Md. Mahfuzur Rahman[1], Md Abrar Jahin[1], Md. Saiful Islam[1], M. F. Mridha[2], and Jungpil Shin[3]

[1]Department of Industrial Engineering and Management, Khulna University of Engineering & Technology (KUET), Khulna, 9203, Bangladesh
[2]Department of Computer Science, American International University-Bangladesh (AIUB), Dhaka, 1229, Bangladesh
[3]Department of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu, 965-8580, Japan

April 01, 2024

QCDC-DR-GA: Optimizing Container Loading and Unloading through Dual-Cycling and Dockyard Rehandle Reduction Using a Hybrid Genetic Algorithm

Md. Mahfuzur Rahman, Md Abrar Jahin, Md. Saiful Islam, M. F. Mridha, *Senior Member, IEEE,* and Jungpil Shin, *Senior Member, IEEE*

***Abstract*—This paper addresses the optimization of container unloading and loading operations at ports, integrating quay-crane dual-cycling (QCDC) with dockyard rehandle minimization. We present a unified model encompassing both operations: ship container unloading and loading by quay crane, and the other is reducing dockyard rehandles while loading the ship. We recognize that optimizing one aspect in isolation can lead to suboptimal outcomes due to interdependencies. Specifically, optimizing unloading sequences for minimal operation time may inadvertently increase dockyard rehandles during loading and vice versa. To address this NP-hard problem, we propose a hybrid genetic algorithm (GA) QCDC-DR-GA comprising 1-dimensional and 2-dimensional GA components. Our model, QCDC-DR-GA, consistently outperforms four state-of-the-art methods in maximizing dual cycles and minimizing dockyard rehandles. Compared to those methods, it reduced 15-20% of total operation time for large vessels. Results underscore the inefficiency of separately optimizing QCDC and dockyard rehandles. Fragmented approaches, such as QCDC Scheduling Optimized by bi-level GA and GA-ILSRS (Scenario 2), show limited improvement compared to QCDC-DR-GA. As in GA-ILSRS (Scenario 1), neglecting dual-cycle optimization leads to inferior performance than our proposed QCDC-DR-GA.**

***Index Terms*—Dual Cycling, Quay Crane, Dockyard Rehandles, Genetic Algorithm, 2D Crossover, 2D Mutation**

I. INTRODUCTION

GLOBAL trade relies heavily on efficient port operations, with shipping containers carrying nearly 80% of the world's goods. Therefore, countries are competing to have large fleets. The operation to receive these mega-ships needs preparations. The goal and challenge of every port is now to reduce the turnaround time of vessels. The most expensive single container handling equipment unit and the main operational bottleneck at ports are the Quay Cranes (QCs) [1]. Ports can decrease ship turnaround time, increase productivity, and boost freight transportation system throughput by increasing QC efficiency [2]. Our research addresses this key bottleneck to port productivity. The approach taken here is a low-cost method; neither new infrastructure nor technologies are needed. Although our strategy would not fix the capacity issue in the long term, it can be applied more quickly than other approaches and be used in conjunction with other methods.

Traditionally, ports adopt a *single cycle* approach (see Fig. 1), where QCs handle loading after completing unloading tasks. However, *dual cycling* presents an advanced strategy allowing simultaneous loading and unloading, thereby reducing empty crane moves and potentially decreasing turnaround times significantly [2] (see Fig. 1).

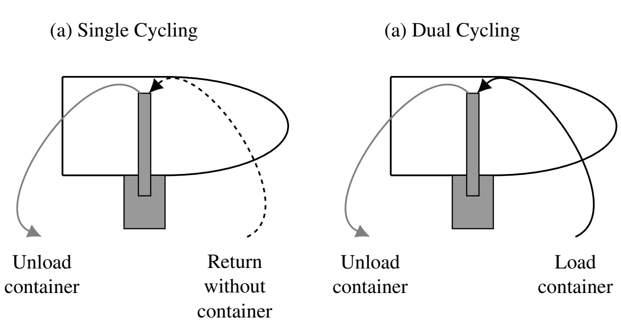


Fig. 1: (a) Unloading Using Single Cycling; (b) Unloading and Loading with Double Cycling

Despite its benefits, maximizing the no. of dual cycles requires careful consideration of factors such as the unloading sequence of stacks within a ship row. Previous studies have proposed greedy, heuristic, and metaheuristic algorithms, including genetic algorithms (GA), to address the NP-hard nature of optimizing quay crane dual-cycling (QCDC) scheduling [3], [4], [5], [6]. Some of the studies focused on overall handling efficiency and the system's stability of container terminals with double cycling and other inbound vehicles of the port [7], [8].

Another issue named *rehandling* of containers arises at the dockyard while loading the ship. Rehandling occurs when the

M. M. Rahman, M. A. Jahin, and M. S. Islam were with the Department of Industrial Engineering and Management, Khulna University of Engineering & Technology (KUET), Khulna, 9203, Bangladesh (e-mail: sm.mahfuz031@gmail.com; abrar.jahin.2652@gmail.com; saifuliem@iem.kuet.ac.bd).

M. F. Mridha is with the Department of Computer Science, American International University-Bangladesh (AIUB), Dhaka, 1229, Bangladesh (e-mail: firoz.mridha@aiub.edu).

J. Shin is with the Department of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu, 965-8580, Japan (e-mail: jpshin@u-aizu.ac.jp).

Manuscript received April 19, 2005; revised August 26, 2015. This work was supported by the Competitive Research Fund of The University of Aizu, Japan. *(Md. Mahfuzur Rahman and Md Abrar Jahin are co-first authors.) (Corresponding author: Jungpil Shin.)*

target container is not on the top of the stack. So, minimizing rehandling in the dockyard, where containers are moved for retrieval or rearrangement, is crucial for efficiency. Numerous works have addressed this issue by creating models and developing solving approaches [9], [10], [11]. This NP-hard problem is also tackled effectively using GA [12].

Our work builds upon existing research by integrating maximization of the no. of dual cycles and minimization of the no. of dockyard rehandles into a unified model. We introduce the Maximizing Quay Crane Dual Cycles and Minimizing Dockyard Rehandles by GA (QCDC-DR-GA) method to solve this model efficiently.

This study presents seven significant contributions:

1) Empirical validation of the correlation between unloading sequence and dockyard rehandles.
2) Development of a comprehensive model integrating dockyard and QCDC operations.
3) Introduction of a novel hybrid GA approach tailored to container handling optimization.
4) Proposal of specialized GA techniques to address unique challenges.
5) Extensive analysis of computational parameters within the GA framework.
6) Rigorous benchmarking analysis against four state-of-the-art algorithms, demonstrating superior performance and reliability.
7) Statistical validation of the significant performance of QCDC-DR-GA using a two-tailed paired t-test.

This article is structured in the following manner: In section "Problem Description", we discuss the problem statement. The "Methodology" section covers model formulation (objectives and constraints) and our approach, QCDC-DR-GA, including its workflow, strategies, and parameters. The "Results" section details scenario generation, computational experiments, and result analysis. Finally, the "Conclusions" section summarizes the work, highlights primary contributions, and suggests future directions.

II. Problem Description

Ship or port yard container is arranged in a three-dimensional matrix of rows, bays, and tiers (see Fig. 2). Containers are stacked in rows, with each row spanning the width of the bay or ship. The operating cycle of a Quay Crane (QC) involves (a) Locking and unlocking the trolley with the container, (b) Horizontal movement of the trolley (with container), and (c) Vertical movement of the trolley (with container).

A. Case Consideration

Upon the arrival of a vessel at the port, with containers to be unloaded and a loading plan for other containers, let U_c and L_c represent the numbers of containers to be unloaded and loaded, respectively, for each stack. Fig. 3 illustrates an example used in this work. Let S be the set of stacks in a row. $|s| = N$ denotes the number of stacks in set S, and P is a permutation of set S indicating the order of stack handling. The sequence in which stacks within each row are handled

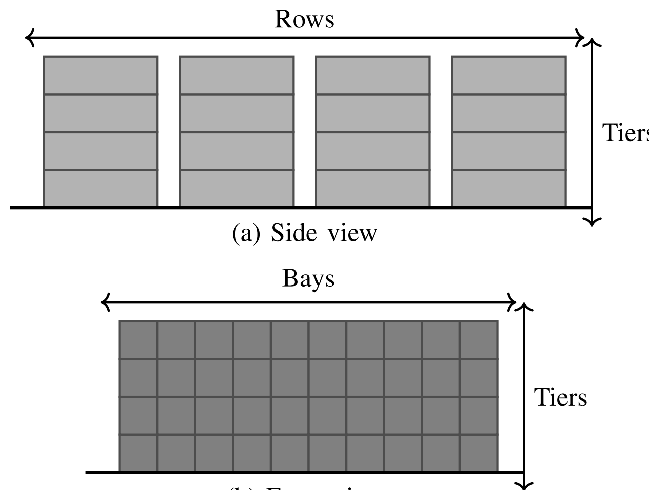


Fig. 2: Illustration of the container arrangement at dockyard or ship. The rows, bays, and tiers are the three axes of the container storing system, where (a) and (b) are the top view and front view, respectively.

affects the total number of cycles, as explained by Goodchild (2006) [2].

1) Generic Double Cycling Method:

(i) Select any unloading permutation, P'. Unload containers stack by stack.
(ii) Select a loading permutation, P, and load stacks according to it.

2) Number of Rehandles in Dockyard: We integrate the nearest lowest stack strategy to address rehandles, as applying the lowest stack strategy alone can be challenging in real-life scenarios. The example in Fig. 3 (b) illustrates a scenario with 3 rehandles under this strategy.

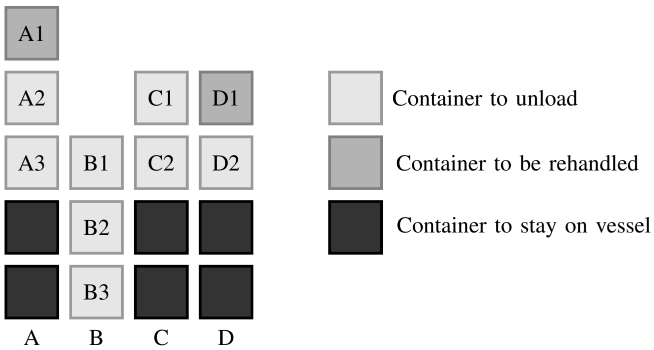
III. Methodology

A. Mathematical Model

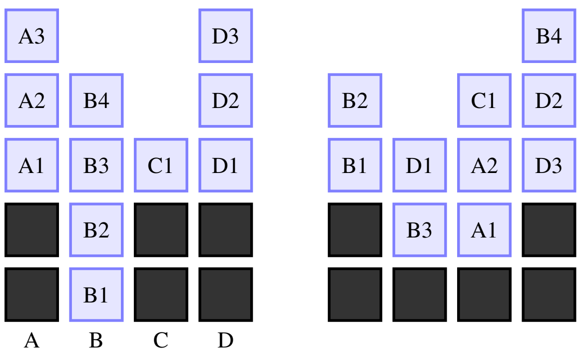
The QCDC problem is modeled as a two-machine flow shop problem.

1) Assumptions: We make the following assumptions:

(i) Containers at the dockside are prepared for loading as needed.
(ii) Unloaded containers are promptly removed from the area and stored appropriately.
(iii) Rehandles of containers on the ship are counted during both the unloading and loading processes.
(iv) Rehandles on the ship are returned to the same stack from which they were taken.
(v) Rehandles on the ship are considered to move between the vessel and the apron; however, in reality, some may only move between stacks on the vessel.
(vi) Rehandles in the dockyard are prioritized using the nearest stack strategy, placing containers on the nearest lowest stack.
(vii) The turnaround time of a vessel, indicative of QC efficiency, is measured by minimizing the total number of



(a) Unloading-plan of a vessel



Ship Loading Sequence Dockyard container's status

(b) Loading plan of the vessel

Fig. 3: Illustration of unloading and loading plan of a ship row.

single (w_s) and dual cycles (w_d) required for unloading and loading.

(viii) Unloading and loading of one row are completed before shifting the crane lengthwise along the ship to the next row. Due to constraints with lateral movement, double cycling across two rows is not feasible.

(ix) No interruptions occur due to inbound vehicles or cranes.

2) Symbols and Decision Variables: The notations are as follows:
m: Bay of containers in the yard
n: Stack of containers in the yard
o: Tier of containers in the yard
U_c: Number of containers to unload in stack $c \in S$
L_c: Number of containers to load in stack $c \in S$
TU_c: Completion time of unloading $c \in S$
TL_c: Completion time of loading $c \in S$
T: Total completion time of unloading loading
R: Number of rehandles of a row in the dockyard
α: Average completion time of a single cycle
β: Average completion time of a double cycle
γ: Average time it takes to tackle a rehandle at the dockyard
μ: Large value
H_{mn}: Highest tier of the yard bay m and stack n
h_{mn}: Height of the yard-bay m and stack n

The decision variables are as follows:
X_{ij}: binary variable for the sequence of unloading jobs (1 if $j \in S$ is loaded after $i \in S$ and 0 otherwise)
Y_{ij}: binary variable for the sequence of loading jobs (1 if $j \in S$ is loaded after $i \in S$ and 0 otherwise)
x_{rmno}: Equals to 1 if the container (m, n, o) is loaded onto the ship-bay and 0 otherwise.

3) Model Establishment: The objective is to minimize the maximum completion time of all jobs while adhering to constraints. The completion time T depends on w and R, given by $T = \alpha w_s + \beta w_b + \gamma R$.

$$minimize, \quad T_{max} \tag{1}$$

subject to,

$$TL_c - TU_c \geq L_c \quad \forall c \in S \tag{2}$$

$$TU_i - TU_j + \mu X_{ij} \geq U_i \quad \forall j, i \in S \tag{3}$$

$$TU_j - TU_k + \mu(1 - X_{ij}) \geq U_j \quad \forall j, i \in S \tag{4}$$

$$TL_i - TL_j + \mu Y_{ij} \geq L_i \quad \forall j, i \in S \tag{5}$$

$$TL_j - TL_i + \mu(1 - Y_{ij}) \geq L_j \quad \forall j, i \in S \tag{6}$$

$$TU_c \geq U_c \quad \forall c \in S \tag{7}$$

$$h_{mn} \leq H_{mn} \tag{8}$$

$$X_{ij} \in 1, 0 \quad \forall j, i \in S \tag{9}$$

$$Y_{ij} \in 1, 0 \quad \forall j, i \in S \tag{10}$$

$$x_{rmno} \in 1, 0 \tag{11}$$

These constraints fully define the model. Constraint (2) ensures a stack is only loaded after necessary unloading. Constraints (3), (4), (9), and (10) sequence unloading stacks and ensure adequate time between them. Constraints (5), (6), (9), and (10) do the same for loading. Constraint (7) ensures sufficient time for unloading. Constraint (8) limits stack height. Constraint (11) enforces binary conditions on flow variables.

B. QCDC-DR-GA

The paper addresses the optimization of unloading sequences and dockyard arrangements for container ships. The problem complexity is defined by $S!$ and $N!$, representing the permutations of stacks and containers, respectively. The aim is to maximize dual cycles during unloading and minimize rehandles in the dockyard, resulting in a complexity of $(S! \times N!)$.

The genetic algorithm (GA) is employed as a metaheuristic approach to finding high-quality solutions. With crossover and mutation, a mixed GA is developed to handle both one-dimensional (1-D) unloading sequences and two-dimensional (2-D) dockyard plans. Key challenges include fitness calculation and integration of unloading sequences and dockyard arrangements. The operating flow path of the QCDC-DR-GA is illustrated in Fig. 4.

1) Set Initial Population: The initial population (P) comprises chromosomes representing unload sequences and dockyard plans. Notations include:
P = population of chromosomes
n = the number of chromosomes in P

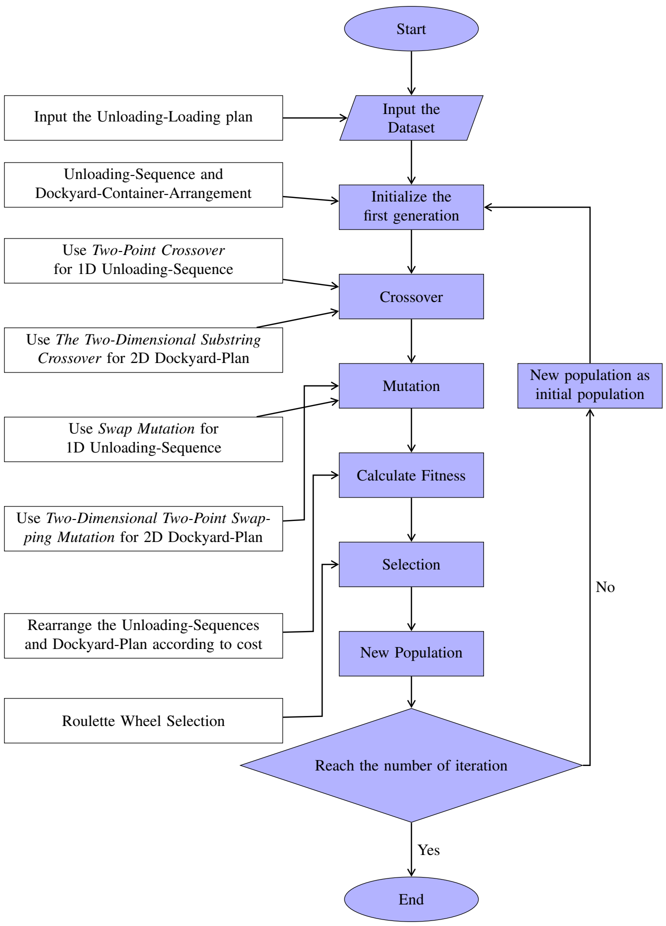


Fig. 4: Methodological flowchart of the proposed QCDC-DR-GA.

c_i = the i^{th} chromosome in P, where $1 \leq i \leq n$
$c_{i(us)}$ = part of chromosome representing unloading sequence
$c_{i(dp)}$ = part of chromosome representing dockyard plan

$$A_1 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix};$$

$$A_2 = \begin{bmatrix} 3A & 3B & 1C & 1D \\ 1A & 2B & & \\ 2A & 1E & 1B & \\ 3C & 2C & 3E & 3D \\ 2D & 2E & 4A & 4D \\ 4B & 4C & 3C & \\ 4E & & & \end{bmatrix};$$

The solution vectors A_1 and A_2 denote the unload sequence and dockyard plan, respectively. Each row in A_2 represents a dockyard stack, with elements indicating the position of containers on ships. For instance, $3A$ denotes a container on the third stack, the first position of the selected ship row. A_1 and A_2 correspond to parts of $c_{i(us)}$ and $c_{i(dp)}$.

2) Crossover: We employ two different methods for crossover to handle the 1D and 2D parts of each chromosome.

One Dimensional Crossover: We employ the *Two-Point Crossover* method, a special case of *N-Point Crossover*. Two random points on the individual chromosomes are selected, and genetic material is exchanged between these points. Common genes are retained, and any dropped genes are appended to the chromosomes (see Fig. 5).

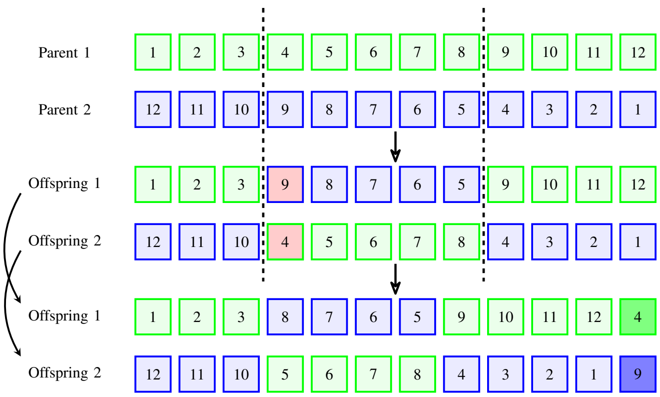


Fig. 5: 1D *Two-Point Crossover* technique.

Two Dimensional Crossover: For the 2D vector, we use the *2D Substring Crossover* method. This involves row swap and column swap operations, which is a modified version of the 2D crossover introduced in the aircraft scheduling problem [13].

Row-wise operation: Two random points are selected, and the entire row of the parents between these points is swapped.

Column-wise operation: The column-wise operation is performed on the selected rows using the *Two-Point Crossover* method previously used for 1D vector crossover.

Repeated items are removed, and any dropped-out items are appended to the offspring (see Fig. 6).

3) Mutation: Mutation, akin to biological mutation, maintains genetic diversity between generations. We employed two methods for distinct chromosome parts. The mutation probability is denoted by P_m, and it occurs probabilistically, governed by algorithm 1.

Algorithm 1: Mutation Algorithm

Input: Two 1D vectors after newly crossed child
Output: Two 2D vectors as mutated child
Generate a random number R.
if $R > P_m$ **then**
  Do not do the mutation operation.
else
  Do mutation operation.

One Dimensional Mutation: We utilize the *Swap Mutation* method for the 1D chromosome part, interchanging two selected genes after crossover (see Fig. 7).

Two Dimensional Mutation: Here, we chose the *2D Two-Point Swapping Mutation* method for the 2D part of our chromosome. This is also the modified version of the mutation method introduced by Tsai et al. [13]. The method is described in algorithm 2 (also see Fig. 8).

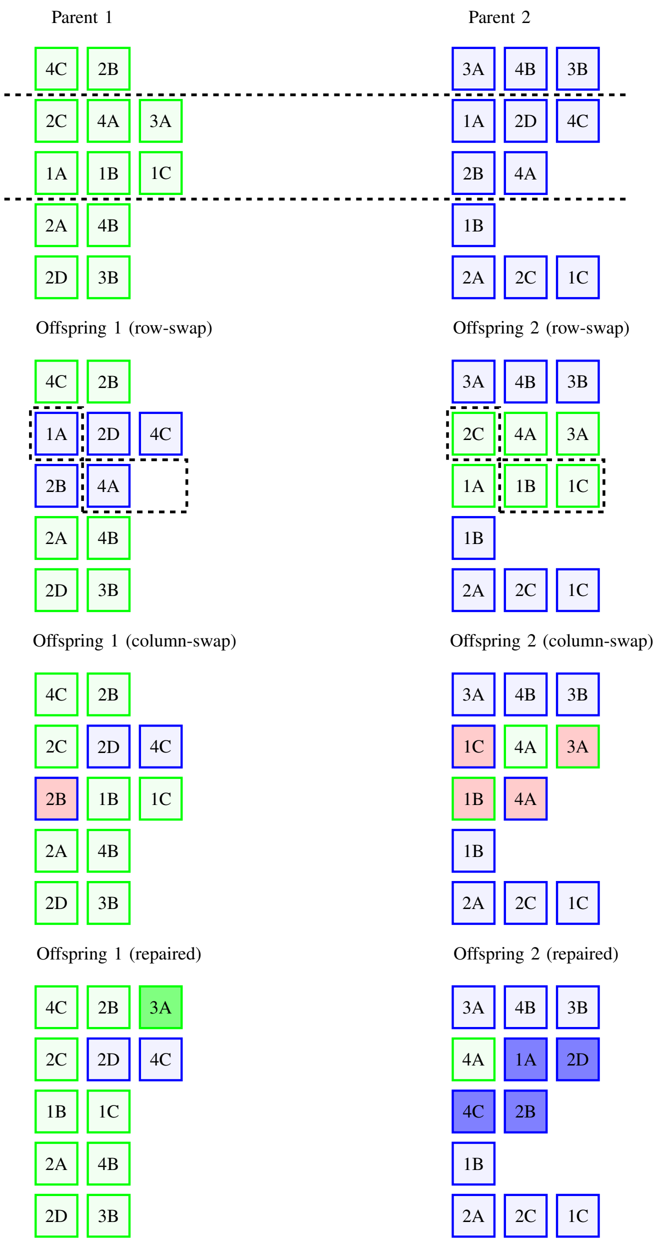


Fig. 6: 2D Two-Point Substring Crossover technique.

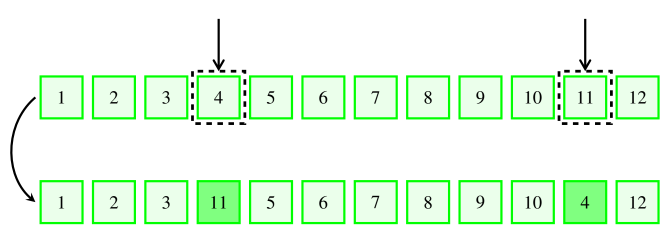


Fig. 7: Swap mutation for 1D vector.

Notations:
R: the number of rows in the 2D chromosome part.
C_{R_i}: the number of columns in the i^{th} row.

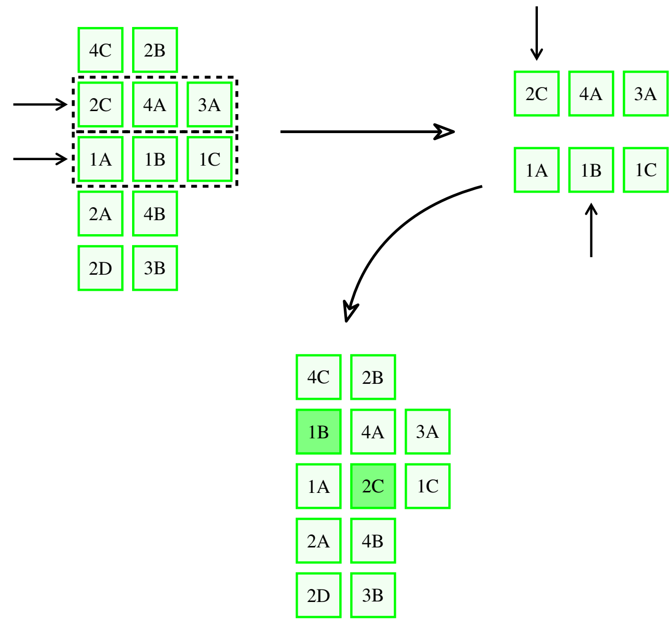


Fig. 8: 2D two-point swapping mutation.

Algorithm 2: 2D Mutation Algorithm

Input: A 2D vector from newly crossed children
Output: A 2D vector as a mutated child

1. Randomly generate $r1$ and $r2$ to select two rows from the 2D vector, where $1 \leq r1, r2 \leq R$
2. Generate random integers $c1$ and $c2$ to select two points from the selected rows, where $1 \leq c1 \leq Cr1$ and $1 \leq c2 \leq Cr2$
3. Interchange the genes between the selected points of the 2D vector

---

4) Calculate Fitness: Chromosomes are evaluated based on their completion time, which is our objective function. The cost, representing the total completion time, is computed for each chromosome in every generation. This cost is then used to select the fittest chromosomes for the next generation. Details of the cost calculation are provided in algorithms 3 and 4.

5) Selection: We employ the *Roulette Wheel* selection technique, a probabilistic method favoring individuals with higher fitness. Unlike traditional roulette, our approach employs weighted probabilities based on fitness (Fig. 9). Notations used include P_E for the percentage of elite chromosomes and E_{rw} denoting the end value of the roulette wheel. The elite class, representing the fittest individuals, automatically advances to the next generation (with P_E set at 20%).

Algorithm 5 outlines the steps of the roulette wheel selection process.

6) Termination: The termination condition of the GA determines when the run ends. Initially, the GA progresses quickly, yielding better solutions every few iterations. However, this progress tends to slow down later, with minimal improvements. To guarantee that our solution approaches optimality, we establish a termination condition as follows: g_i denotes the i^{th} generation, G represents the maximum number of generations, and N_s stands for the number of successive

Algorithm 3: Cost function

Input: Loading plan, unloading plan, dockyard container arrangement, maximum dockyard container stack height
Output: No. of single cycles, no. of double cycles, no. of dockyard rehandles

Function `unload_first_stack`(*unloading plan, unloading sequence*) **:**
  for *container* $\in$ *the dockyard stack of unloadingSequence* **do**
    if *the container will not stay on the vessel* **then**
      unload the container
      no_of_single_cycles += 1

Function `calculate_rehandles`(*target container*) **:**
  Let, no_of_rehandles $\leftarrow 0$
  Let, found_the_container $\leftarrow false$
  for $i \in$ *stacks of dockyard* **do**
    for $j \in$ *containers of current stack* **do**
      if $j =$ *target container* **then**
        found_the_container $\leftarrow true$
        while *until containers are shifted from the top of the target container one by one* **do**
          no_of_rehandles $+ = 1$
          Shift the container nearest lowest stack
        return *no_of_rehandles*
  if *found_the_container* $= false$ **then**
    Warning! container not found
  return *0*

Function `loading_operation`(*unloading plan, loading plan, unloading sequence*) **:**
  if *the current loading stack is empty* **then**
    if *the current unloading stack is empty* **then**
      go to the next loading stack
    else
      return *false, 0*
  Load the current container from dockyard
  return *true,* `calculate_rehandles`(*current container to be loaded at dockyard*)

Let, no_of_single_cycles $\leftarrow 0$
Let, no_of_double_cycles $\leftarrow 0$
Let, no_of_rehandles $\leftarrow 0$
`unload_first_stack`(*unloading plan, unloading sequence*)

Algorithm 4: Cost function (continued)

while *until all the stacks are unloaded from ship* **do**
  for *container* $\in$ *current stack* **do**
    if *the container will not stay on the vessel* **then**
      unload the container
      if *there is any container to load* ***and*** *any stack of the ship is free for loading* **then**
        flag, rehandles $\leftarrow$ `loading_operation`(*unloadig plan, loading plan, unloading sequence*)
        no_of_rehandles += rehandles
        no_of_dual_cycles += 1

while *complete loading the remaining stacks* **do**
  flag, rehandles $\leftarrow$ `loading_operation`(*unloadig plan, loading plan, unloading sequence*)
  no_of_rehandles += rehandles
  no_of_single_cycles += 1

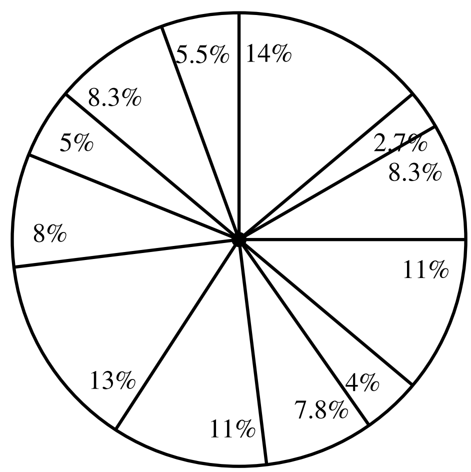


Fig. 9: Weighted roulette wheel.

Algorithm 5: Roulette Wheel Selection Algorithm

Input: Probability against the fitness value of each chromosome
Output: A selected chromosome

Define a 1D vector RW of size n for storing the fitness value of each chromosome. The fitness value is stored as a *cumulative sum* order where E_{rw} is the total sum of all fitness.
for $i \leftarrow 1$ **to** n **do**
  Generate a random number r, where $0 \leq r \leq E_{rw}$.
  Select a chromosome as a parent for crossover.

generations where the fittest chromosome incurs the same cost. The genetic algorithm (GA) execution concludes according to the criteria specified in Algorithm 6.

Algorithm 6: GA termination algorithm

```
Input: Number of successive generations in which the
       cost of the fittest chromosome is the same and
       iteration number
Output: Boolean value to take termination decision
N_s ← Number of successive generations in which the
  fittest chromosome costs the same.
if g_i = G or N_s = 100 then
  | Terminate the GA run.
else
  | Continue
```

7) Parameters: The GA control parameters are shown in Table I. The parameters that best fit our model, such as population size, crossover technique, elite percentage, mutation probability, selection method, etc., are selected. As the solution to our problem is a smooth landscape type and the complexity of our problem is medium, we selected these parameters to fit the situation.

TABLE I: GA control parameters

Parameter	**Value**
Population size	200
1D crossover strategy	Two-Point Crossover
2D crossover strategy	2D Substring Crossover
Crossover rate	0.80
1D mutation strategy	Swap Mutation
2D mutation strategy	2D Two-Point Swapping Mutation
Mutation rate	0.30
Selection strategy	Roulette wheel
Elite class	0.20
Consecutive iterations	100

IV. Results and Discussion

This section addresses the magnitude of *QCDC-DR-GA*. We offer tools to translate cycle-based benefits into time equivalents and validate those estimates against real-world double-cycling data. With an eye on the present and future, we analyze the financial impact of double cycling, estimating potential rewards for both existing vessels and those gracing the waves in the years ahead. The results of the experiments were obtained using a computer with 8 gigabytes of RAM, an Ubuntu 22.04 operating system, and an Intel Core i5 8th Gen. The algorithm was implemented using Python libraries- Pandas and NumPy.

A. Performance Comparisons of the Algorithms

We compared our QCDC-DR-GA algorithm with three established methods:

(a) *Dual-Cycling Greedy Upper Bound Approach (Greedy UB):* This heuristic sorts container stacks in descending order for dual-cycle loading/unloading without considering dockside rehandles [2].

(b) *Mixed-Integer Programming Model for QDCS (bi-level GA):* This method improves upon the Greedy UB by integrating QDCS optimization within a bi-level genetic algorithm framework [5].

(c) *GA-ILSRS:* Explored in two scenarios, this approach optimizes dockyard rehandles using a genetic algorithm combined with Iterated Local Search, neglecting loading/unloading in one scenario and focusing solely on dockyard rehandling in the other [12].

While the Greedy-upper-bound focuses solely on dual cycling, the QDCS-bilevel GA enhances it by incorporating QDCS optimization. Conversely, the GA-ILSRS solely optimizes dockyard rehandles. In contrast, our QCDC-DR-GA considers both dual cycling and dockyard rehandles, optimizing them using a sophisticated genetic algorithm approach.

B. Datasets

Six scenarios were created with varying numbers of stacks (5 to 30) and maximum stack heights (4 to 10) for container rows, reflecting typical container ship characteristics. Loading and unloading plans for each scenario, along with dockyard container arrangements, were generated by the program. Configuration details for the six datasets are summarized in Table II. Sample unloading and loading plans for a small ship are presented in Tables III and IV, respectively.

TABLE II: Loading-unloading plan configuration

Scenario	**No. of stacks**	**Maximum stack height**
1	30	10
2	25	10
3	20	10
4	15	8
5	10	5
6	5	4

TABLE III: Unloading plan of a ship's row

Stack No. \ Tier No.	**1**	**2**	**3**	**4**	**5**
1	1A	1B	1C	1D	1E
2	2A	2B	2C		
3	F	3A	3B	3C	3D
4	4A	4B			
5	5A	5B	5C	5D	5E
6	F	6A	6B	6C	6D
7	F	7A	7B	7C	7D
8	F	8A	8B	8C	8D
9	F	9A	9B	9C	9D
10	F	10A	10B	10C	10D

Tables III and IV provide container location information. For instance, $1B$ indicates a container located in the 1st stack, the 2nd tier of the row, with F indicating the container

TABLE IV: Loading plan of a ship's row

Stack No. \ Tier No.	1	2	3	4	5
1	1A	1B	1C	1D	1E
2	2A	2B	2C	1D	2E
3	3A	3B			
4	4A	4B	4C	4D	4E
5	5A	5B	5C	5D	5E
6	6A	6B			
7	7A	7B	7C	7D	
8	8A				
9	9A	9B	9C	9D	
10	10A	10B	10C	10D	

remaining on the ship. Assuming a maximum stack height of 6, the dockyard plan is discussed further in Section III-B1 as A_2. Detailed generated data is available as supplementary material at "https://dx.doi.org/10.21227/cj08-qn62" due to spatial constraints.

C. Numerical Tests

The six scenarios detailed in subsection IV-B serve as the basis for numerical testing. Processing times for QCs in single and dual cycling, as reported by Goodchild [2], are 90 and 170 seconds, respectively. Container rehandling time by a gantry crane at the dockyard follows a uniform distribution of 60 seconds.

1) Test Results: We thoroughly assess the QCDC-DR-GA algorithm's performance, comparing it with four established methods for port container handling optimization. The evaluation utilizes the six datasets, representing various scenarios with differing container numbers and ship configurations. Table V presents simulation results from each method on these datasets. Additionally, Fig. 10 visually juxtaposes QCDC-DR-GA's performance with that of other approaches.

2) Key Findings: The key findings and remarks of the simulation are as follows:

- The proposed QCDC-DR-GA model consistently outperforms other methods by maximizing dual cycles and minimizing container handling. This demonstrates its effectiveness in optimizing the total unloading-loading time.
- Combining QCDC optimization with dockyard rehandle minimization in QCDC-DR-GA yields superior results compared to fragmented approaches like QCDC Scheduling Optimized by bi-level GA and GA-ILSRS (Scenario 2).
- Neglecting dual-cycling in QC operation optimization, as seen in GA-ILSRS (Scenario 1), leads to inferior performance compared to QCDC-DR-GA. This underscores the importance of simultaneous consideration of both aspects for optimal resource utilization.

D. Significance Test

A two-tailed paired t-test compared the operation times given by the QCDC-DR-GA strategy with others for the numerous datasets.

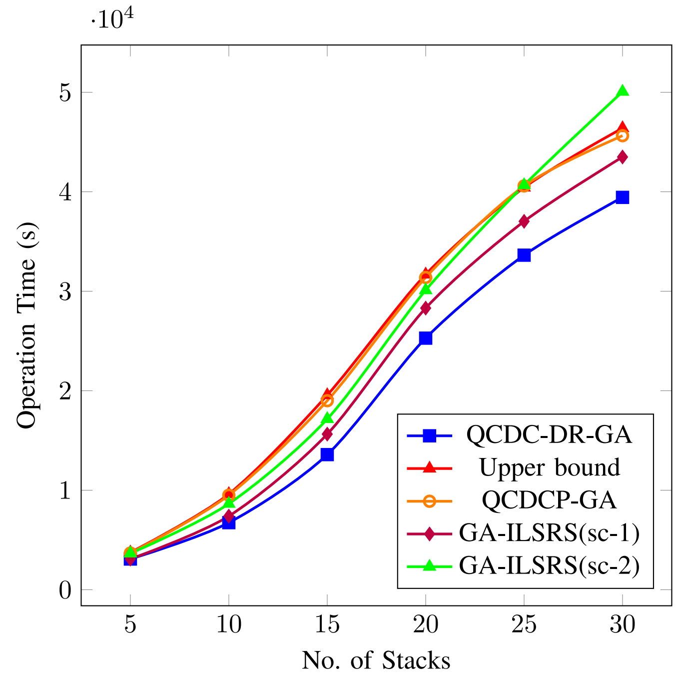


Fig. 10: Comparative performance of QCDC-DR-GA against the state-of-the-art algorithms

- *Null hypothesis*: No significant difference in operation time between QCDC-DR-GA and the compared strategy.
- *Alternative hypothesis*: Significant difference in operation time between QCDC-DR-GA and the compared strategy.

Using a significance level (α) of 0.05, the t-statistic and p-value were computed for each pair of strategies. The results and minimum, maximum, mean, and standard deviation of operation times are summarized in Table V.

V. Conclusions

Utilizing a heuristic approach, we proposed a hybrid QCDC-DR-GA algorithm that optimizes ship unloading and loading processes using dual cycling and reduces dockyard rehandles. Our model consistently outperforms existing methods across the six scenarios of datasets that we created from small to large, with particularly notable improvements for large vessels. However, certain limitations warrant consideration. The model assumes immediate loading container availability at the dockside, potentially neglecting pre-staging requirements. Furthermore, it focuses solely on rehandling between the ship and the apron, overlooking potential relocations within ship stacks. Lastly, disruptions from inbound vehicles or cranes are not accounted for, possibly leading to underestimating operational variability. Future research avenues could address these limitations by incorporating pre-staging needs, exploring intra-ship rehandle optimization, and integrating dynamic disruptions to enhance practical applicability.

References

[1] C.-Y. Chu and W.-C. Huang, "Aggregates cranes handling capacity of container terminals: the port of kaohsiung," *Maritime Policy & Management*, vol. 29, no. 4, pp. 341–350, 2002.

TABLE V: Two-tailed paired t-test results of the proposed QCDC-DR-GA against the other strategies

Scenarios	No. of stacks	Maximum stack height	Strategies	Operation time (min)							$t_{19}(0.05)$ = **2.093**	Improvement of QCDC-DR-GA (%)
				Min	Max	Mean	Standard deviation	Pearson correlation coefficient (r)	T-statistic	P-value	Significance	
1	30	10	Proposed QCDC-DR-GA	615.25	704.92	667.62	25.29	-	-	-	-	-
			Greedy Upper Bound	718.58	825.92	780.33	35.04	-0.16317	10.58051	2.10E-09	Yes	14.44%
			QCDC-bi-level-GA	714.92	813.25	763.00	31.26	-0.24243	9.298231	1.68E-08	Yes	12.50%
			GA-ILSRS-Scenario-1	669.25	764.25	708.95	28.18	0.077286	4.952353	8.84E-05	Yes	5.83%
			GA-ILSRS-Scenario-2	820.75	851.25	835.09	9.59	-0.25862	24.9385	5.57E-16	Yes	20.05%
2	25	10	Proposed QCDC-DR-GA	522.42	604.75	559.02	27.24	-	-	-	-	-
			Greedy Upper Bound	645.33	710.75	674.20	20.93	-0.00945	14.5489	9.41E-12	Yes	17.08%
			QCDC-bi-level-GA	652.17	722.75	672.10	22.52	0.450328	18.67514	1.10E-13	Yes	16.82%
			GA-ILSRS-Scenario-1	535.42	631.08	579.37	28.28	0.075218	2.349253	0.029778	Yes	3.51%
			GA-ILSRS-Scenario-2	666.00	691.00	680.79	8.11	-0.26691	17.44741	3.75E-13	Yes	17.89%
3	20	10	Proposed QCDC-DR-GA	381.42	452.08	417.35	22.51	-	-	-	-	-
			Greedy Upper Bound	491.08	563.08	523.60	19.66	-0.37587	13.22962	4.90E-11	Yes	20.29%
			QCDC-bi-level-GA	484.08	563.08	519.85	25.76	-0.33001	11.33876	6.71E-10	Yes	19.72%
			GA-ILSRS-Scenario-1	397.42	465.75	425.77	19.96	0.259185	1.415272	0.173171	Yes	1.98%
			GA-ILSRS-Scenario-2	492.25	511.50	502.76	4.98	-0.13529	15.71027	2.43E-12	Yes	16.99%
4	15	8	Proposed QCDC-DR-GA	201.67	259.00	229.01	18.85	-	-	-	-	-
			Greedy Upper Bound	295.33	358.67	327.83	18.90	-0.29172	14.19992	1.44E-11	Yes	30.14%
			QCDC-bi-level-GA	295.50	349.33	323.00	17.54	0.013371	16.01655	1.73E-12	Yes	29.10%
			GA-ILSRS-Scenario-1	211.33	277.67	248.65	17.79	0.510959	4.719124	0.000149	Yes	7.90%
			GA-ILSRS-Scenario-2	279.00	295.00	286.56	5.52	-0.12894	12.34932	1.59E-10	Yes	20.08%
5	10	5	Proposed QCDC-DR-GA	89.50	138.67	119.40	16.39	-	-	-	-	-
			Greedy Upper Bound	138.50	186.67	164.44	14.66	-0.09791	8.523389	6.46E-08	Yes	27.39%
			QCDC-bi-level-GA	147.00	183.67	162.13	12.50	-0.51102	7.397294	5.26E-07	Yes	26.36%
			GA-ILSRS-Scenario-1	97.00	147.67	123.13	15.56	-0.06245	0.69881	0.493137	Yes	3.03%
			GA-ILSRS-Scenario-2	137.00	152.00	145.76	4.52	0.294877	7.337783	5.90E-07	Yes	18.09%
6	5	4	Proposed QCDC-DR-GA	37.67	70.42	55.43	10.75	-	-	-	-	-
			Greedy Upper Bound	48.67	81.42	63.82	12.87	0.061497	2.250737	0.036439	Yes	13.15%
			QCDC-bi-level-GA	47.67	80.42	65.60	10.75	0.064153	3.015466	0.007114	Yes	15.51%
			GA-ILSRS-Scenario-1	37.67	70.42	54.48	10.78	-0.25326	-0.24295	0.810645	No	-1.74%
			GA-ILSRS-Scenario-2	55.50	66.00	60.05	3.05	0.352321	1.998582	0.060167	Yes	7.70%

[2] A. V. Goodchild and C. F. Daganzo, “Double-cycling strategies for container ships and their effect on ship loading and unloading operations,” *Transportation science*, vol. 40, no. 4, pp. 473–483, 2006.

[3] Y.-M. Fu, A. Diabat, and I.-T. Tsai, “A multi-vessel quay crane assignment and scheduling problem: Formulation and heuristic solution approach,” *Expert Systems with Applications*, vol. 41, no. 15, pp. 6959–6965, 2014.

[4] A. Diabat and E. Theodorou, “An integrated quay crane assignment and scheduling problem,” *Computers & Industrial Engineering*, vol. 73, pp. 115–123, 2014.

[5] Q. Zeng, A. Diabat, and Q. Zhang, “A simulation optimization approach for solving the dual-cycling problem in container terminals,” *Maritime Policy & Management*, vol. 42, no. 8, pp. 806–826, 2015.

[6] X. Zhang, Q. Zeng, and J.-B. Sheu, “Modeling the productivity and stability of a terminal operation system with quay crane double cycling,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 122, pp. 181–197, 2019.

[7] J. He, H. Yu, C. Tan, W. Yan, and C. Jiang, “Quay crane scheduling for multiple hatches vessel considering double-cycling strategy,” *Industrial Management & Data Systems*, vol. 120, no. 2, pp. 253–264, 2020.

[8] E. Ahmed, M. S. El-Abbasy, T. Zayed, G. Alfalah, and S. Alkass, “Synchronized scheduling model for container terminals using simulated double-cycling strategy,” *Computers & Industrial Engineering*, vol. 154, p. 107118, 2021.

[9] D.-H. Lee, J. X. Cao, Q. Shi, and J. H. Chen, “A heuristic algorithm for yard truck scheduling and storage allocation problems,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 5, pp. 810–820, 2009.

[10] S. Sauri and E. Martin, “Space allocating strategies for improving import yard performance at marine terminals,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 6, pp. 1038–1057, 2011.

[11] M. Caserta, S. Voß, and M. Sniedovich, “Applying the corridor method to a blocks relocation problem,” *OR spectrum*, vol. 33, pp. 915–929, 2011.

[12] M. Ji, W. Guo, H. Zhu, and Y. Yang, “Optimization of loading sequence and rehandling strategy for multi-quay crane operations in container terminals,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 80, pp. 1–19, 2015.

[13] M.-W. Tsai, T.-P. Hong, W.-T. Lin *et al.*, “A two-dimensional genetic algorithm and its application to aircraft scheduling problem,” *mathematical Problems in Engineering*, vol. 2015, 2015.

Md. Mahfuzur Rahman received his B.Sc. Eng. in industrial & production engineering degree in 2024 from the Department of Industrial Engineering & Management of Khulna University of Engineering & Technology (KUET), Bangladesh. He developed his career in competitive programming during his academic journey, having the highest contest ratings in ’CodeChef’ is ⋆⋆⋆, 1696 and in ’Codeforces’ is 1369, also placed 6th position in Intra KUET Programming Contest 2022 (team contest). As a competitive programmer, his main research areas are data structures and algorithms, operations research, and optimization.

Md Abrar Jahin , CSCA™ (Certified Supply Chain Analyst), is an AI researcher and multidisciplinary scientist who obtained his B.Sc. Eng. degree in industrial & production engineering from Khulna University of Engineering & Technology (KUET) in Bangladesh. Abrar's exceptional leadership skills and innovative thinking led him to win the "Smart Roads – Winter Road Maintenance Hackathon 2021" and subsequently earn a three-week research internship at UiT - The Arctic University of Norway. During this internship, he actively contributed to the DIT4BEARs Smart Roads project and conducted research in the field of Machine Learning. In 2021, he received a fully funded scholarship as a 'Research Intern' at Japan's Okinawa Institute of Science and Technology Graduate University (OIST). Under the supervision of Prof. Miller in the Physics and Biology Unit, he focused on investigating perfectly conserved sequences in humans and other species. Building on his previous achievements, he was offered another opportunity in 2023 as a 'Visiting Research Student' at OIST. He has been actively collaborating as a 'Visiting Researcher' since 2024 with Prof. Miller, Dr. Biller, Dr. Zifcakova, Mr. Parker, and Dr. Pnini, delving into the evolutionary dynamics of strongly conserved sequences in vertebrates and insects. Mr. Jahin's research interests span a wide range of disciplines, including Machine Learning, Deep Learning, Quantum Computing, Explainable AI (XAI), Natural Language Processing (NLP), Physics-Informed Neural Networks (PINN), Conformal Prediction, Comparative Genomics, Human-Computer Interaction (HCI), and Supply Chain Analytics. He serves as a peer reviewer for IEEE Access, contributing his expertise to ensure the quality and rigor of scholarly publications in his field.

Md. Saiful Islam has been working as an Assistant Professor in the Department of Industrial Engineering and Management at Khulna University of Engineering & Technology, Khulna, Bangladesh, since 2017. He received his B.Sc. Eng. in industrial & production engineering degree in 2013 and M.Sc. Eng. in Industrial Engineering and Management degree from the Department of Industrial Engineering and Management, Khulna University of Engineering & Technology in 2018. His main research areas are warehouse, inventory, and supply chain management. He has published more than 10 research articles in the corresponding fields.

Dr. M. F. Mridha (Senior Member, IEEE) is currently working as an Associate Professor in the Department of Computer Science at American International University-Bangladesh (AIUB). Before that, he worked as an Associate Professor and chairman in the Department of Computer Science and Engineering at the Bangladesh University of Business and Technology (BUBT). He also worked as a CSE department faculty member at the University of Asia Pacific and as a graduate head from 2012 to 2019. He received his Ph.D. in AI/ML from Jahangirnagar University in 2017. His research experience in academia and industry has resulted in over 120 journal and conference publications. His research contributed to the reputed Journal of Scientific Reports Nature, Knowledge-Based Systems, Artificial Intelligence Review, IEEE Access, Sensors, Cancers, Biology and Applied Sciences, etc. His research interests include artificial intelligence (AI), machine learning, deep learning, and natural language processing (NLP). For more than 10 (Ten) years, he has been with the masters and undergraduate students as a supervisor of their thesis work. His research interests include artificial intelligence (AI), machine learning, natural language processing (NLP), big data analysis, etc. He has been a program committee member in several international conferences/workshops. He served as an academic editor for several journals, including PLOS ONE Journal. He has served as a reviewer of reputed journals like IEEE Transactions on Neural Networks, IEEE Access, Knowledge-Based Systems, Expert Systems with Applications, Bioinformatics, Springer Nature, MDPI, etc. and conferences like ICCIT, HONET, ICIEV, IJCCI, ICAEE, ICCAIE, ICSIPA, SCORED, ISIEA, APACE, ICOS, ISCAIE, BEIAC, ISWTA, IC3e, ISWTA, CoAST, icIVPR, ICSCT, 3ICT, DATA21, etc.

Jungpil Shin (Senior Member, IEEE) received a B.Sc. degree in computer science and statistics and an M.Sc. degree in computer science from Pusan National University, South Korea, in 1990 and 1994, respectively, and a Ph.D. degree in computer science and communication engineering from Kyushu University, Japan, in 1999. He was an Associate Professor, a Senior Associate Professor, and a Full Professor at the School of Computer Science and Engineering, The University of Aizu, Japan, in 1999, 2004, and 2019, respectively. He has co-authored more than 300 published papers for widely cited journals and conferences. His research interests include pattern recognition, image processing, computer vision, machine learning, human–computer interaction, non-touch interfaces, human gesture recognition, automatic control, Parkinson's disease diagnosis, ADHD diagnosis, user authentication, machine intelligence, handwriting analysis, recognition, and synthesis. He is a member of ACM, IEICE, IPSJ, KISS, and KIPS. He received a scholarship from the Japanese Government (MEXT). He served as a program chair and program committee member for numerous international conferences. He serves as an Editor for IEEE journals, Sensors (MDPI), Electronics (MDPI), and Tech Science. He serves as a reviewer for several major IEEE and SCI journals.