INTERNATIONAL JOURNAL OF MARITIME TECHNOLOGY

IJMT Vol.16/ Summer 2021 (13-27)

Available online at: www. ijmt.ir

Integrated Management of Equipment in Automated Container Terminals

Hassan Rashidi[1*], Fatemeh Habibi Ehsaee[2], FereshtehAzadi Parand

[1] *Professor, Faculty of Statistics, Mathematics and Computer Science, Allameh Tabataba'i University; hrashi@atu.ac.ir*
[2] *MSc in Computer Science, Faculty of Statistics, Mathematics and Computer Science, Allameh Tabataba'i University; f.habibi.ehsaparandni@gmail.com*
[3] *Assistant Professor, Faculty of Statistics, Mathematics and Computer Science, Allameh Tabataba'i University; f.a.parand@gmail.com*

ARTICLE INFO

Article History:
Received: 09 Sep. 2021
Accepted: 08 Feb. 2022

Keywords:
Automated Container Terminal, Automated Guided Vehicle, Quay Crane, Yard Crane

ABSTRACT

The efficiency of ports and container terminals is strongly related to the process of loading containers onto and unloading containers from the docked ships. In this research, an issue of integrated equipment management in automated container terminals with the aim of increasing efficiency has been studied. Due to this issue falls into NP-Hard problems, it was divided into two sub-problems: Allocating resources to containers and arranging the containers serviced by automated guided vehicles. Both sub-problems were formulated and expressed using the linear integer-programming model. The first sub-problem is solved by the allocation of random process resources with uniform distribution and the second part is solved using a Sorting Genetic Algorithm. The main parameters of the proposed solution methods were determined with Minitab software and Taguchi techniques. In order to evaluate the efficiency and effectiveness of the proposed solution methods, many numerical experiments have been examined and evaluated. The experimental results show that the proposed solutions are efficient for estimating the service time and the number of automated guided vehicles required to transporting the containers in the container ports.

1. Introduction

International trade and exchanging commodities between countries are increasing and are continued exponentially in the current age. Depending on their needs, each country imports the commodities it needs and exports its surplus to other countries. The automated container terminals (ACT) are developed to transport commodities by cargo ships on which loading containers or unloading containers from them. The main functions of these terminals are delivering containers to consignees and receiving containers from shippers, loading containers onto and unloading containers from ships, and storing containers temporarily to account either for the efficiency of the equipment or for the difference in arrival times of the sea and land carriers. Containers are usually handled in two important compartments. Figure 1 shows the layout of the automated container terminal with two main compartments. The first compartment is called the quay-side and the second one is the yard-side. Between the yard-side and quay-side, the automated guided vehicles transport the containers. Anchoring of ships in an ACT spends high costs because of the expensive equipment used loading, unloading, and transporting the containers as well.

Figure 2 shows the loading and unloading process of containers. In the quay-side, there are a number of berths where the ships are docked for loading and unloading operations by the quay cranes. Only one ship is allowed to dock at the same time in a berth. Each quay crane has three important parts: the main trolley, the transfer platform, and the portal trolley. The main trolley is responsible for getting the container from the ship and put it on the platform or vice versa. The transfer platform is responsible for maintaining the container. The trolley is responsible for getting the container from the transfer platform and put it onto the automated guided vehicle or vice versa. These trucks can move just one container in each operation.

In the yard-side, there are many storage locations and yard cranes. Each storage location has separate parts called blocks. Each block contains two short-term and long-term storage locations. Each storage area has a yard crane. For example, the front crane is in short-term storage and the back crane is in long-term storage.

The compartment between the beach and the yard is the place where automated guided vehicles (AGV) move. This middle part of the terminal is called the transmission location. This section encompasses rails and automated guided vehicles. The AGVs move in the rails. They are unmanned vehicles that are responsible for transporting and moving containers from the quay-side to the yard-side or vice versa. Each AGV is capable of carrying one container at the same time.

The process of loading inbound containers is transmitting containers from the ship to the storage area. At first cargo ship docked at the berth for doing operations. Several quay cranes start working on the ship. Then, on the shore, the main trolley of each crane picks the container from the ship and puts it on the transfer platform. The portal trolley picks the container from the transfer platform and puts it to the AGV. Trucks carry the container from the quay-side to the yard-side. On the yard-side, the front yard crane picks up the container from the AGV and places it in short-term storage. Then the backyard crane removes the container from the short-term storage area and places it in the long-term storage area.

The process of loading the outbound containers is transmitting containers from the storage place onto the ship. At first, the backyard crane put the container from the long-term storage place to the short-term storage place. Then, the front yard crane picks up the container from the short-term storage area and delivers it to the AGV. The AGV transfer the container from the yard-side to the quay-side. On the quay-side, the portal trolley put the container from the AGV and places it on the transfer platform. Then the main trolley in the crane lifts the container dock from the moving platform and places it on the ship. The main trolley is responsible for get the container from the ship and put it on the platform or vice versa. The transfer platform is responsible for maintaining the container. The trolley is responsible for getting the container from the transfer platform and put it onto the automated guided vehicle or vice versa. These trucks can move just one container in each operation.

In the yard-side, there are a number of storage locations and yard cranes. Each storage location has separate parts called blocks. Each block contains two short-term and long-term storage locations. Each storage area has a yard crane. For example, the front crane is in short-term storage and the back crane is in long-term storage.

The compartment between the beach and the yard is the place where the AGVs move. This middle part of the terminal is called the transmission location. This section includes rails and automated guided vehicles. The path of AGVs is determined using rails. They are unmanned vehicles that are responsible for transporting and moving containers from the quay-side to the yard-side or vice versa. Each AGV is capable of carrying one container at the same time.

The process of loading inbound containers is transmitting containers from the ship to the storage area. At first cargo ship docked at the berth for doing operations. A number of quay cranes start working on the ship. Then, on the shore, the main trolley of each crane picks the container from the ship and puts it on the transfer platform. The portal trolley picks the container from the transfer platform and puts it to the AGV. Trucks carry the container from the quay-side to the yard-side. On the yard-side, the front yard crane picks up the container from the AGV and places it in short-term storage. Then the backyard crane removes the container from the short-term storage area and places it in the long-term storage area.

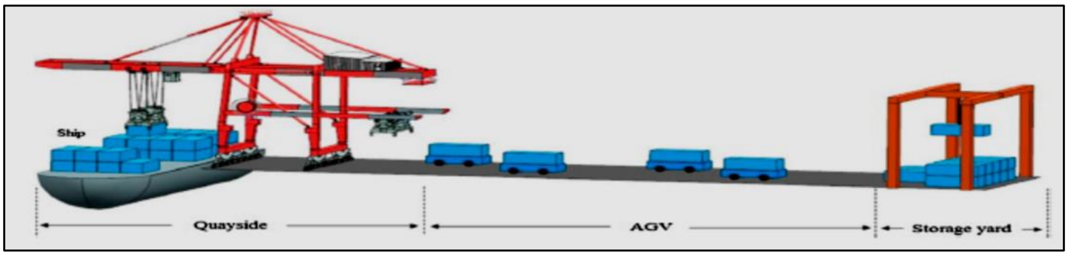

Figure 1. Layout of the automated container terminal (adopte from [1].)

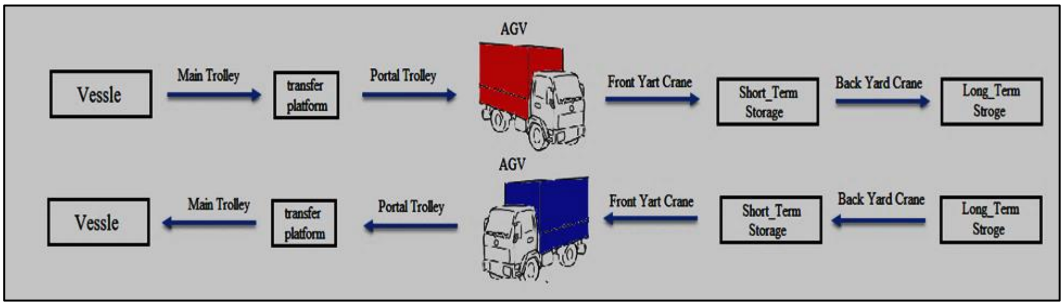

Figure 2. Loading and Unloading Process (adopted from[1]).

The process of loading the outbound containers is transmitting containers from the storage place onto the ship. At first, the backyard crane put the container from the long-term storage place to the short-term storage place. Then, the front yard crane picks up the container from the short-term storage area and delivers it to the AGV. The AGV transfer the container from the yard-side to the quay-side. On the quay-side, the portal trolley put the container from the AGV and places it on the transfer platform. Then the main trolley in the crane lifts the container dock from the moving platform and places it on the ship.

The efficiency of each ACT depends on the time each docked ship spent on the quay-side for service. To increasing this efficiency, the speed of sending the import containers from the docked ships to the yard-side or from the yard-side to the docked ships in the terminal must be improved. The study presents an integrated planning for the equipment available in the container terminals with the aim of reducing the service time of ships. The rest of this paper is organized as follows. Section 2 presents the related works and reviews the latest researches devoted to the container terminals. Section 3 formulated the problem. Section 4 proposes the solution methods. Section 5 makes the numerical experiments to evaluate the efficiency of the solution methods, and finally Section 6 provides the summary and conclusions.

2. Related works

In this section, we review the latest research devoted to automated container terminals. Steenken et al. (2004) examined the issue of Quay planning and resource optimization in automated container terminals [2[. They provided a network queue model for logical operations related to the process of arriving, docked, and leaving ships at container terminals. To solve the problem, they used the "what if" optimization approach for the dock planning problem.

Chen et al. (2013) studied an interaction between crane handling and truck transportation problems in a container terminal by addressing them simultaneously [3]. Internal trucks are shared among different ships, which helps to reduce empty truck trips in the terminal area. The problem was formulated as a constraint programming model and a three-stage algorithm was developed. At the first stage, crane schedules were generated by a heuristic method. At the second stage, the multiple-truck routing problem was solved based on the precedence relations of the transportation tasks derived from the first stage. At the last stage a complete solution was developed by using a disjunctive graph. The three procedures are connected by an iterative structure, which facilitates the search for a good solution. The experimental results indicated that the three-stage algorithm is effective for finding high-quality solutions and can efficiently solve large-size problems.

Tang et al. (2014) considered the coordination of the two types of equipment to reduce their idle time between performing two successive tasks, addressing the joint quay crane and truck scheduling problem at a container terminal [4]. For the one-way flow problem with only inbound containers, in which trucks go back to quayside without carrying outbound containers, a mixed-integer linear programming model was formulated to minimize the makespan. Several valid inequalities and a property of the optimal solutions for the problem were derived, and two lower bounds were obtained. Then, an improved Particle Swarm Optimization (PSO) algorithm was developed to solve this problem, in which a new velocity updating strategy is considered to improve the quality of the solutions. For small-sized problems, this research compared the solutions of the proposed PSO with those of solutions obtained by the CPLEX software. The solutions of the proposed PSO for large-sized problems were compared to the two lower bounds because CPLEX could not solve the problem optimally in a reasonable time. For the more general situation considering both inbound and outbound containers, internal trucks may go back to quay-side with outbound containers. The model was extended to handle this problem with two-way flow. The experiment showed that the improved PSO is efficient to solve the joint scheduling problem of quay cranes and trucks.

He et al. (2015) addressed the problem of integrated quay cranes (QC) scheduling, Internal Truck (IT) scheduling, and yard cranes (YC) scheduling[5]. Firstly, this problem is formulated as a mixed integer programming model (MIP), in which the objective is to minimize the total leaving delay of all vessels and the total transportation energy consumption of all tasks. Furthermore, an integrated simulation-based optimization method is developed for solving the problem, where the simulation is designed for evaluation and optimization algorithm is designed for searching solution space. The optimization algorithm integrates the genetic algorithm (GA) and particle swarm optimization (PSO) algorithm, where the GA is used for global search and the PSO is used for local search. Finally, numerical experiments are conducted to verify the effectiveness of the proposed method. The results show that the proposed method can coordinate the scheduling of the three types of handling equipment and can realize the optimal trade-off between time-saving and energy-saving.

Roy and Koster (2018) developed a new integrated stochastic model for analyzing the performance of overlapping loading and unloading operations that capture the complex stochastic interactions among quayside, vehicle, and stack-side processes[6]. This research used a network of open and semi-open queues to make an analytical model. The model was solved using an iterative algorithm based on the parametric decomposition approximation approach. The system

[DOR: 20.1001.1.23456000.2021.16.0.6.1]

performance is tested at varying container traffic levels. This research found that the percent absolute errors in throughput times compared to simulation are less than 10% for all cases. The model was used to generate design insights and also rapidly analyze what-if scenarios. For example, this research showed that the best yard layout configurations for single (either loading or unloading) operations and the best for overlapping (both loading and unloading) operations largely overlap. The best configurations have relatively few stack blocks and many rows per block. The model is generic and amenable to obtain other design and operational performance insights.

Yang et al. (2018) proposed an integrated scheduling method for routing AGVs at container terminals[7]. In this case, the goal was to reduce the duration of the ship's deployment and the process of loading or unloading containers. They formulated the problem with an integer linear programming model and proposed a two-level genetic algorithm to solve it.

Vahdani et al. (2019) studied a combination of the assignment of quay cranes at container terminals and internal truck sharing assignment among them [8]. For this purpose, a bi-objective optimization model was developed. In the proposed model, several assignment phases, including the assignments of the vessel to container terminals, cranes to terminals, cranes to vessels, and trucks to cranes were performed. The model also aimed to increase and improve the efficiency and effectiveness of internal trucks by sharing them among different terminals, so that there was an appropriate balance between the volume of workloads of the terminals and the trucks in question. The first objective function in the proposed model was to minimize operational costs and the second objective function was to minimize the maximum overflowed workload in the container terminals. Furthermore, in order to solve the proposed model, two meta-heuristic multi-objective algorithms, including modified non-dominated sorting genetic algorithm-II (MNSGA-II) and modified multi-objective particle swarm optimization (MMOPSO) were presented. Several numerical examples have been investigated and analyzed to show the accuracy of the proposed model and the methods. In addition, the results demonstrated that the simultaneous consideration of the assignments and the sharing of trucks would reduce the remaining workload in the container terminals.

Zhao et al. (2019) developed a collaborative scheduling model for automated quayside cranes (AQCs) and AGVs [9]. In the model, the capacity limitation of the transfer platform on AQCs was considered. The minimum total energy consumption of (AQCs) and Automatic Guided Vehicles (AGVs) was taken as the objective function. A two-stage taboo search algorithm was adopted to solve the problem of collaborative scheduling optimization. This algorithm integrated AQC scheduling and AGV scheduling. The optimal solution to the model was obtained by feedback from the two-stage taboo search process. Finally, the Qingdao Port was taken as an example of a data experiment. Ten small-size test cases were solved to evaluate the performance of the proposed optimization methods. The results showed the applicability of the two-stage taboo search algorithm since it can find near-optimal solutions, precisely and accurately.

Castilla et al. (2020) developed an intelligent system that integrates Artificial Intelligence techniques and simulation tools to aid managers in container terminals [10]. The system combines an intelligent evolutionary algorithm to generate high-quality schedules for the cranes with a simulation model that incorporates uncertainty and the impact of internal delivery vehicles. The joint use of these tools provides managers with enhanced information to decide on the quality and robustness of the proposed schedules, resulting in better solutions for everyday situations. The intelligent system based on the optimization-simulation model provides clear benefits to maritime terminal management. This system efficiently identified high-quality schedules and can be used to evaluate its robustness. It was also flexible and can easily be adapted if other components need to be introduced, which may affect the goodness of a schedule.

Kizilay et al. (2020) proposed constraint-programming models for integrated container terminal operations[11]. The aim was to reduce the ship's circulation time and increase the port's efficiency. Also in this model, import and export containers are considered in the same way. (For complex examples, a two-step optimization approach can be used).

Yue et al. (2021) disclosed that meeting individual needs increases competition between container terminals [12]. To this end, they examined the issue of integrated scheduling of existing equipment and divalent AGVs. They formulated the problem with a two-stage mixed correct planning model to maximize customer satisfaction and minimize service latency. Then they used a sorting genetic algorithm to solve the problem. Numerical results showed the effectiveness of the proposed model and algorithm.

Table 1 shows a summary of the literature review of integrated handling equipment scheduling in automated container terminals. Major equipment includes Quay Cranes (QC), Yard Cranes (YC), Automated Guided Vehicle (AGV), and Automated Lifting Vehicles (ALV).

[DOR: 20.1001.1.23456000.2021.16.0.6.1]

Table 1. Summary of the review around integrated handling equipment scheduling [13]

Author s (Year)	Handling Equipment	Objective	Constraints	Model	Solution method
Steenken et al. (2001)	QC, IT and YC	Avoid waiting times at the quay cranes	Coordinated between vehicles and cranes	Just-in-time scheduling model	"what if" simulation
Homayouni et al. (2011)	QC, IT and YC	Minimize delay	Coordinated between vehicles and cranes	Mixed Linear Programming	Genetic algorithm (GA)
Chen et al. (2013)	QC, IT and YC	Minimize makespan	ITs are shared among different ships	Mixed Linear Programming	A three-stage algorithm consisting of heuristic and disjunctive graph
Tang et.al.(2014)	QC, IT and YC	Minimize makespan	QC and IT Minimize makespan Inequalities and lower bounds MILP	Mixed Linear Programming	CPLEX and particle swarm optimization (PSO)
He et al. (2015)	QC, IT and YC	Minimize delay and energy consumption	Time-saving and energy-saving	Mixed Linear Programming	GA and PSO
Yang et al. (2018)	QC, AGV, and YC	Minimize makespan prevention	Conflictive and congestion	Bi-level programming model	Bi-level GA
Roy and de Koster(2018)	QC and ALV	Improve seaside processes	Vehicle queuing network	Integrated stochastic models	Markov chain analysis and traffic simulation
Zhao et al. (2019)	QC and AGV	Minimize energy consumption	Transfer platform capacity	Mixed Linear Programming	A two-stage taboo search algorithm
Vahdani et al.(2019)	QC and IT	Minimize costs and minimize the maximum workload	Distribution and sharing of trucks	Bi-objective optimization model	NSGA-II and multi-objective particle swarm optimization
Zhong et al. (2020)	QC, AGV, and YC	Minimize makespan	Coordination of main trolley and portal trolley of QC	Mixed Linear Programming	Hybrid GAPSO algorithm with adaptive
Castilla et al. (2020)	QC and IT	Minimization cost	System uncertainty	Mixed Linear Programming	Simulation
Kizilay et al. (2020)	QC, AGV, and YC	minimizing the turnover times of the vessels	Coordinated between vehicles and cranes	Constraint programming	two-step optimization approach
Yue et al. (2021)	QC and AGV	Maximize customer satisfaction, minimize delay of QCs and idle time of AGV	Customer satisfaction, buffer capacity of blocks and AGV endurance	Two-stage and bi-objective Mixed Linear Programming	GUROBI and NSGA-III

3. Problem Description and Modeling

In this section, the problem of equipment management of a container terminal is investigated with the aim of reducing the duration of the ship at the berth and increasing the speed of the service process. A scenario is considered to examine the problem. In this scenario, a ship anchors at zero time for loading and unloading a number of containers at the berth. During anchoring, it is known how many quay cranes and which quay cranes are operating on the ship. It also specifies how many containers should be unloaded from the ship and how many containers should be loaded on the ship. The source and destination of each container job are also specified at the time of ship anchoring. A number of automated guided vehicles are responsible for transporting these container jobs. The problem, here, is to find the shortest possible time to transfer containers from the quay-side to the yard-side or vice versa. The problem was formulated and expressed in terms of complexity in the following.

3.1. Complexity of the problem

The proposed problem has a very large search space and is one of the NP-Hard problems. Given N as the number of container jobs to be carried and C_V as the number of container jobs to be carried by the vehicle *v*, we can calculate the size of search space to find the optimal solution. For example, if all N container jobs must be carried by only one automated guided vehicle, the number of containers jobs to be carried by this vehicle is the C_1. Hence, we have the equation (1) and the size of the search space, in this case, is equal to the number of permutations in the transportation of N container jobs, it will be $(N)!$.

$$C_1 = N. \qquad (1)$$

In addition, if we have M automated guided vehicles and only one container job to be carried, the size of the search space will be equal to the number of non-negative correct answers of the equation (2), i.e. M.

$$C_1 + C_2 + \cdots + C_{\mathrm{M}} = 1. \qquad (2)$$

Therefore, if the problem has M automated guided vehicles and N containers, the problem search space will be equal to all permutations of the non-negative correct answers of the equation (3). Therefore, in general, the size of the search space is equal to the value of equation (4).

$$C_1 + C_2 + \cdots + C_{\mathrm{M}} = N \qquad (3)$$

$$(C_1)! + (C_2)! + \cdots + (C_M)! = \binom{N+M-1}{M-1} = \frac{(N+M-1)!}{(N)!\,(M-1)!} \qquad (4)$$

The problem can be compared with the Minimum Cost Flow (MCF) model, formulated in Chapter 4 of the book [14]. To do this, we assume a directional graph GAGV = (NAGV, EAGV), with four types of nodes as follows:

a) **$AGVN_m$:** a supply node corresponding to AGV m with one unit supply (AGVN stands for the AGV Node). There are M AGVs in the problem. Hence, there are M supply nodes in the GAGV. We define the following set for these supply nodes:
SAGVN: a set of M supply nodes as denoted by SAGVN = {$AGVN_m$ | m=1,2,…,M}.

b) **$JPUN_i$:** It is a node in which an AGV pick-up job i. It stands for the Job-Pick-Up Node. There is neither supply nor demand in this node, i.e. it is a transshipment node. We define the following set for these transshipment nodes:
SJPUN: It is a set of N Job-Pick-Up nodes in the GAGV, denoted by SJPUN = {$\mathbf{JPUN}_i$ | i=1,2,…,N}.

c) **$JDPN_i$:** a node in which an AGV delivers the job i. It stands for the Job-Delivery-Point Node. Like the previous nodes, there is neither supply nor demand in this node. We define the following set for these transshipment nodes:
SJDPN: It is a set of N Job- Delivery-Point nodes in the GAGV, denoted by **SJDPN** = {$\mathbf{JDPN}_i$ | i=1,2,…,N}

d) **SINK:** It stands for a Sink node or a demand node in the NAGV with M units demand.

Therefore, if we have the number of M AGV and the number of N container jobs in the problem, the total number of nodes in the MCF model will be equal to $M + 2 \times N + 1$. The set of nodes in GAGV is according to equation (5):

$$NAGV = SAGVN \cup SJPUN \cup SJDPN \cup SINK \qquad (5)$$

We have four types of edges in the GAGV as follows:

a) **Inward Arcs:** There is a directed arc from every AGV node, to the Job-Pick-Up node of job i. We define the following notation for these arcs as below:
ARC_{inward} : a set of arcs from SAGVN to SJSN, denoted by ARC_{inward} ={ (m, j) | ∀m ∈ SAGVN, ∀j ∈ SJPUN}
The number of these arcs in the GAGV is M×N. Each arc has the lower bound zero, and the upper bound one, i.e., only one AGV goes through each of these arcs. As we mentioned before (see Assumption 5-10), our objectives are to minimize waiting and travelling times of the AGVs and the lateness times of jobs. The cost between node m and node j is calculated as described in Chapter 4 of the book [14].

b) **Intermediate Arcs:** There is a directed arc from every Job-Delivery-Point node i to other Job- Pick-Up node j. We define the following notation for these arcs:
$ARC_{intermediate}$: It is a set of arcs from SJPUN to SJDPN, denoted by $ARC_{intermediate}$ ={ (i, j) | ∀i ∈ SJPUN, ∀j ∈ SJDPN, j≠ $JPUN_i$ }. The number of these arcs in the GAGV is N×(N-1). Each arc has the lower bound zero, and the upper bound one, i.e., only one AGV goes through from one job to another. The cost

[DOR: 20.1001.1.23456000.2021.16.0.6.1]

between node i and node j in the GAGV is calculated as what described in Chapter 4 of the book [17].

c) **Outward Arcs:** There is a directed arc from every Job-Delivery-Point node i and AGV node m to SINK. We define the following notation for these arcs as follows:
$ARC_{outward}$: It is a set of arcs from SJPUN and SJDPN to SINK, denoted by $ARC_{outward} = \{ (i, j) \mid \forall i \in SAGVN \; U \; SJPUN, j=SINK \}$. These arcs show that an AGV can remain idle after serving any number of jobs or without serving any job. Therefore, a cost of zero is assigned to these arcs.

d) **Auxiliary Arcs:** There is a directed arc from every JPUN i to its JDPN. We define the following notation for these arcs as follows:
$ARC_{auxiliary}$: a set of arcs from SJPUN to SJDPN, denoted by $ARC_{auxiliary} = \{ (i, j) \mid \forall i \in$ SJPUN, j=an unique Job-Delivery-Point node in SJDPN, correspond to the JPUN i}. These arcs have unit lower and upper bounds. The transition cost across these arcs is the distance time between the source and destination of container jobs. These auxiliary arcs guarantee that every JPUN and JDPN is visited once only so that each job is served.

Therefore, the set of arcs in GAGV is according to equation (6) and the number of arcs is M×N+N× (N-1)+M+2×N.

$$EAGV = ARC_{inward} \; U \; ARC_{interm ediate} \; U \; ARC_{outward} \; U \; ARC_{auxiliary} \qquad (6)$$

In this model, the problem search space is equal to finding the number of M paths, starting from each node in the *SAGVN* and ending at the **SINK**. In these routes, all nodes at the beginning and end of each container job must be covered. Figure 3 shows the graph for 2 AGV and 4 container jobs. Suppose that for some values of arc costs, the solution paths are 1 → 3 → 4 → 9 → 10 → 11 and 2 → 5 → 6 → 7 → 8 → 11. This states that AGV 1 is assigned to serve container jobs 1 and 4, and AGV 2 is assigned to serve container jobs 2 and 3, respectively.

Since the cost of arcs in the minimum cost flow model is an integer value, it enables us to model the problem as an integer linear program. The known parameters before decision making and decision variables are shown in Tables 2 and 3, respectively.

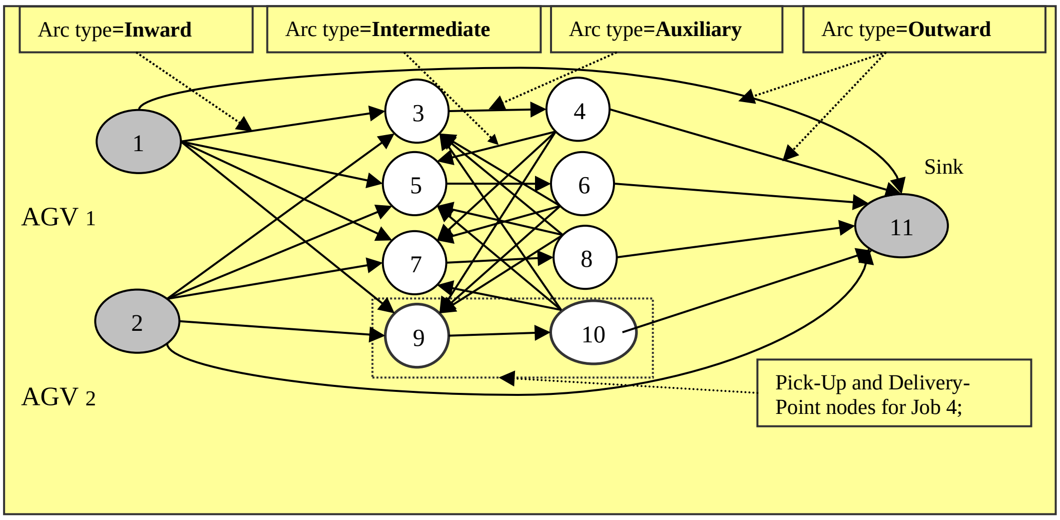

Figure 3. An example of the MCF model for 2 AGVs and 4 container jobs [14[

[DOR: 20.1001.1.23456000.2021.16.0.6.1]

Table 2. Known Parameters before Decision Making

Number	Symbols	Description
(7)	$V = \{1.2.3.\ldots.\ldots v\}$	Set of Automated Guided Vehicles
(8)	$B = \{1.2.3.\ldots.\ldots b\}$	Set of total storage blocks in the terminal.
(9)	$Q = \{1.2.3.\ldots q\}$	Set of total quay cranes in the terminal.
(10)	$Q_{Active} \subset Q$	Set of active quay crane on the ship.
(11)	$C_{Inbound} = \{1.2.3.\ldots.i\}$	Set of inbound containers.
(12)	$C_{Outbound} = \{1.2.3.\ldots.j\}$	Set of outbound containers.
(13)	$C_{Total} = C_{Inbound} \cup C_{Outbound}$	Set of total containers.
(14)	$C_v \subset C_{Total} .\ \forall v \in V$	The number of container jobs to be carried by the vehicle *v*
(15)	$\bigcup_{v=1}^{M} C_v = C_{Total}; \bigcap_{v=1}^{M} C_v = \emptyset$	Set of container jobs that each AGV must carry.
(16)	$S = Q_{Active} \cup B$	Set of pick-up nodes of containers.
(17)	$D = Q_{Active} \cup B$	Set of delivery-point nodes of containers.
(18)	CN	Set of cross nodes in path.
(19)	$N^* = Q \cup B \cup CN$	Set of total nodes in the path.
(20)	$L = \{l_1.l_1.\ldots.l_v\}.\ l_v \in N^*$	Set of fist location of AGV.
(21)	$W_{s_i.d_i}$	The distance-time from the location s_i to the location d_i
(22)	$s_i \in Q_{Active} \cup B$	The pick-up node of container job i
(23)	$d_i \in Q_{Active} \cup B$	The delivery-point node of container job i
(24)	$If\ i = 1 \Rightarrow AT_{v1} = \begin{cases} 0 & if\ l_v = s_1 \\ W_{l_v.s_1} & else\ l_v \neq s_1 \end{cases}$ $Otherwise\ AT_{vi} = \begin{cases} 0 & if\ d_{i-1} = s_i \\ W_{d_{i-1}.\ s_i} & else\ d_{i-1} \neq s_i \end{cases}$	The arrival time of the AGV v to the starting point of the container i
(25)	$TT_{vi} = W_{s_i.d_i}$	Duration of movement of container job i from the source node to the destination by the AGV v

Table 3. Decision Variables

Row	Variables	Description
(26)	$Y_{imn} = \{0\ or\ 1\}$	If container job i is sent from its source location by the crane m (Quay-side or yard-side) to its destination location by the crane n (Quay-side or yard-side), then Y_{imn}=1, otherwise it is zero.
(27)	$X_{vi} = \{0\ or\ 1\}$	If the AGV *v* carries container i, X_{vi} =1 otherwise it is zero

[DOR: 20.1001.1.23456000.2021.16.0.6.1]

11.4.2 Problem Formulation

The objective function of the model is to minimize the total time to handle all container jobs by the set of vehicles in the container terminal, according to the following function:

$$\text{Min}\left\{\underset{\forall v\in V.\forall i\in C_{Total}}{\text{Max}}\ (AT_{vi}+TT_{vi})\right\} \tag{28}$$

The constraints are as follows:

$$\sum_{q\in Q_{Active}}\sum_{b\in B} Y_{iqb}=1;\forall i\in C_{Inbound} \tag{29}$$

$$\sum_{b\in B}\sum_{q\in Q_{Active}} Y_{ibq}=1;\ \forall i\in C_{Outbound} \tag{30}$$

$$\sum_{v\in V} X_{vi}=1\ ;\ \forall i\in C_{Total} \tag{31}$$

The constraints in the equation set (29) ensures that each inbound container is sent from the quay-side to the yard-side. The constraints in the equation set (30) ensures that each outbound container is sent from the yard-side to the quay-side. The constraints in the equation set (31) ensures that each container job is handled by only one automated guided vehicle.

4. The proposed method

As mentioned before section, at the start of the ship processes, parameters such as the number of the containers to be serviced, the number of quay cranes have to work on the ship, the number of AGVs that must transfer the containers, and the number of storage blocks Characterized. To simplify the problem, the problem is divided into two sub-problems. The first part is assigning the equipment to the container job in the terminal. In this step, it should be specified which container job should be serviced with which quay crane and which automated guided vehicle. The first part of the problem uses a greedy algorithm to assign source and destination to each container job.

An automated container terminal is provided to examine the proposed method. In this scenario, there is a container terminal with 8 quay cranes, 8 blocks, and 8 automated guided vehicles. There are 5 quay cranes with numbers 4 to 8 operating on the ship. 10 container jobs must be loaded from the ship and sent to the blocks for storage. 6 container jobs should be loaded on the ship and sent to the dock crane for delivery. All container jobs are equal to the sum of inbound and outbound container jobs. Figure 4 shows the source and destination for 10 inbound container jobs and 6 outbound container jobs. For example, container 1 delivered into storage block 3, and container 11 should be delivered to quay crane 7.

The layout of a docked ship and the location of the quay crane and blocks is shown in Figure 5. In this picture, the container terminal includes 8 quay cranes and 8 storage blocks. In order to prevent congestion and accidents, the movement path AGV was considered clockwise and the speed of all AGV was considered 5 meters per second.

The second part of the problem is finding the order of servicing container jobs for each automated guided vehicle and routing to transport container jobs from the source to the destination. In each container terminal, there is a specific path for the AGV to transport the container from the quay-side to the yard-side or vice versa. Since finding the number of containers and the optimal order for servicing container jobs and navigating automated guided vehicles is a NP-Hard problem, in this study, a sorting genetic algorithm is used to find the optimal local solution. The flowchart of the sorting genetic algorithm presented in Figure 6 is shown.

QC=4	QC=8	QC=8	QC=7	QC=7	QC=7	QC=5	QC=7	QC=4	QC=7	B=1	B=3	B=1	B=1	B=7	B=6
B=3	B=8	B=1	B=4	B=4	B=7	B=7	B=2	B=4	B=4	QC=7	QC=7	QC=7	QC=5	QC=7	QC=7
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Figure 4. The source and destination of 16 containers

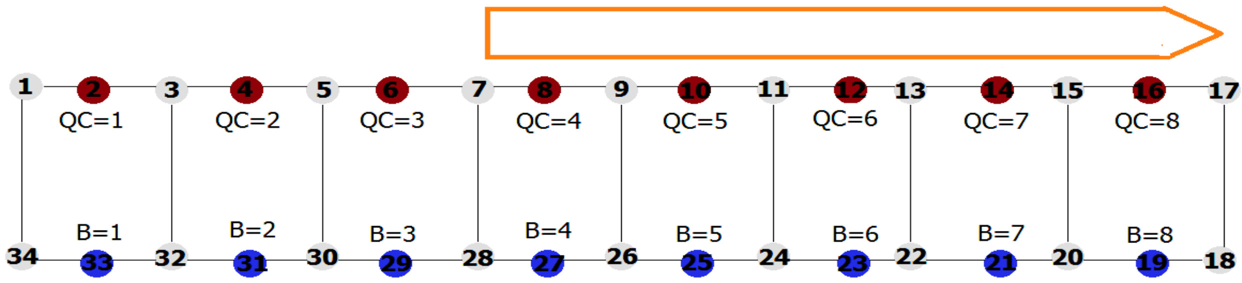

Figure 5. A docked ship with 5 quay cranes worked on

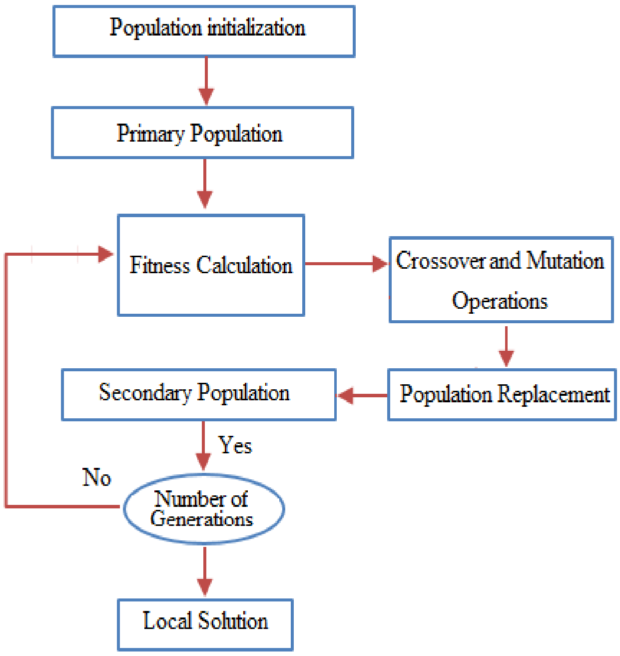

Figure 6. The Flowchart of the presented sorting genetic algorithm.

Each chromosome represents the order of service of existing container jobs. The amount of fit function for each chromosome is equal to the total time from the previous location node to the Job-Pick-Up node plus the time of carrying the container job from the Job-Pick-Up node to the Job-Delivery-Point node for all container jobs. To solve the problem of the second part, the number of AGVs for servicing container jobs is considered as one. In this case, the problem can be solved by a genetic sorting algorithm.

4.1. Chromosome

The proposed sorting genetic algorithm uses a three-level chromosome. Because in the first part of the problem, the source and destination of each container are specified, the node number of the Job-Pick-Up and Job-Delivery-Point of each container job is identified. In the first level of chromosome, the Job-Pick-Up node number is placed, in the level of the node number, the Job-Delivery-Point of the container job is placed, and in the third level, the number of container jobs to be serviced is placed. In Figure 7, a three-level chromosome for 16 container jobs with 10 inbound jobs and 6 outbound jobs has been shown.

4.2. Crossover operator

In order to apply the crossover operation, at first, a number of parents must be selected from the existing population based on the specified rate. Depending on the fitness function, each parent is likely to be selected to perform crossover operations. Each chromosome that has a better fitness function, small is better, is more likely to be selected. The crossover operation is performed in four stages. In the first stage, the desired points for the intersection are determined and in the second stage, the existing container jobs are changed between the two intersection points. In the third stage, non-duplicate container jobs, and in the fourth stage, duplicate container jobs are inserted in the chromosomes. Each step has been described in more detail as follows.

perform the intersection operation. After selecting two chromosomes as parent 1 and parent 2, in this operation, two random numbers with uniform distribution are selected as the intersection points in the parent chromosome. For example, in Figure 8 the two selected parents with intersection points 6 and 10 are shown.

8	16	16	14	14	14	10	14	8	14	33	29	33	33	21	23	Source
29	19	33	27	27	21	21	31	27	27	14	14	14	10	14	14	Destination
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Container

Figure 7. The three level chromosome for 16 container jobs, consist of 10 inbound jobs and 6 outbound jobs

[DOR: 20.1001.1.23456000.2021.16.0.6.1]

A two-point intersection operator has been used to
In the next step, the genes are exchanged between the two intersection points. The value zero is replaced in the rest of the jobs because the number of container jobs cannot be duplicated. In Figure 9 the stage of gene exchanging between intersection points has been shown.

After exchanging genes between the intersection points, the number of non-duplicate container jobs must be added to the children. For example, in child 1, the container job 1 is inserted to the desired location because it is not duplicate, but in child 2, the container job 7 is not allowed to be inserted in the desired location due to duplication. The step of adding non-repetitive container jobs is shown in Figure 10.

Parent 1:

8	16	16	14	14	14	10	14	8	14	33	29	33	33	21	23
29	19	33	27	27	21	21	31	27	27	14	14	14	10	14	14
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Parent 2:

10	14	21	33	14	14	33	14	23	8	16	29	33	16	14	8
21	31	14	14	27	21	10	27	14	27	33	14	14	19	27	29
7	8	15	13	5	6	14	10	16	9	3	12	11	2	4	1

Figure 8. Selected Parents with intersection points 6 and 10

Child 1:

0	0	0	0	0	0	33	14	23	8	0	0	0	0	0	0
0	0	0	0	0	0	10	27	14	27	0	0	0	0	0	0
0	0	0	0	0	0	14	10	16	9	0	0	0	0	0	0

Child 2:

0	0	0	0	0	0	10	14	8	14	0	0	0	0	0	0
0	0	0	0	0	0	21	31	27	27	0	0	0	0	0	0
0	0	0	0	0	0	7	8	9	10	0	0	0	0	0	0

Figure 9. Genes Exchanging between Intersection Points.

Child 1:

8	16	16	14	14	14	33	14	23	8	33	29	33	0	21	0
29	19	33	27	27	21	10	27	14	27	14	14	14	0	14	0
1	2	3	4	5	6	14	10	16	9	11	12	13	0	15	0

Child 2:

0	0	21	33	14	14	10	14	8	14	16	29	33	16	14	8
0	0	14	14	27	21	21	31	27	27	33	14	14	19	27	29
0	0	15	13	5	6	7	8	9	10	3	12	11	2	4	1

Figure 10. Adding non-repetitive Containers to children.

[DOR: 20.1001.1.23456000.2021.16.0.6.1]

Child 1:

8	16	16	14	14	14	33	14	23	8	33	29	33	10	21	14
29	19	33	27	27	21	10	27	14	27	14	14	14	21	14	31
1	2	3	4	5	6	14	10	16	9	11	12	13	7	15	8

Child 2:

33	23	21	33	14	14	10	14	8	14	16	29	33	16	14	8
10	14	14	14	27	21	21	31	27	27	33	14	14	19	27	29
14	16	15	13	5	6	7	8	9	10	3	12	11	2	4	1

Figure 11. Adding non-repetitive Containers to children.

4.3. Mutation Operator

In the proposed solution, a swap mutation operator had been used. At first, a number of parents are selected based on the Mutation rate. Then, two random numbers are generated using a uniform distribution. The numbers generated indicate the container job places that need to be swapped. For example, in Figure 12 swap mutation operation to the displacement of container jobs 4 and 11 are shown. In this example, the container job 4 is an inbound container and the container job 11 is an outbound container.

5. Simulation and Evaluation

As mentioned in the previous section, increasing the efficiency of container terminals is directly related to the speed and service life of anchored ships. In this research, the issue of integrated container equipment scheduling has been investigated. The proposed method determines the appropriate order of service based on the origin and destination of the container job. In this section, the simulation details of the proposed model are discussed. The proposed algorithm is developed using a structured programming method. The proposed method was implemented using MATLAB programming language and the algorithm parameters were calculated using Taguchi method. Finally, the proposed algorithm was compared with the Particle Swarm Optimization (PSO) algorithm and combinations of the PSO algorithm and Genetic Algorithm. Due to the random method of solving, each problem was solved 10 times. In the end, the execution time and the objective function values were reported for a number of problems. All tests were performed on a computer with a 2.4 GHz processor and 2 GB of RAM.

5.1. Parameters

The proposed method has 4 main factors: number of generations, population number, crossover rate, and mutation rate. For each factor, four different levels have been examined. The values checked are reported in Table 4. The parameters of the proposed method were examined using Minitab software and Taguchi method. To investigate 256 problems were designed and due to the random nature of the algorithm, each problem was performed 10 times and the mean of the objective function values for each problem was reported. Then the values obtained for each problem were standardized by Robust parameter design (RPD) method and analysed by Taguchi method.

Parent:

8	16	16	14	14	14	10	14	8	14	33	29	33	33	21	23
29	19	33	27	27	21	21	31	27	27	14	14	14	10	14	14
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Child:

8	16	16	33	14	14	10	14	8	14	14	29	33	33	21	23
29	19	33	14	27	21	21	31	27	27	27	14	14	10	14	14
1	2	3	11	5	6	7	8	9	10	4	12	13	14	15	16

Figure 12. Swap Mutation Operation

[DOR: 20.1001.1.23456000.2021.16.0.6.1]

Table 4. Factors Names and Values.

Factor Name	Values			
Number of Generations	100	200	300	400
Number of Populations	30	40	50	60
Crossover Ratio	0.5	0.6	0.7	0.8
Mutation Ratio	0.02	0.04	0.07	0.1

Figures 13 and 14 show the Main Effects Plot for Means and SN ratio, respectively, for determining the importance of factors in the solution method.

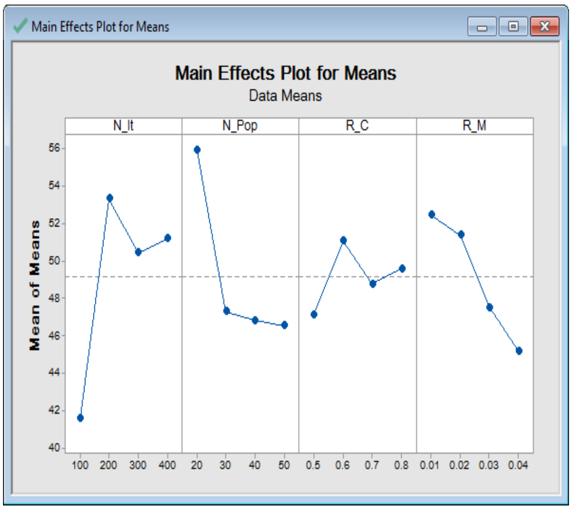

Figure 13. Main Effects Plot for Means

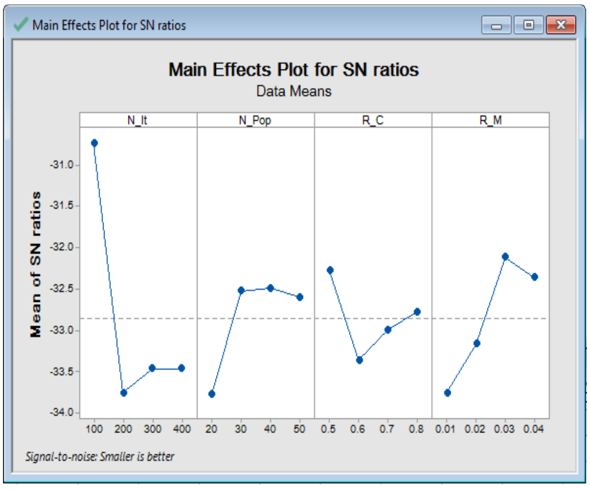

Figure 14. Main Effects Plot for SN ratios.

From these figures, we can observe that:

- **Observation-1**: Figure 13 identifies which factor has had the greatest impact on response changes. Because they have a wide range, it shows that they are more important. In this figure, the smaller the objective function, the better all four factors are almost equally important. Between each level in each factor is at least better. For example, for the number of Generation (iteration) 100 is a better choice.
- **Observation-2**: Figure 14 shows the importance of factors in the solution method and identifies which factor has had the greatest impact on response changes. However, in this figure, the value of each factor as bigger is better. Then, due to the smaller, the objective function is better all four factors are almost equally important. Between each level in each factor is at least better. For example, for the number of Generation (iteration) 100 is a better choice.

According to the analysis of the graphs obtained from the Taguchi method, as shown in Figures 13 and 14, the number of production iterations is 100, the population is 40, the crossover rate is 0.5 and the jump rate is 0.3. The parameters used in the genetic algorithm are shown in Table 5.

Table 5. parameters and values.

Parameters	Values
Number of Generations	100
Number of Populations	40
Crossover Rate	0.5
Mutation Rate	0.03

5.2. Numerical Experiments

In order to evaluate the efficiency and effectiveness of the proposed method, a number of problems were designed, and then its methods and combinations were examined with the proposed method. Figure 15 shows the objective function for the number of iterations of 100 generations and the population size of 50 when the number of tasks in container 16 (8 inbound containers - 8 outbound containers). From this figure, we can observe that:

- **Observation-3:** As can be seen, the convergent genetic algorithm finds the optimal local solution for the expressed scenario. In this experiment, the value of the objective function is equal to 201 after 100 generations.

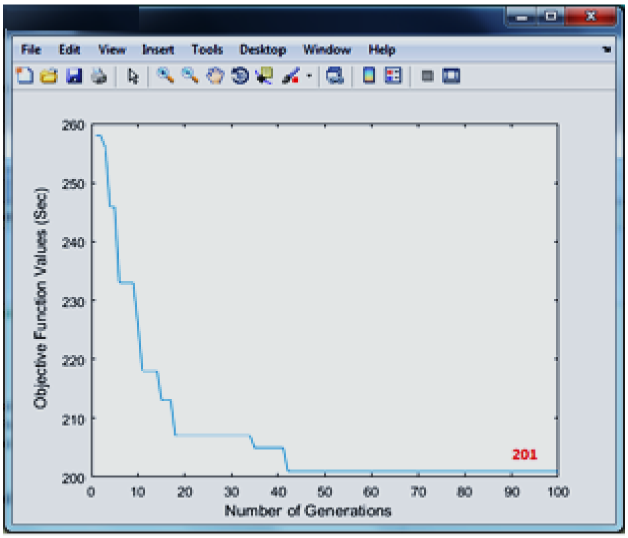

Figure 15. Convergence of GA for case with 100 generations.

Figures 16 and 17 show a comparison of CPU time and the values objective function when the problems are solved by GA, PSO, GA+PSO, and PSO+GA algorithms, respectively. From these figures, we can observe that:

- **Observation-4**: The genetic algorithm has less execution time than the other three algorithms. Accordingly, Figure 17 shows the genetic algorithm has a better value for the objective function than the other three algorithms.

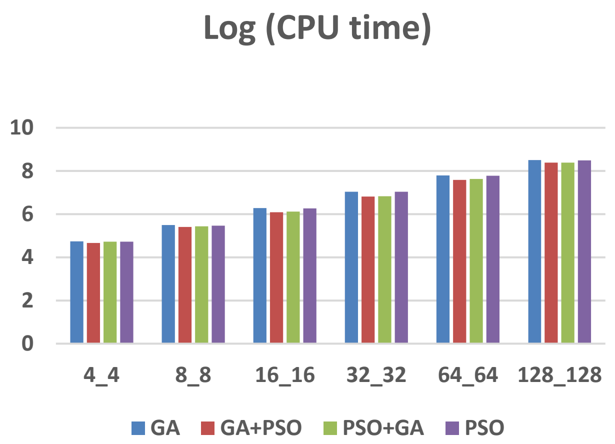

Figure 16. A Comparison of Log (CPU Time Spent by GA, PSO, GA+PSO, and PSO+GA algorithms).

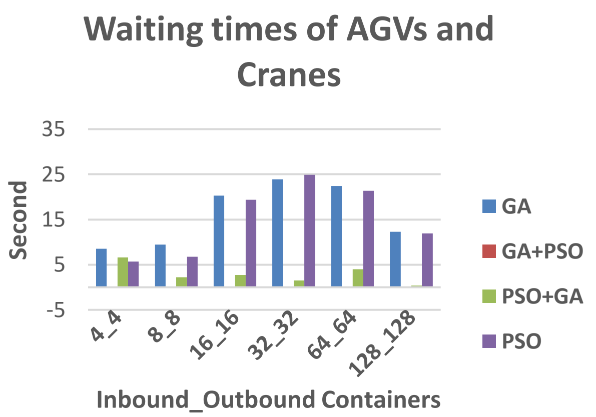

Figure 17. Comparison of objective function values for GA, PSO, GA+PSO, and PSO+GA algorithms.

To compare the efficiency of the proposed method with the three other algorithms, we calculated the waiting time of the AGVs and Cranes. The result of this calculation is shown in Figure 18. From this figure, we observe that:

- **Observation-5:** the waiting time of the vehicles and cranes in solving the problems by GA is slightly more than the three other algorithms**.**

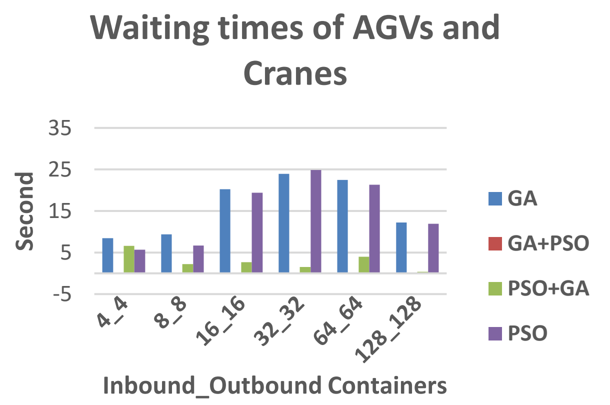

Figure 18. Comparison Gap of the objective function values.

Given that the container jobs and their location of source and destination, randomly with a uniform distribution, for each scenario designed, the proposed simulated method is performed 10 times, and the average execution time and objective function are reported in Table 5. Is. In all designed scenarios, the number of inbound jobs is equal to the number of outbound jobs. From this table, we observe that:

- **Observation-6**: The results of experiments 1,3,6,9,12 show that by doubling the number of container jobs, the amount of objective function and execution time almost doubles.
- **Observation-7**: Experiments 3, 4, 5 show that by doubling the number of AGVs, the amount of objective function remains almost half and the execution time remains constant.
- **Observation-8**: Experiments 2, 5, 8, 11 show that by keeping the ratio of container job to AGVs constant (half), the objective function is almost constant (slightly increased) but the execution time is increased.

Table 5. The main results of the experiments

	Inbound-Outbound Containers	AGV	Active Quay Crane	Objective Function Values	CPU Time (Sec)
1	4-4	1	5	106	396.11
2	4-4	2	5	52	883.46
3	8-8	1	5	199	059.16
4	8-8	2	5	107	233.17
5	8-8	4	5	61	480.17
6	16-16	1	5	430	693.24
7	16-16	2	5	226	764.26
8	16-16	8	5	70	379.27
9	32-32	1	5	945	777.44
10	32-32	2	5	493	360.46
11	32-32	16	5	75	001.47
12	64-64	1	5	2073	540.83
13	64-64	2	5	1076	074.90
14	128-128	1	2	4253	231.180
15	128-128	1	5	4452	477.165
16	128-128	1	8	4766	392.163

6. Conclusions and future work

In this research, the problem of integrated management of equipment in automated container terminals with the aim of reducing the service time of berthed ships. The literature on the problem, including decisions, solutions, formulation, and implementation was reviewed. The complexity of the proposed problem was investigated and then the problem was formulated as a linear integer-programming model. A solution based on a combination of the greedy algorithm and the genetic algorithm was proposed. This solution was named Sorting Genetic Algorithm (SGA). The parameters of the proposed method were investigated using Minitab software and Taguchi method to determine the appropriate values. To show the efficiency and effectiveness of the proposed method, the results were compared with the PSO algorithm and its combinations with the proposed method. Finally, execution time and objective function values of the comparison were reported.

The results show that not only the sorting genetic algorithm increased the efficiency and productivity of container terminals by adjusting the order of container operations but also can be used for measurement and prediction of the time required to docked the ship at the berth to load and unload containers. Additionally, the proposed method showed a reduction in execution time and finding a better local solution. For future research, the proposed method for dynamic scenarios will be considered. In addition, another heuristic algorithm can be used as a solution and predictions of needed service time.

References

[1]. Kim, J.; Hong, E.J.; Yang, Y.; Ryu, K.R. Noisy Optimization of Dispatching Policy for the Cranes at the Storage Yard in an Automated Container Terminal. Applied Sciences. 2021, 11, 6922.

[2]. Steenken D, Winter T, Zimmermann U.T, “Stowage and Transport Optimisation in Ship Planning”, Springer, Berlin, pp. 731-745, 2001.

[3]. Chen, L., Langevin, A., Lu, Z., Integrated scheduling of crane handling and truck transportation in a maritime container terminal. Eur. J. Oper. Res. 225 (1), 142e152. 2013.

[4]. Tang, L., Zhao, J., Liu, J., Modeling and solution of the joint quay crane and truck scheduling problem. European Journal of Operation Research, 236 (3), 978e990, 2014.

[5]. He, J., Huang, Y., Yan, W., Integrated internal truck, yard crane and quay crane scheduling in a container terminal considering energy consumption. Expert Syst. Appl. 42 (5), 2464e2487, 2015, .

[6]. Roy, D., de Koster, R., Stochastic modeling of unloading and loading operations at a container terminal using automated lifting vehicles, European Journal of Operational Research,. 266 (3), 895e910, 2018, .

[7]. Yang, Y., Zhong, M., Dessouky, Y., & Postolache, O., An integrated scheduling method for AGV routing in automated container terminals. Computers & Industrial Engineering, 126, 482-493, 2018.

[8]. Vahdani, B., Mansour, F., Soltani, M., Bi-objective optimization for integrating quay crane and internal truck assignment with challenges of trucks sharing. Knowl. Base Syst. 163, 675e692, 2019, https://doi.org/10.1016/j.knosys.2018.09.025.

[9]. Zhao, Q., Ji, S., Guo, D., Research on cooperative scheduling of automated quayside cranes and automatic guided vehicles in automated container terminal. Math. Probl Eng. 1e15.

[10]. Castilla Rodríguez, I., Exposito-Izquierdo, C., Melian-Batista, B., Simulation-optimization for the management of the transshipment operations at maritime container terminals. Expert System Application, 139, 112852, 2020.

[11]. Kizilay, D., Van Hentenryck, P., & Eliiyi, D. T., Constraint programming models for integrated container terminal operations. European Journal of Operational Research, 286(3), 945-962, 2020.

[12]. 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. Journal of Cleaner Production, 292, 126019,2021.

[13]. Lijun Y., Houming F., Mengzhi M., Optimizing configuration and scheduling of double 40 ft dual-trolley quay cranes and AGVs for improving container terminal services, Journal of Cleaner Production 292 (2021) 126019, 2021.

[14]. Rashidi, H and Tsang, E., (2015). Vehicle Scheduling in Port Automation: Advanced Algorithms for Minimum Cost Flow Problems, Second Edition (2nd ed.). CRC Press.

[DOR: 20.1001.1.23456000.2021.16.0.6.1]