mathematics

Article

An Optimization Approach to Berth Allocation Problems

Shu-Chuan Chang [1], Ming-Hua Lin [2] and Jung-Fa Tsai [3,*]

[1] College of Management, National Taipei University of Technology, Taipei 106344, Taiwan
[2] Department of Urban Industrial Management and Marketing, University of Taipei, Taipei 111036, Taiwan
[3] Department of Business Management, National Taipei University of Technology, Taipei 106344, Taiwan
[*] Correspondence: jftsai@ntut.edu.tw

Abstract: The berth allocation problem determining the berthing time and position for incoming vessels in port operations has garnered increased attention within the global transportation network. This study focuses on the berth allocation problem with a continuous quay and dynamic vessel arrivals. With the overarching goal of enhancing service quality and optimizing berth utilization rates, this article proposes a mathematical programming model that minimizes the total waiting time of vessels and the overall completion time of vessel service. The formulated model is a mixed-integer linear programming problem that deterministic optimization techniques can globally solve. For large-scale problems, this study develops a genetic algorithm optimization approach to improve computational efficiency in reaching a near-optimal solution. Several numerical experiments are conducted to demonstrate the effectiveness and efficiency of the proposed approach.

Keywords: transportation; berth allocation problem; global optimization; genetic algorithm

MSC: 90B06

1. Introduction

With the globalization of the supply chain, many products are manufactured in multiple countries and shipped to customers worldwide. As international trade increases, the demand for transportation significantly rises, making ports crucial in logistics management. Port authorities must enhance service quality and optimize resource utilization to strengthen their competitiveness in the global transportation network. Berths are the most critical resources in port terminals due to the difficulty of expanding them quickly and their construction costs being the highest among all relevant costs for other facilities in the terminal. Developing an efficient and effective berth allocation strategy is essential for port authorities to manage the heavy and increasing ship traffic in the global transportation network [1–4].

The terminal operation system comprises three subsystems: vessel planning in the quayside area, storage and stacking in the yard, and transportation in the landside area [5,6]. The productivity and throughput of the container terminal between the seaside and the landside depend on the efficiency of the terminal handling system; therefore, optimizing the berth scheduling plan is critical for port authorities to increase port throughput and improve service quality [7,8]. This article focuses on the initial operation of seaside planning in container terminals, known as the berth allocation problem (BAP). In the survey of berth allocation and quay crane scheduling problems in container terminals reviewed by Bierwirth and Meisel [5], the BAP is commonly referred to as the assignment of quay space and service time to vessels that have to be unloaded and loaded at a terminal.

The BAP has been categorized into two classes based on the berth type and vessel arrivals [5]. According to a present partitioning of the quay into berths, the BAP can be modeled with a discrete or a continuous quay. In the discrete case, the quay is divided into several partitioned sections. Each single berth can be occupied by only one vessel

check for updates

Citation: Chang, S.-C.; Lin, M.-H.; Tsai, J.-F. An Optimization Approach to Berth Allocation Problems. *Mathematics* **2024**, *12*, 753. https://doi.org/10.3390/math12050753

Academic Editors: Ioannis G. Tsoulos and Andrea Scozzari

Received: 28 December 2023
Revised: 29 February 2024
Accepted: 29 February 2024
Published: 2 March 2024

simultaneously. In the continuous case, the quay is a continuous space that vessels can berth at arbitrary positions within the boundaries of the quay. Berth planning in a continuous layout leads to better utilization of quay space but is more computationally complicated than in a discrete layout [5,7,9]. In terms of vessel arrivals, the BAP can be divided into static and dynamic. The static one underlies the assumption that all vessels have arrived at the port before berth planning starts. Hence, the arrival time of vessels does not affect berthing assignment. The dynamic BAP allows vessels to arrive at any time during the planning time horizon, but their arrival times are known in advance, and each vessel will be served after its arrival. In common practice, the static assumption occurs only in a bustling port [5,10].

Since a continuous quay results in a higher utilization than a discrete quay, and the port authorities usually assign the berth to a vessel based on the expected arrival time before its arrival, this study focuses on the BAP with a continuous quay and dynamic vessel arrivals. The berth allocation aims to provide efficient and reliable services for each arriving vessel. Many studies have discussed the BAP with several practical factors and considered various constraints and objective functions.

Complete surveys of problem formulations, problem classification schemes, and solution algorithms for the BAP through 2022 are provided by Bierwirth and Meisel [5,11] and Rodrigues and Agra [3].

Rodrigues and Agra [3] categorized papers based on their objective functions and pointed out that the time spent by vessels at ports directly affected the service level. In the literature surveyed by Rodrigues and Agra [3], most papers incorporated minimized waiting time, handling time, completion time, or tardiness as components of the objective function [12,13]. The BAP can be addressed by minimizing waiting time, handling time, completion time, or tardiness to enhance port service levels or achieve optimal berth utilization.

The methods employed for addressing the BAP are diverse, broadly falling into two main categories: exact methods and heuristic methods. Exact methods typically employ mathematical models and run them through commercial solvers. However, it is noted that most exact methods are limited to solving small-scale instances and are often utilized to evaluate the performance of the proposed heuristic algorithms. On the other hand, heuristic methods constitute a significant portion of the approaches used to tackle BAP. The survey results from Rodrigues and Agra [3] indicate that the number of papers proposing heuristic solution methods is greater than those considering exact methods. A wide range of solution algorithms have been utilized over time, including genetic algorithms (GA), simulated annealing (SA), Tabu search (TS), artificial bee colony algorithm (ABC), squeaky wheel optimization (SWO), grey wolf optimization (GWO), and adaptive differential evolution algorithm (ADEA). Among these, genetic algorithms are the most commonly chosen method. Heuristics are employed in approximately 55% of the papers surveyed, with genetic algorithms being utilized in approximately 19% of the papers. Lv et al. [14], Rodrigues and Agra [15], and Wu and Miao [16] utilize genetic algorithms (GA) as part of the method for solving the BAP.

The following briefly reviews the literature on continuous and dynamic BAP. Lim [17], Kim and Moon [1], Guan and Cheung [18], Wang and Lim [6], and Lee et al. [19] transformed the berthing problem into a restricted form of the two-dimensional packing problem. Lim [17] used a graph-theoretical representation to capture the problem and proposed a heuristic method to minimize the length of quay occupied by vessels. Kim and Moon [1] aimed to minimize additional travel costs resulting from non-optimal berthing locations and penalty costs resulting from a delayed departure after the requested due time. A simulated annealing algorithm was developed for finding near-optimal solutions. Guan and Cheung [18] proposed a tree search procedure to minimize total weighted flow time, the sum of waiting and processing time of vessels. The weights reflect the relative importance of vessels. Wang and Lim [6] solved the problem by using a stochastic beam search algorithm. The constructed mixed-integer programming model determines the berthing time and the position to minimize the total berthing cost consisting of unallocation cost, berthing position cost, and delay berthing cost. Lee, Chen and Cao [19] followed the mathematical formulation of Guan and Cheung [18] to develop two versions of the greedy randomized adaptive search procedure (GRASP) in identifying the possible locations for the next vessel in the time-space diagram.

This study aims to improve port throughput and efficiency by optimizing berth utilization and enhancing service quality. Optimal berth utilization is achieved by minimizing the overall completion time of vessel service, while service quality improvement is pursued by minimizing the total waiting time for vessels. The proposed approach introduces a deterministic optimization model that formulates the BAP as a restricted form of two-dimensional packing problem solvable through existing deterministic optimization techniques to reach the globally optimal solution. Relating to the packing problem, the BAP has been shown to be an NP-hard problem [17]. The computational time to solve the deterministic optimization model significantly increases as the problem size increases. Consequently, this study also develops a genetic algorithm (GA) optimization approach to solve large-scale problems efficiently. The contributions of this paper include

- Formulate the BAP as a mathematical programming model of a two-dimensional packing problem solvable to reach the globally optimal solution;
- Develop a GA optimization approach to solve large-scale problems efficiently;
- Compare the performance of solving the BAP between a deterministic optimization model and a GA optimization approach.

The rest of this article is organized as follows. Section 2 formulates the BAP as a deterministic optimization model. Subsequently, a GA optimization approach is developed to solve large-scale BAP problems in Section 3. Section 4 presents some numerical examples to illustrate the proposed method. Finally, concluding remarks are made in Section 5.

2. Problem Formulation

This research considers the BAP with continuous quay and dynamic vessel arrivals. A deterministic optimization model is constructed to transform the BAP into a two-dimensional packing problem.

Figure 1 indicates the berth allocation time-space graph. The vertical axis represents the quay space, and the horizontal axis represents the berthing time. The quay space is divided into integer units. Each unit accommodates one vessel each time, and each vessel occupies an integer number of consecutive units when it is moored. Each rectangle represents a vessel; the height and length of the rectangle are the vessel's length and operation time, respectively. The x-coordinate of the bottom-left of the rectangle stands for the starting time of the operation. The y-coordinate of the bottom-left of the rectangle stands for the starting position of the operation. The BAP aims to decide the x-coordinate and y-coordinate of each rectangle in the space formed by the quay space and the berthing time. All vessels must be placed non-overlapping in the time-space graph for a feasible berthing assignment. Other constraints, such as the draft restrictions at the assigned berths for the vessels, must also be satisfied. The notations used throughout this study are defined below and indicated in Figure 2.

Berthing parameters, such as the number of berths required by the vessel, the estimated arrival time, and the estimated operation time, are typically filled 8 to 36 h before vessel arrival in the maritime practice (Lee and Chen [20]).

Parameters:

Y: the number of all partitioned berths in the quay;
n: the number of vessels to be scheduled;
p_i: the estimated operation time of vessel i;
q_i: the number of berths required by vessel i, $q_i \leq Y$;
k_i: the estimated arrival time of vessel i;
d_i: the draft of vessel i;
Bs_i: the starting position suitable for berthing vessel i;

Bn_i: the ending position suitable for berthing vessel i;

$$M = \max_i \{p_i,\ q_i\}.$$

Decision variables:

T: the overall completion time of handling all vessels at the port;
x_i: the starting time of handling vessel i;
y_i: the starting position of handling vessel i;
u_{ij}, v_{ij}: 0–1 variables used to control the relative positions of vessels i and j.

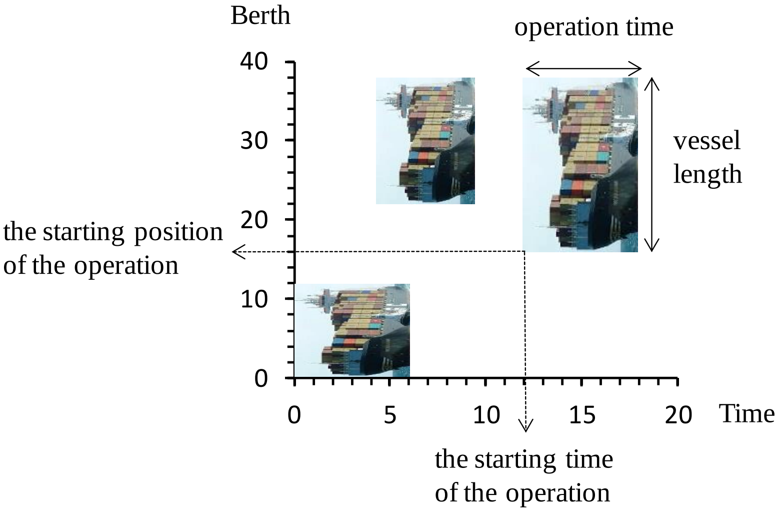
Berth
operation time
40
30
20
10
0
vessel length
the starting position of the operation
0
5
10
15
20
Time
the starting time of the operation

Figure 1. Time-space graph for berth allocation.

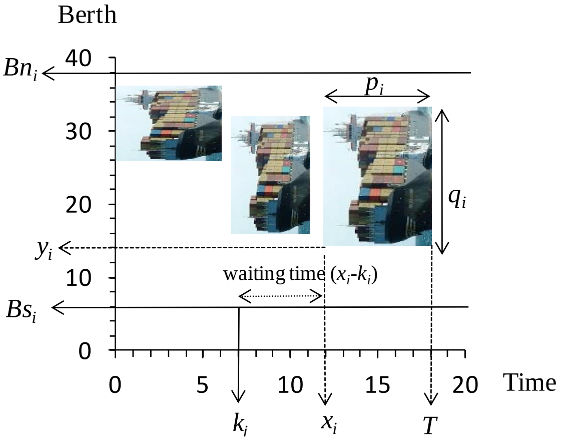

Figure 2. Variables used in the proposed model.

Four cases of non-overlap between two vessels i and j can be formulated by two binary variables u_{ij} and v_{ij} as follows and indicated in Figure 3.

Condition 1: $u_{ij} = 0$ and $v_{ij} = 0$ if and only if vessel i is at the right of vessel j.
Condition 2: $u_{ij} = 1$ and $v_{ij} = 0$ if and only if vessel i is at the left of vessel j.
Condition 3: $u_{ij} = 0$ and $v_{ij} = 1$ if and only if vessel i is above vessel j.
Condition 4: $u_{ij} = 1$ and $v_{ij} = 1$ if and only if vessel i is below vessel j.

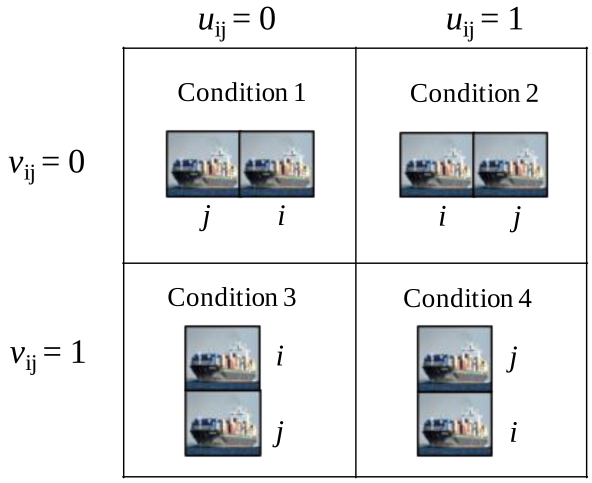

Figure 3. Diagram of relative positions of two vessels.

The following assumptions are introduced for formulating the BAP:

1. Each berth serves only one vessel at a time. The length of each berth is identical. Each vessel may occupy more than one berth according to the length of the vessel.
2. Vessels are served after arrival.
3. The operation time of each vessel is assumed deterministic and known in advance.
4. The number of berths required for each vessel is assumed deterministic and known in advance.
5. When a vessel starts operations at any berth, it will only move once loading/unloading is completed.
6. The draft of the vessel must be less than the water depth in the assigned berths.

The parameters Bs_i and Bn_i are determined by d_i. p_i is the estimated operation time of vessel i, and q_i is the number of berths required by vessel i. When the BAP is formulated as a two-dimensional packing problem, p_i and q_i are the length and height of a rectangle representing vessel i in a time-space graph. $M = \max_i\{p_i, q_i\}$ is used in constraints to guarantee that all rectangles will not overlap. That means all vessels cannot occupy the same berth simultaneously.

According to the discussions above, the mathematical formulation of the berth allocation optimization problem can be expressed as follows:

$$\text{Minimize}\left[\sum_{i=1}^{n}(x_i - k_i)\right] + T \tag{1}$$

subject to

$$(x_i - x_j) + u_{ij}M + v_{ij}M \geq p_j,\ \forall i, j,\ i < j, \tag{2}$$

$$(x_j - x_i) + (1 - u_{ij})M + v_{ij}M \geq p_i,\ \forall i, j,\ i < j, \tag{3}$$

$$(y_i - y_j) + u_{ij}M + \left(1 - v_{ij}\right)M \geq q_j,\ \forall i, j,\ i < j, \tag{4}$$

$$(y_j - y_i) + \left(1 - u_{ij}\right)M + \left(1 - v_{ij}\right)M \geq q_i,\ \forall i, j,\ i < j, \tag{5}$$

$$T \geq x_i + p_i,\ \forall i, \tag{6}$$

$$x_i \geq k_i,\ \forall i, \tag{7}$$

$$Bn_i \geq y_i + q_i,\ \forall i, \tag{8}$$

$$y_i \geq Bs_i,\ \forall i. \tag{9}$$

The objective function is to minimize the total waiting time of all vessels and the overall completion time of vessel service. Constraints (2)–(5) guarantee that vessels cannot occupy the same berth simultaneously. In Constraint (6), T is the overall completion time of the service for all vessels at the port. Constraint (7) indicates that each vessel is served after arrival. Constraints (8) and (9) ensure that each vessel must be assigned to the appropriate position considering the draft of the vessel.

Since the model is a mixed-integer linear programming problem, it can be globally solved by conventional mixed-integer linear programming techniques to obtain an optimal berth assignment. Although the proposed mixed-integer linear programming model of the BAP can be solved to derive a globally optimal solution, the solution time significantly increases as the number of vessels increases. Consequently, this study develops a GA optimization approach to efficiently obtain a near-optimal solution for large-scale problems.

3. Proposed GA Optimization Approach

The proposed approach combines a genetic algorithm and a berth allocation strategy. The GA is responsible for evolving the chromosomes representing the vessels' scheduling sequence. The berth allocation strategy is a critical process that constructs a berth allocation plan using the vessels' scheduling sequence defined in the previous phase. After reaching the terminal condition of the GA, a near-optimal berth allocation layout is obtained for all vessels. Figure 4 indicates the flow chart of the proposed GA optimization approach.

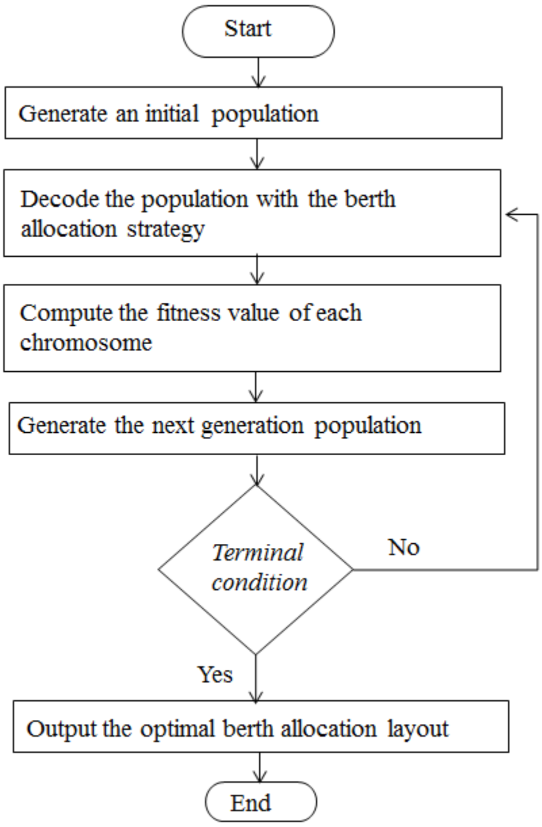
Start
Generate an initial population
Decode the population with the berth allocation strategy
Compute the fitness value of each chromosome
Generate the next generation population
Terminal condition
No
Yes
Output the optimal berth allocation layout
End

Figure 4. Flow chart of the proposed GA optimization approach.

The following describes the processing steps in the flow chart.

Step 1: Generate an initial population

This step is to generate the chromosomes in the first generation. Each chromosome representing a vessel's berthing sequence is made of n genes where n is the number of vessels. The genes of each chromosome of the initial population are randomly generated.

Figure 5 displays a chromosome for solving a nine-vessel scheduling problem. The vessel scheduling sequence is No. 1, No. 2, No. 4, No. 9, No. 5, No. 6, No. 8, No. 3, and No. 7.

1	2	4	9	5	6	8	3	7

Figure 5. Example of a chromosome.

A large population provides a large solution search space, but GA will slow down if there are too many chromosomes. A population size index with the size of a problem is a good solution [21]. After conducting several experiments, the suitable population size for this problem is two times the number of vessels.

Step 2: Decode the population with the berth allocation strategy

This step is to construct a berth allocation plan according to the chromosomes in the population. Jakobs [22] designed the well-known bottom-left algorithm (BLA) integrated with a GA to obtain the packing pattern for solving a packing problem. In BLA, each item is moved as close as possible to the bottom of the object and then as close as possible to the left. Inspired by this idea, the current study develops the berth allocation strategy called the best-position algorithm (BPA). The BPA places each vessel at the best position according to the scheduling sequence, i.e., each vessel selects the bottom berth of suitable berthing positions, cannot overlap with others, and minimizes its waiting time. The detailed algorithm is as follows (Algorithm 1).

Algorithm 1: Best-position algorithm of berth allocation.

```
$V_i$ be the $i$th vessel to be allocated to berths
for $i$ = 1 to $n$ do
  read data of $V_i$
  $x_i = k_i$
  found = no
  repeat
    if ($q_i$ continuous berths in [$Bs_i$, $Bn_i$] are available at time $x_i$) then
      found = yes
    else
      $x_i = x_i + 1$
    end if
  until found = yes
  $y_i$ = starting position of available berths
end
```

The following example demonstrates how to assign berthing locations for vessels. The original data of three vessels, assigned to berths 0 to 20, are shown in Table 1. According to the BPA, the starting positions for vessels No. 1 and No. 2 are (0, 0) and (6, 0), respectively. These positions, (0, 0) and (6, 0), are also the best positions for vessels No. 1 and No. 2 because the waiting time of the two vessels is zero. The starting bottom-left coordinate of vessel No. 3 is (5, 0), as indicated by 1 in Figure 6. The dotted block in Figure 6 illustrates the quay space and service time that will be assigned to vessel No. 3. Since berths have been allocated to vessels No. 1 and No. 2 during the 5th to 11th time units, vessel No. 3 can only be allocated to the unoccupied berths. According to the BPA, positions (5, 14), (6, 0), and (6, 12) are tried for vessel No. 3 sequentially, as indicated by 2, 3, and 4 in Figure 7, respectively. The feasible bottom-left coordinate of vessel No. 3 is (6, 12). The waiting time for vessel No. 3 is 1, calculated from x_i minus k_i. The final assigned berthing positions of three vessels are listed in Table 1.

Table 1. Data of three vessels in the berth allocation problem.

Original Data of Three Vessels						Assigned Berthing Position		
Vessel No	k_i Estimated Arrival Time	p_i Estimated Operation Time	q_i Number of Berths Needed	Bs_i Starting Position of Suitable Berths	Bn_i Ending Position of Suitable Berths	x_i Starting Time	y_i Starting Position	Waiting Time
1	0	6	14	0	20	0	0	0
2	6	8	12	0	20	6	0	0
3	5	6	8	0	20	6	12	1

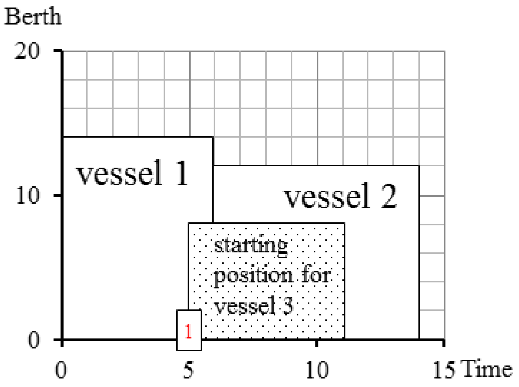

Figure 6. Starting bottom-left coordinate of vessel No. 3 based on BPA.

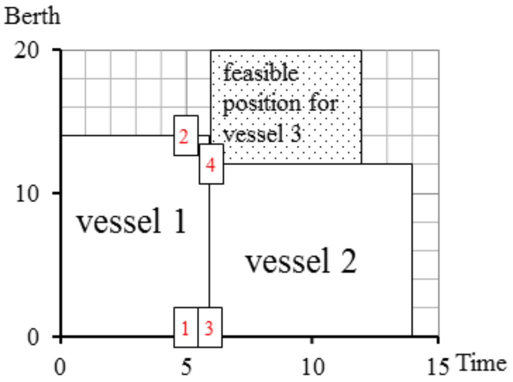

Figure 7. Four feasible locations for vessel No. 3 based on BPA.

Step 3: Compute the fitness value of each chromosome

The fitness value is the sum of the waiting time of each vessel plus the overall completion time of vessel service. A lower fitness value indicates a better berth allocation plan.

$$\text{Fitness} = \sum_{i=1}^{n} (x_i - k_i) + max\{x_i + p_i,\ i = 1, \ldots, n\} \tag{10}$$

According to the data from Table 1, the total waiting time of all vessels is 1. Observing the berth allocation layout of Figure 7, the overall completion time limited by vessel No. 2 is 14. The fitness value is 15.

Step 4: Generate the next-generation population

This step is to adopt an evolutionary strategy that can extract the most desirable features from parents' genes and combine them with new genes to form the next generation. Figure 8 illustrates that the chromosomes of the next-generation population are composed of three parts: part 1 is comprised of the top-performing chromosomes from the previous generation, part 2 consists of chromosomes generated through mutation operations from those in part 1, and part 3 is generated randomly. The following steps show how to form the next-generation population.

Step 4.1: Sort the chromosomes in ascending order of their fitness value.
Step 4.2: Directly transfer a selected percentage (referred to as 'noChroRes' hereafter) of the top-performing chromosomes to the next generation, indicated as part 1 in Figure 8. It is called an elitist strategy [23].
Step 4.3: In the berth allocation process, the layout of front vessels affects how berths are assigned to the rear ones. A better berth allocation plan means the front vessels form a better layout. Therefore, sequentially retrieve each chromosome from part 1 and perform the mutation operation to generate an increasing number of chromosomes in the next generation, as indicated by part 2 in Figure 8. The number of chromosomes generated through mutation option from each chromosome of part 1 is determined by 'noGen'. The mutation operation retains a certain percentage of genes from each chromosome of part 1 (referred to as 'mRate' hereafter), while other genes are randomly generated to form a new chromosome. Figure 9 illustrates the result of the mutation operation on one chromosome representing nine vessels. Assuming "noGen" is 1, it means one mutated chromosome is generated from each chromosome of part 1. With mRate equal to 30% (9 × 30% = 2.7), the mutated chromosome inherits the first and second genes from the parent, represented by '1' and '2' in Figure 9, while genes 3 through 9 are randomly generated. The chromosome after the mutation operation represents a new vessel's berthing sequence. Therefore, when randomly generating the third gene, its value cannot be the same as the first or second gene's value, and similarly, when randomly generating the fourth gene, its value cannot be the same as the first through third genes' values.
Step 4.4: Chromosomes of part 3, indicated in Figure 8, are generated randomly.

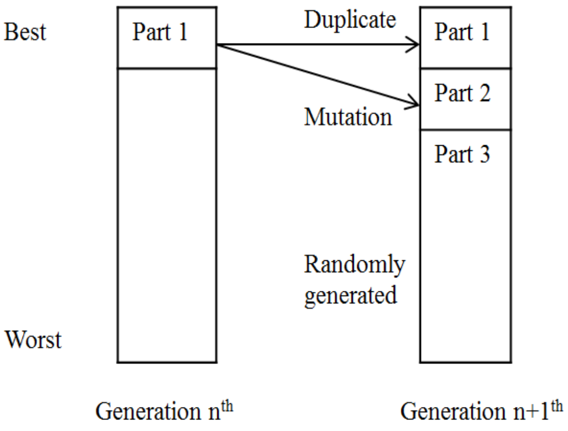

Figure 8. Evolutionary strategy.

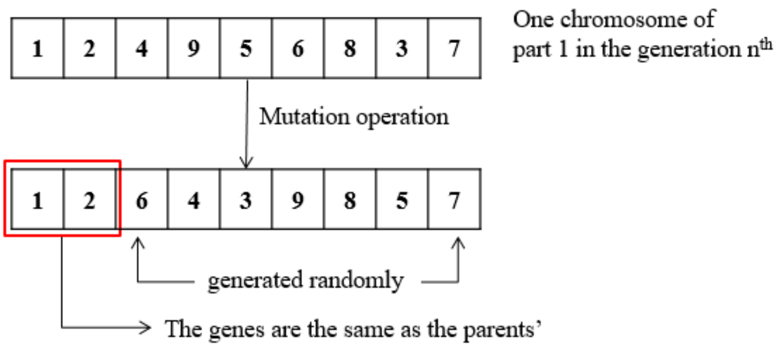
1 2 4 9 5 6 8 3 7
One chromosome of part 1 in the generation n^th
Mutation operation
1 2 6 4 3 9 8 5 7
generated randomly
The genes are the same as the parents'

Figure 9. Result of mutation operation on one chromosome representing nine vessels.

Lin [24] indicated that the same chromosome segments of different chromosomes do not lead to the same rectangle deployment in assortment problems. Therefore, a genetic algorithm with crossover operation for solving assortment problems performs poorly. The current study transforms the BAP into a two-dimensional packing problem, similar to the assortment optimization problem. Hence, crossover operation preserving some chromosome segments seems unnecessary in the BAP because the final layouts of the same chromosome segments depend on the genes before these segments. This study adopts only mutation operation to guarantee population diversity in the genetic algorithm without any crossover operation.

In this study, the operation of the genetic algorithm (GA) is determined by the following three factors. Considering the preservation of highly fit gene combinations with moderate proportions and the introduction of new genes through mutation operations to increase population diversity, the recommended values for these three factors are as follows:

1. The percentage of highly fit chromosomes to retain, denoted as 'noChroRes' in Table 2. Method 1 retains only the best chromosome, while methods 2 through 5 retain 12.5% of highly fit chromosomes.
2. The number of offspring each highly fit chromosome should generate, denoted as 'noGen' in Table 2. Method 1 does not generate offspring from each highly fit chromosome, while methods 2 and 3 generate 2 offspring from each highly fit chromosome, and methods 4 and 5 generate 1 offspring from each highly fit chromosome.
3. The number of genes inherited from the parent chromosome through the mutation operation, denoted as 'mRate' in Table 2. In method 1, no genes are inherited from highly fit chromosomes, while in methods 2 and 4, 50% of genes are inherited, and in methods 3 and 5, 30% of genes are inherited.

Table 2. Data of five methods.

Method	Combination of Parameters			Fitness Value of Fifty Trials of 3000 Evaluations			
	noChroRes	noGen	mRate	Mean	Std. Deviation	Minimum	Maximum
1	The best one	0	0	563.82	11.48	526	582
2	12.5%	2	50%	550.86	18.33	515	581
3	12.5%	2	30%	542.16	16.67	509	565
4	12.5%	1	50%	551.78	17.18	504	580
5	12.5%	1	30%	539.32	16.57	502	573

Proper settings of these factors directly impact the performance and convergence speed of the algorithm and should be determined based on specific problem characteristics and algorithm performance. Methods 1 to 5 in Table 2 consider the combinations of different parameters for these three factors. By conducting experiments and analyses on these five methods listed in Table 2, the most appropriate parameter settings for this study are identified.

One-way ANOVA was conducted to test the significance of the difference among the five methods. However, Levene's test showed that variances of the five methods were unequal (Leven statistics = 3.609, df1 = 4, df2 = 245, significance = 0.007), indicating that the assumption of ANOVA is invalid. Instead, Brown–Forsythe [25] was employed as the robust test of equality of the means. The results showed significant (BF = 17.561, df1 = 4, df2 = 229.155, significance = 0.000), implying at least one mean distinct from others. Because the sample size was over 50, the Games–Howell test was performed for post hoc multiple comparisons. Conclusively, the fitness value of method 1 is the worst. Also, the mean of fitness values in method 3 (mean = 542.16) and method 5 (mean = 539.32) are better than the other three methods.

The results show that preserving the genes from highly fit chromosomes to generate the next generation helps obtain better solutions. The results of fifty trials of 3000 evaluations suggest that noChroRes = 12.5%, noGen = 1 or 2, and mRate = 30% appear to be better combinations than others. According to the results in Table 2, this research adopts the combination of parameters in method 5, which produces a lower mean of fitness value than other methods for the experiments in Section 4. If 81 vessels need to be scheduled, then the population size is 162. Part 1 consists of 20 chromosomes (162 × 12.5%). Part 2 comprises 20 chromosomes generated through mutation operations from those in part 1 (noGen = 1), with each mutated chromosome inheriting 24 genes (81 × 30%) from the parent generation. Part 3 consists of 122 chromosomes (162–20–20).

4. Numerical Examples

Due to the strong heterogeneity of BAP models, making a fair comparison among them seems impossible. Because of the lack of benchmarks involving different types of BAPs [11], this study uses a set of real-world simulated instances to demonstrate the effectiveness and efficiency of the proposed method. First, Examples 1 and 2 are used to evaluate the performance of the GA optimization approach. Then, Example 3 is presented to examine the practicability of the GA optimization approach. Finally, a convergence diagram shows that the GA optimization approach is an effective tool for solving the BAP problem. All the examples were solved on a personal computer with Intel Core i5 M460 @2.53 GHz CPU. The GA optimization approach was implemented with Java programming language, while the deterministic optimization model was solved with LINGO 20.0 (2009).

Example 1 is to assign berthing locations for 27 vessels in four types of quays. The data of quays and vessels are shown in Tables 3 and 4, respectively. Table 3 lists four types of quay that provide specific terminal facilities for loading/unloading in a port. Every quay has limitations on its berths and water depth. Table 4 lists the data of 27 vessels scheduled to arrive within twenty-four hours.

Table 3. Data of four types of quays in the berth allocation problem.

Type of Quay	Passenger and Cargo	General Cargo		Container		Cement Carrier
Serial number of berths	1~20	21~80	81~120	121~150	151~220	221~240
Depth (Meter)	7	6	10	8	13	11

Table 4. Data of 27 vessels in the berth allocation problem.

No	Type of Ship	k_i Estimated Arrival Time	p_i Estimated Operation Time	q_i Number of Berths Needed	d_i Draft	Bs_i Starting Position of Suitable Berths	Bn_i Ending Position of Suitable Berths
1	Container	0	8	55	8	121	220
2	Container	2	6	50	6	121	220
3	Container	15	3	40	8	121	220
4	Container	4	5	62	9	151	220
5	Container	7	10	45	12	151	220
6	Container	11	12	48	7	121	220
7	Container	16	4	24	10	151	220
8	Container	12	11	30	12	151	220
9	Container	4	7	43	10	151	220
10	General Cargo	8	6	17	6	21	120
11	General Cargo	20	3	8	7	81	120
12	General Cargo	9	15	9	4	21	120
13	General Cargo	12	8	16	6	21	120
14	General Cargo	22	8	15	9	81	120
15	General Cargo	19	10	13	9	81	120
16	General Cargo	2	7	5	6	21	120
17	General Cargo	6	13	14	6	21	120
18	General Cargo	3	3	9	7	81	120

Table 4. *Cont.*

No	Type of Ship	k_i Estimated Arrival Time	p_i Estimated Operation Time	q_i Number of Berths Needed	d_i Draft	Bs_i Starting Position of Suitable Berths	Bn_i Ending Position of Suitable Berths
19	General Cargo	13	5	20	7	81	120
20	General Cargo	12	4	16	8	81	120
21	General Cargo	3	7	12	9	81	120
22	General Cargo	8	4	6	6	21	120
23	Cement Carrier	15	9	18	9	221	240
24	Cement Carrier	5	7	12	10	221	240
25	Passenger and Cargo	12	4	15	7	1	20
26	Passenger and Cargo	16	3	11	4	1	20
27	Passenger and Cargo	10	2	10	6	1	20

Example 2 is to assign berthing locations for 54 vessels in four types of quays. The data set in Table 4 is duplicated for Example 2 so that the number of vessels increases from 27 to 54. In Example 2, the estimated operation time (p_i) and the number of berths needed (q_i) are divided by two for all vessels. The estimated arrival time (k_i) of vessels No. 28 to No. 54 is increased by five units.

In order to evaluate the performance of the GA optimization approach, Examples 1 and 2 were solved using the deterministic optimization model and the GA optimization approach, respectively. Table 5 compares the results of the two methods. Solving the deterministic optimization models on LINGO 20.0, the obtained globally optimal values of Examples 1 and 2 are 98 and 36, respectively. Fifty trials of 3000 evaluations were ran by GA optimization approach with noChroRes = 12.5%, noGen = 1 and mRate = 30%. Observing the computational results in Table 5, the best solution of the GA optimization approach is identical to the globally optimal solution obtained from the deterministic optimization model. In addition, the objective values of the global optimal solution and the average solution of the GA optimization approach are very close. Therefore, the results demonstrate that the GA optimization approach provides competitive results in terms of solution quality.

Table 5. Experimental results of Examples 1 and 2.

Example	No of Vessels	Deterministic Optimization Model		GA Optimization Approach		
		Globally Optimal Solution	CPU Time (mm:ss)	Best Solution	Average Solution	Average CPU Time (mm:ss)
1	27	98	00:51	98	98	00:02
2	54	36	07:36	36	36.3	00:12

Example 3 verifies that the GA optimization approach yields near-optimal solutions efficiently for more complicated problems. Example 3 contains 81 vessels that are increased threefold from Table 4. Data of vessels No. 1 to No. 27 are the same as in Table 4. Data of vessels No. 28 to No. 54 and vessels No. 55 to No. 81 are generated from Table 4 where the estimated arrival time (k_i) of each vessel is shifted backward by increasing five and eight units, respectively. Table 6 lists the experimental results of the two methods. The deterministic optimization model involving 81 vessels cannot be solved within two hours by LINGO 20.0. The developed GA optimization approach takes about three minutes to obtain a near-optimal solution with the best objective value of 1324. The GA optimization approach provides a near-optimal solution for allocating berths to vessels if the deterministic optimization model cannot globally solve the BAP within a reasonable amount of computational time. Therefore, the results demonstrate that the GA optimization approach is a more practical method for large-scale problems.

Table 6. Experimental results of Example 3.

Example	No of Vessels	Deterministic Optimization Model		GA Optimization Approach		
		Global Optimal Solution	CPU Time (mm:ss)	Best Solution	Average Solution	Average CPU Time (mm:ss)
3	81	N/A	>120:00	1324	1397.74	02:51

The optimal berth allocation layout of Example 3 is presented as a time-space diagram indicated in Figure 10. Herein, 81 vessels are assigned to the corresponding type of quay and do not overlap with each other.

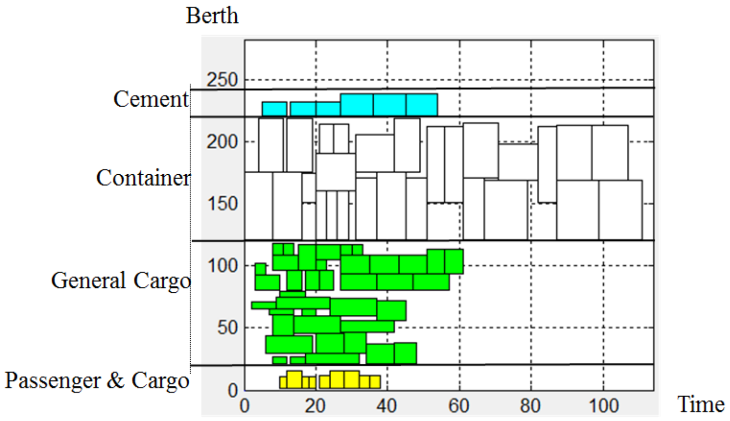

Figure 10. Optimal berth allocation layout for 81 vessels.

Figure 11 illustrates the convergence of the GA optimization approach in 100, 500, 1500, and 3000 evaluations for Example 2. The best solutions to 100, 500, 1500, and 3000 evaluations are identical to the globally optimal value of Example 2, 36. The objective value stabilizes toward the globally optimal solution as the number of evaluations increases. The convergence diagram indicates that the GA optimization approach is an effective tool for solving the BAP.

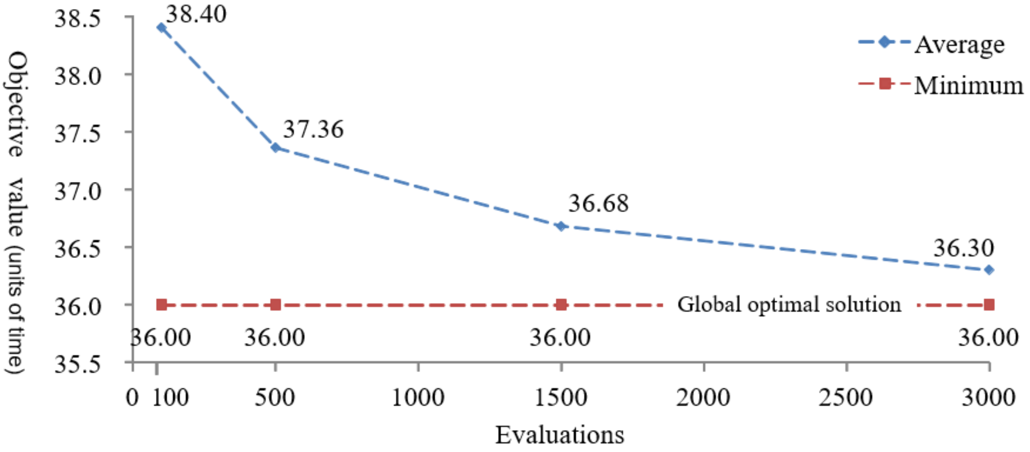

Figure 11. Convergence diagram of the GA optimization approach.

5. Conclusions

This research offers a twofold solution for the port authorities to enhance their competitiveness in the global transportation network. This study focuses on the continuous and dynamic berth allocation problems. The proposed method models the berth allocation problem as a mixed-integer linear programming problem that minimizes the total waiting time of vessels and the overall completion time of vessel service to improve the berth utilization rate. The formulated model can be globally solved by conventional mixed-integer linear programming techniques. For practical scenarios involving a significant number of vessels and complex conditions, a GA optimization approach is developed to provide a near-optimal solution within a reasonable amount of computational time. The experimental results indicate the effectiveness and efficiency of the proposed method. Finally, the results of berth assignment are presented as time-space diagrams, aiding port authorities in making more informed and timely decisions regarding berth allocation.

The nature of seaside operations, encompassing berth allocation problems (BAP), quay crane assignment problems (QCAP), and quay crane scheduling problems (QCSP), is comprehensive. While this study emphasizes time-related factors in berth allocation, port authorities and carriers are also concerned with several related issues, including cost factors and berthing priority. Consequently, more factors should be considered in future research. Additionally, the GA optimization approach can reach a near-optimal solution of the BAP, but it still cannot guarantee the global optimality of the solution. Developing an efficient deterministic method to obtain a globally optimal solution for practical cases with heavy vessel traffic is also worth investigating in the future.

Author Contributions: Conceptualization, J.-F.T. and S.-C.C.; methodology, J.-F.T. and S.-C.C.; software, S.-C.C. and M.-H.L.; validation, J.-F.T. and M.-H.L.; formal analysis, S.-C.C. and M.-H.L.; investigation, S.-C.C. and M.-H.L.; resources, S.-C.C. and M.-H.L.; data curation, S.-C.C. and M.-H.L.; writing—original draft preparation, S.-C.C. and J.-F.T.; writing—review and editing, M.-H.L.; visualization, S.-C.C.; supervision, J.-F.T. and M.-H.L.; project administration, J.-F.T.; funding acquisition, J.-F.T. and M.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Science and Technology Council in Taiwan under Grants NSTC 111-2410-H-027-006, NSTC 112-2410-H-027-005-MY2 and NSTC 111-2410-H-845-012-MY2.

Data Availability Statement: Data are contained within the article.

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

References

1. Kim, K.H.; Moon, K.C. Berth scheduling by simulated annealing. *Transp. Res. Part B Methodol.* **2003**, *37*, 541–560. [CrossRef]
2. Du, Y.; Chen, Q.; Quan, X.; Long, L.; Fung, R.Y.K. Berth allocation considering fuel consumption and vessel emissions. *Transp. Res. Part E Logist. Transp. Rev.* **2011**, *47*, 1021–1037. [CrossRef]
3. Rodrigues, F.; Agra, A. Berth allocation and quay crane assignment/scheduling problem under uncertainty: A survey. *Eur. J. Oper. Res.* **2022**, *303*, 501–524. [CrossRef]
4. Liu, C.; Xiang, X.; Zheng, L. Two decision models for berth allocation problem under uncertainty considering service level. *Flex. Serv. Manuf. J.* **2017**, *29*, 312–344. [CrossRef]
5. Bierwirth, C.; Meisel, F. A survey of berth allocation and quay crane scheduling problems in container terminals. *Eur. J. Oper. Res.* **2010**, *202*, 615–627. [CrossRef]
6. Wang, F.; Lim, A. A stochastic beam search for the berth allocation problem. *Decis. Support Syst.* **2007**, *42*, 2186–2196. [CrossRef]
7. Imai, A.; Sun, X.; Nishimura, E.; Papadimitriou, S. Berth allocation in a container port: Using a continuous location space approach. *Transp. Res. Part B Methodol.* **2005**, *39*, 199–221. [CrossRef]
8. Stahlbock, R.; Voß, S. Operations research at container terminals: A literature update. *OR Spectr.* **2008**, *30*, 1–52. [CrossRef]
9. Xu, Z.; Lee, C.-Y. New lower bound and exact method for the continuous berth allocation problem. *Oper. Res.* **2018**, *66*, 778–798. [CrossRef]
10. Imai, A.; Nishimura, E.; Papadimitriou, S. The dynamic berth allocation problem for a container port. *Transp. Res. Part B Methodol.* **2001**, *35*, 401–417. [CrossRef]
11. Bierwirth, C.; Meisel, F. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *Eur. J. Oper. Res.* **2015**, *244*, 675–689. [CrossRef]

12. Mohammadi, M.; Forghani, K. Solving a stochastic berth allocation problem using a hybrid sequence pair-based simulated annealing algorithm. *Eng. Optim.* **2018**, *51*, 1810–1828. [CrossRef]
13. Liu, C.; Xiang, X.; Zheng, L. A two-stage robust optimization approach for the berth allocation problem under uncertainty. *Flex. Serv. Manuf. J.* **2019**, *32*, 425–452. [CrossRef]
14. Lv, X.; Jin, J.G.; Hu, H. Berth allocation recovery for container transshipment terminals. *Marit. Policy Manag.* **2020**, *47*, 558–574. [CrossRef]
15. Rodrigues, F.; Agra, A. An exact robust approach for the integrated berth allocation and quay crane scheduling problem under uncertain arrival times. *Eur. J. Oper. Res.* **2021**, *295*, 499–516. [CrossRef]
16. Wu, Y.; Miao, L. An efficient procedure for inserting buffers to generate robust berth plans in container terminals. *Discret. Dyn. Nat. Soc.* **2021**, *2021*, 6619538. [CrossRef]
17. Lim, A. The berth planning problem. *Oper. Res. Lett.* **1998**, *22*, 105–110. [CrossRef]
18. Guan, Y.; Cheung, R.K. The berth allocation problem: Models and solution methods. *OR Spectr.* **2004**, *26*, 75–92. [CrossRef]
19. Lee, D.-H.; Chen, J.H.; Cao, J.X. The continuous berth allocation problem: A greedy randomized adaptive search solution. *Transp. Res. Part E Logist. Transp. Rev.* **2010**, *46*, 1017–1029. [CrossRef]
20. Lee, Y.; Chen, C.-Y. An optimization heuristic for the berth scheduling problem. *Eur. J. Oper. Res.* **2009**, *196*, 500–508. [CrossRef]
21. Gonçalves, J.F. A hybrid genetic algorithm-heuristic for a two-dimensional orthogonal packing problem. *Eur. J. Oper. Res.* **2007**, *183*, 1212–1229. [CrossRef]
22. Jakobs, S. On genetic algorithms for the packing of polygons. *Eur. J. Oper. Res.* **1996**, *88*, 165–181. [CrossRef]
23. Golberg, D.E. Genetic algorithms in search, optimization, and machine learning. *Addion Wesley* **1989**, *1989*, 102.
24. Lin, C.-C. A genetic algorithm for solving the two-dimensional assortment problem. *Comput. Ind. Eng.* **2006**, *50*, 175–184. [CrossRef]
25. Brown, M.B.; Forsythe, A.B. Robust tests for the equality of variances. *J. Am. Stat. Assoc.* **1974**, *69*, 364–367. [CrossRef]

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