13% and by 26% against the two-phase method and the B-MILP method, respectively. Compared to the B-MILP method, the proposed L-GA adjusts the task orders and the AGV's speed properly following a detail energy consumption representation and thus saves more energy. The two-phase method separates optimizing the task order and optimizing the AGV speed and, therefore, the energy reduction is less than the proposed L-GA. In general, we conclude that optimizing task orders and processing times simultaneously might result in a considerable energy reduction compared to optimizing them separately.

From the results of Table VII, we highlight the influence of the number of vehicles on the performance indices J_C and J_E. For the same setting of QCs and containers, when the number of AGVs increases, the makespan decreases, since more AGVs reduce the waiting time of the QCs to accelerate the container exchanges between the QCs and the AGVs. When the number of AGVs grows, the energy consumption of AGVs decreases, because each AGV becomes less busy. Each AGV can thus reduce energy by lowering its vehicle speed. These features are common for each method in Table VII.

Now, we compare the consumed energy for loading and unloading containers by the lift AGVs of three energy-efficient methods in Fig. 6. J_E^{in} and J_E^{out} represent energy consumption for transporting inbound and outbound containers, respectively. It is observed from Fig. 6 that, when moving the inbound and outbound containers, the proposed L-GA consumes less energy than the B-MILP method, because the L-GA method considers a significant more detailed model than the B-MILP method. The advantage of the L-GA method against the two-phase method for outbound containers cannot be seen for small-scale cases, because each outbound task has limited destination choices. Regarding the larger-scale cases, the L-GA method slightly outperforms the two-phase method.

Fig. 7 compares the average time for computing their best-found solutions of the proposed L-GA in comparison to two-phase and B-MILP methods. The proposed L-GA method and the B-MILP method all return their best solutions within a reasonable computational time. When the instance scale increases (for larger cases of 5QC), the two-phase method requires a considerably longer computation time, because solving the NLP of the second phase is time-consuming.

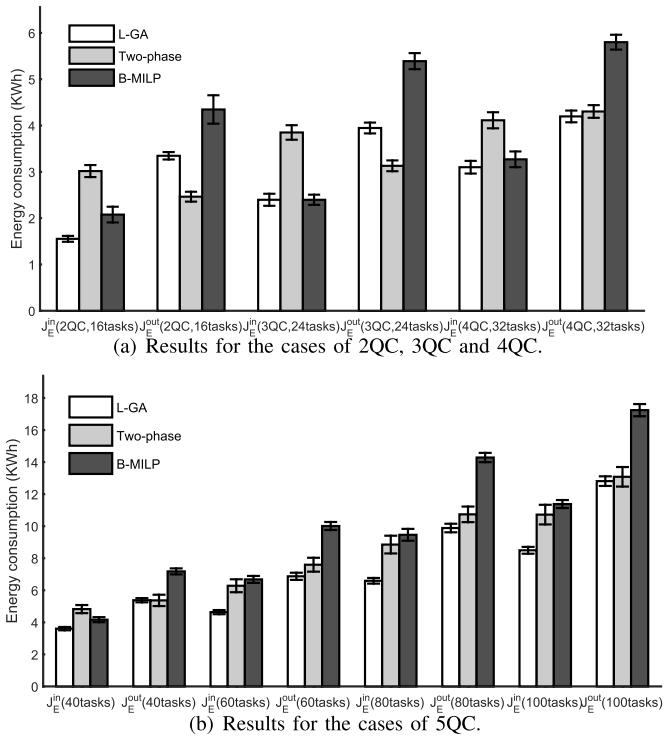

Fig. 6. Averaged energy consumption of lift AGVs for loading and unloading containers by adopting the proposed methods.

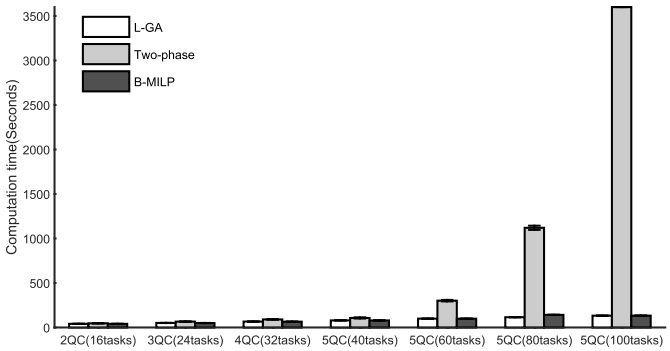

Fig. 7. Averaged computation times of the proposed GAs against the two-phase and the B-MILP methods.

D. Analysis of Pareto Frontier

This part presents the Pareto frontier between minimization of the makespan and the energy consumption for the studied container handling system. This is obtained by using the ϵ-constraint method in Algorithm 2. Fig. 8 shows one example of the Pareto frontier for the instance 3QC-3. The computed Pareto solution for this example is reported in Table VIII.

It can be clearly seen in Fig. 8 that, when obtaining the makespan (471 seconds), the energy consumption of the lift AGVs can be further reduced from about 6.25 KWh to about 5.30 KWh. This means that it is possible to further minimize energy consumption while keeping the shortest possible makespan. Fig. 8 also shows that, when the energy consumption is lower than its minimum for the shortest makespan, the makespan increases accordingly.

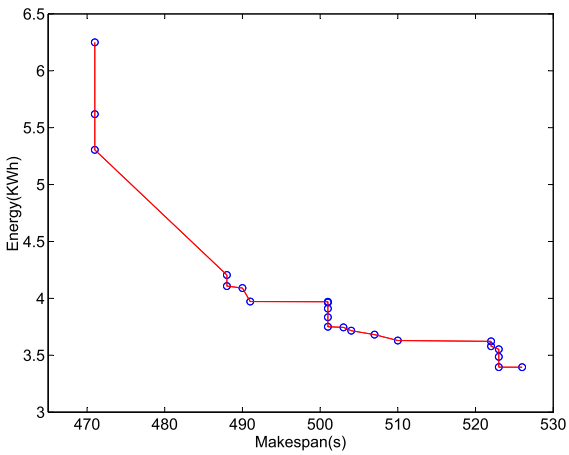

Fig. 8. Pareto frontier using the developed Algorithm 2 of one experiment from Case 3QC-3.

TABLE VIII
COMPUTED PARETO SOLUTIONS FOR SCENARIO 5

Iteration	Computational times (Seconds)	J_C (Seconds)	J_E (*Kwh*)
1	51.63	471	6.25
2	51.85	471	5.62
3	50.25	471	5.30
4	51.83	488	4.21
5	51.26	488	4.11
6	50.20	490	4.09
7	50.56	491	3.97
8	51.09	501	3.97
9	51.92	501	3.96
10	51.93	501	3.91
11	50.32	501	3.83
12	51.94	501	3.75
13	51.91	503	3.75
14	50.97	504	3.72
15	51.60	507	3.68
16	50.28	510	3.63
17	50.84	522	3.62
18	51.83	522	3.58
19	51.58	523	3.55
20	51.92	523	3.49
21	51.31	523	3.40
22	50.07	526	3.39

Table VIII reports that the ϵ-constraint method identifies 12 non-dominated solutions (at iterations 3, 5, 6, 7, 12, 13, 14, 15, 16, 18, 21 and 22). For the minimal makespan ($J_C = 471$), energy consumption can be reduced considerably, compared with the case when J_E is not optimized. The solution at iteration 3 becomes a turning point of the Pareto frontier regarding the shortest makespan. As Algorithm 2 is based on the developed GA, the computation time at different iterations is quite close to each other, considering that at each iteration a single objective optimization problem is solved by the same GA configuration.

VI. CONCLUSION AND FUTURE RESEARCH

This paper investigates a new scheduling problem in automated container terminals, aimed at minimizing energy consumption while maintaining a competitive makespan. In the scheduling problem, both the task sequences and the operation times are optimized, while a bi-objective optimization problem is formulated based on a flow shop representation. Due to the

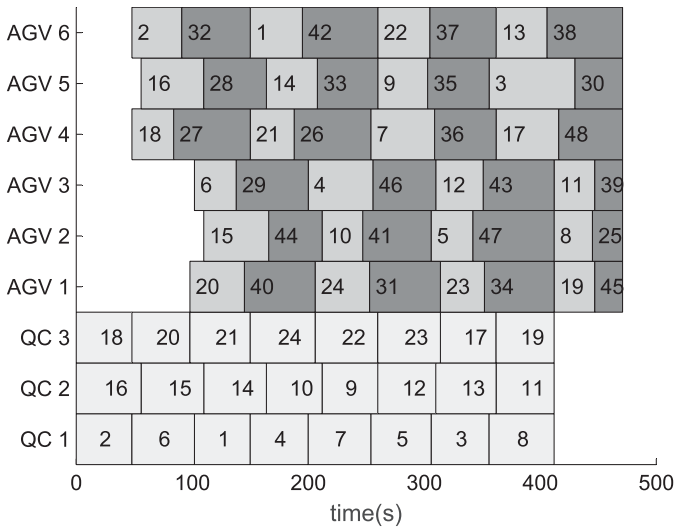

Fig. 9. Gantt chart of 3QC-1 determined by the proposed L-GA for solving the MINLP.

nonlinearity of the energy objective, the formulated problem is a non-convex mixed-integer nonlinear programming (MINLP), which suffers from computational intractability.

To address the computational challenge in solving the MINLP, a dedicated and customized genetic algorithm is developed considering the characteristics of the considered problem. The lexicographic and weighted-sum strategies are proposed to minimize energy consumption while keeping high handing capacity. The two strategies are tested in a container benchmark system, in comparison to a commercial MINLP solver and two state-of-the-art methods. The numerical experiment results show that the MINLP solver cannot provide a feasible schedule for the majority of the investigated case studies, whereas the developed lexicographic GA can quickly provide good quality solutions. For either the lexicographic strategy or the weighted-sum strategy, the proposed GA outperforms the VNS and TS, in terms of both minimizing the makespan and energy consumption. Specifically, at least 10% energy is saved without increasing the makespan. The proposed GA also reduces the energy consumption by 13% and by 26%, on average, compared with the two-phase method and the B-MILP method without deteriorating the makespan.

The considerable energy reduction using the proposed methods concludes that simultaneously optimizing task orders and processing times might result in a significantly reduced energy consumption than optimizing them separately. As a result, logistics operations managers can have more energy-efficient operations to satisfy the environmental requirement and further reduce the operation costs. The energy consumption reduction in container terminals requires closer and more detailed coordination between different types of machines by considering the difference among the tasks to be processed. To achieve a high handling capacity using much less energy consumption, container terminals need to be automatized using more advanced and more flexible integration methods. Different types of machines should be coordinated, and task orders and operation times should be optimized simultaneously based on fully used cargo information.

Further research should consider the coordination of operations with multiple vessels for a large-scale terminal. Meanwhile, lower bound methods for the considered MINLP problem will be investigated in our future works.

APPENDIX
ILLUSTRATIVE EXAMPLE

Fig. 9 gives a Gantt chart of 3QC-3 using the L-GA for solving the MINLP. In this case, 24 tasks (24 inbound containers 1-24 and 24 outbound containers 25-48) are processed. The numbers shown in the QC schedule correspond to each task. For the AGV schedules, the number in the light gray grid corresponds to the inbound container and the number in the dark gray gird matches the outbound container.

REFERENCES

[1] S. N. Sirimanne *et al.*, *Review of Maritime Transport 2019*. New York, NY, USA: United Nations, 2019.
[2] "Container intelligence quarterly. Second quarter," Clarksons Res., London, U.K., Tech. Rep., 2019.
[3] C. Caballini, S. Fioribello, S. Sacone, and S. Siri, "An MILP optimization problem for sizing port rail networks and planning shunting operations in container terminals," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 4, pp. 1492–1503, Oct. 2016.
[4] J. P. Rodrigue, Ed., *The Geography of Transport Systems*, 5th ed. New York, NY, USA: Routledge, 2020.
[5] B. Dragovic, Z. Yang, and S. Papadimitriou, "Guest editorial," *Maritime Bus. Rev.*, vol. 1, no. 5, pp. 2–4, 2020.
[6] T. Spengler and G. Wilmsmeier, "Sustainable performance and benchmarking in container terminals—The energy dimension," in *Green Ports*. Amsterdam, The Netherlands: Elsevier, 2019, pp. 125–154.
[7] Ç. Iris and J. S. L. Lam, "A review of energy efficiency in ports: Operational strategies, technologies and energy management systems," *Renew. Sustain. Energy Rev.*, vol. 112, pp. 170–182, Sep. 2019.
[8] D. Yu, D. Li, M. Sha, and D. Zhang, "Carbon-efficient deployment of electric rubber-tyred gantry cranes in container terminals with workload uncertainty," *Eur. J. Oper. Res.*, vol. 275, no. 2, pp. 552–569, Jun. 2019.
[9] J. Xin, R. R. Negenborn, and G. Lodewijks, "Energy-aware control for automated container terminals using integrated flow shop scheduling and optimal control," *Transp. Res. C, Emerg. Technol.*, vol. 44, pp. 214–230, Jul. 2014.
[10] C. Caballini, C. Pasquale, S. Sacone, and S. Siri, "An event-triggered receding-horizon scheme for planning rail operations in maritime terminals," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 365–375, Feb. 2014.
[11] X. Yang, W. Mi, X. Li, G. An, N. Zhao, and C. Mi, "A simulation study on the design of a novel automated container terminal," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2889–2899, Oct. 2015.
[12] B. Dragović, E. Tzannatos, and N. K. Park, "Simulation modelling in ports and container terminals: Literature overview and analysis by research field, application area and tool," *Flexible Services Manuf. J.*, vol. 29, no. 1, pp. 4–34, Mar. 2017.
[13] H. L. Ma, S. H. Chung, H. K. Chan, and L. Cui, "An integrated model for berth and yard planning in container terminals with multi-continuous berth layout," *Ann. Oper. Res.*, vol. 273, nos. 1–2, pp. 409–431, Feb. 2019.
[14] A. Alessandri, C. Cervellera, M. Cuneo, M. Gaggero, and G. Soncin, "Modeling and feedback control for resource allocation and performance analysis in container terminals," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 4, pp. 601–614, Dec. 2008.
[15] K. Shintani, R. Konings, and A. Imai, "Combinable containers: A container innovation to save container fleet and empty container repositioning costs," *Transp. Res. E, Logistics Transp. Rev.*, vol. 130, pp. 248–272, Oct. 2019.
[16] R. T. Cahyono, E. J. Flonk, and B. Jayawardhana, "Discrete-event systems modeling and the model predictive allocation algorithm for integrated berth and quay crane allocation," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1321–1331, Mar. 2020.
[17] F. Zheng, X. Man, F. Chu, M. Liu, and C. Chu, "Two yard crane scheduling with dynamic processing time and interference," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3775–3784, Dec. 2018.

[18] K. H. Kim and J. W. Bae, "A look-ahead dispatching method for automated guided vehicles in automated port container terminals," *Transp. Sci.*, vol. 38, no. 2, pp. 224–234, May 2004.
[19] J. Schmidt, C. Meyer-Barlag, M. Eisel, L. M. Kolbe, and H.-J. Appelrath, "Using battery-electric AGVs in container terminals—Assessing the potential and optimizing the economic viability," *Res. Transp. Bus. Manage.*, vol. 17, pp. 99–111, Dec. 2015.
[20] J. Luo and Y. Wu, "Scheduling of container-handling equipment during the loading process at an automated container terminal," *Comput. Ind. Eng.*, vol. 149, Nov. 2020, Art. no. 106848.
[21] Y. Yang, M. Zhong, Y. Dessouky, and O. Postolache, "An integrated scheduling method for AGV routing in automated container terminals," *Comput. Ind. Eng.*, vol. 126, pp. 482–493, Dec. 2018.
[22] T. Qin, Y. Du, J. H. Chen, and M. Sha, "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.*, vol. 285, no. 3, pp. 884–901, Sep. 2020.
[23] Gottwald. (2018). *Gottwald Lift AGV*. Accessed: Jun. 1, 2018. [Online]. Available: http://www.konecranes.com
[24] D. Kress, S. Meiswinkel, and E. Pesch, "Straddle carrier routing at seaport container terminals in the presence of short term quay crane buffer areas," *Eur. J. Oper. Res.*, vol. 279, no. 3, pp. 732–750, Dec. 2019.
[25] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Closed-loop scheduling and control of waterborne AGVs for energy-efficient inter terminal transport," *Transp. Res. E, Logistics Transp. Rev.*, vol. 105, pp. 261–278, Sep. 2017.
[26] J. He, Y. Huang, W. Yan, and S. Wang, "Integrated internal truck, yard crane and quay crane scheduling in a container terminal considering energy consumption," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2464–2487, Apr. 2015.
[27] J. He, Y. Huang, and W. Yan, "Yard crane scheduling in a container terminal for the trade-off between efficiency and energy consumption," *Adv. Eng. Informat.*, vol. 29, no. 1, pp. 59–75, Jan. 2015.
[28] M. Sha *et al.*, "Scheduling optimization of yard cranes with minimal energy consumption at container terminals," *Comput. Ind. Eng.*, vol. 113, pp. 704–713, Nov. 2017.
[29] J. Xin, R. R. Negenborn, and G. Lodewijks, "Event-driven receding horizon control for energy-efficient container handling," *Control Eng. Pract.*, vol. 39, pp. 45–55, Jun. 2015.
[30] T. Jonker, M. Duinkerken, N. Yorke-Smith, A. de Waal, and R. Negenborn, "Coordinated optimization of equipment operations in a container terminal," *Flexible Services Manuf. J.*, vol. 33, pp. 281–311, Jun. 2021.
[31] S. Burer and A. N. Letchford, "Non-convex mixed-integer nonlinear programming: A survey," *Surveys Oper. Res. Manage. Sci.*, vol. 17, no. 2, pp. 97–106, Jul. 2012.
[32] N. V. Sahinidis. (2017). *BARON 17.8.9: Global Optimization of Mixed-Integer Nonlinear Programs, User's Manual*. [Online]. Available: http://archimedes.cheme.cmu.edu
[33] C.-T. Young, Y. Zheng, C.-W. Yeh, and S.-S. Jang, "Information-guided genetic algorithm approach to the solution of MINLP problems," *Ind. Eng. Chem. Res.*, vol. 46, no. 5, pp. 1527–1537, Feb. 2007.
[34] A. H. Halim and I. Ismail, "Combinatorial optimization: Comparison of heuristic algorithms in travelling salesman problem," *Arch. Comput. Methods Eng.*, vol. 26, no. 2, pp. 367–380, Apr. 2019.
[35] A. D'Ariano, L. Meng, G. Centulio, and F. Corman, "Integrated stochastic optimization approaches for tactical scheduling of trains and railway infrastructure maintenance," *Comput. Ind. Eng.*, vol. 127, pp. 1315–1335, Jan. 2019.
[36] J. Xin, R. R. Negenborn, F. Corman, and G. Lodewijks, "Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 377–396, Nov. 2015.
[37] P. Hansen, N. Mladenović, R. Todosijević, and S. Hanafi, "Variable neighborhood search: Basics and variants," *EURO J. Comput. Optim.*, vol. 5, no. 3, pp. 423–454, Sep. 2017.
[38] F. Glover and E. Taillard, "A user's guide to tabu search," *Ann. Oper. Res.*, vol. 41, no. 1, pp. 1–28, 1993.
[39] Y. Wang, A. D'Ariano, J. Yin, L. Meng, T. Tang, and B. Ning, "Passenger demand oriented train scheduling and rolling stock circulation planning for an urban rail transit line," *Transp. Res. B, Methodol.*, vol. 118, pp. 193–227, Dec. 2018.
[40] T. Huang, Y.-J. Gong, S. Kwong, H. Wang, and J. Zhang, "A niching memetic algorithm for multi-solution traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 508–522, Jun. 2020.
[41] L. Chen, A. Langevin, and Z. Lu, "Integrated scheduling of crane handling and truck transportation in a maritime container terminal," *Eur. J. Oper. Res.*, vol. 225, no. 1, pp. 142–152, Feb. 2013.

Jianbin Xin (Member, IEEE) received the B.Sc. degree in electrical engineering from Xidian University, China, in 2007, the M.Sc. degree in control science and engineering from Xi'an Jiaotong University, China, in 2010, and the Ph.D. degree from the Department of Maritime and Transport Technology, Delft University of Technology, The Netherlands, in 2015.

He is currently an Associate Professor with the Department of Automation, Zhengzhou University, China. His research interests include modeling and control of smart logistics systems and hybrid systems control.

Chuang Meng received the bachelor's degree in electrical engineering from the Luoyang Institute of Science and Technology in 2018. He is currently pursuing the master's degree with the School of Electrical Engineering, Zhengzhou University. His research interest includes planning of multiple robots for manufacturing and logistics.

Andrea D'Ariano received the B.S. and M.S. degrees in computer science and automation engineering from Roma Tre University, and the Ph.D. degree from the Department of Transport and Planning, Delft University of Technology, under the supervision of Prof. I. A. Hansen. He is currently an Associate Professor with the Department of Engineering, Roma Tre University. His main research interest includes study of scheduling problems with application to public transportation and logistics. He is also an Associate Editor of well-known international journals, such as *Transportation Research Part B: Methodological* and conferences, such as IEEE ITSC.

Dongshu Wang received the bachelor's degree in mechanical manufacture technique and equipment, the master's degree in mechanical manufacture and automation, and the Ph.D. degree in control theory and control engineering from Northeastern University, China, in 1996, 2002, and 2006, respectively.

He is currently an Associate Professor with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China. His research interests include autonomous mental development and artificial intelligence.

Rudy R. Negenborn received the M.Sc. degree in computer science from Utrecht University in 2003, and the Ph.D. degree from the Delft Center for Systems and Control, Delft University of Technology, in 2007.

He is currently a Full Professor "Multi-Machine Operations and Logistics" and the Head of the Section "Transport Engineering and Logistics" at the Delft University of Technology. His research interests include distributed control, multi-agent systems, model predictive control, and optimization.