A Parallel Berth Allocation Problem in Multipurpose Inland Waterway Terminals

A. Pasha[a,b,*], J. Liu[b], R. Rastogi[a]

[a] Department of Civil Engineering, Indian Institute of Technology Roorkee, Roorkee, India
[b] Loughborough Business School, Loughborough University, Loughborough, United Kingdom
apasha@ce.iitr.ac.in, j.y.liu@lboro.ac.uk, rajat.rastogi@ce.iitr.ac.in
* Corresponding author

Extended abstract submitted for presentation at the 12th Triennial Symposium on Transportation Analysis conference (TRISTAN XII)
June 22-27, 2025, Okinawa, Japan

March 27, 2025

Keywords: Berth Allocation Problem, Inland water transport, Multipurpose terminal, Parallel berthing, Memetic Algorithm

1 INTRODUCTION

Inland Water Transport (IWT) is an essential, yet underutilized, part of global logistics networks, especially in developing countries. It offers numerous advantages, such as high capacity, cost-effectiveness, safety, and sustainability. In India, despite government initiatives and a waterway network of over 20,000 kms, IWT accounts for less than 2% of total freight movement. The potential of IWT remains underutilized due to challenges like outdated infrastructure, limited funding, and inefficient terminal operations. For terminal operations, the Berth Allocation Problem (BAP) plays a vital role, as it directly impacts operational efficiency and IWT competitiveness.

BAP has been widely researched based on attributes related to *spatial* aspects, *temporal* aspects, *handling* time, and *performance measures* (Bierwirth and Meisel, 2015). However, majority of this research has primarily focused on seaport terminals handling specialized cargos like Container and Bulk. In contrast, IWT terminals in developing countries are multipurpose terminals which handle a variety of cargo types within a single integrated facility using heterogenous handling equipment. Barges may carry multiple cargo types, berths can be earmarked for specific cargos, or the cargo may be directly transloaded between trucks and barges without any storage.

Moreover, to maximize the relatively limited capacity of multipurpose IWT terminals, these facilities often employ a "Parallel Berthing" approach, where two barges can be berthed side by side, one on the inner side and the other on the outer side. This setup complicates the handling process and requires novel approach with respect to all BAP attributes discussed above.

Consequently, this study aims to address the BAP in multipurpose IWT terminals explicitly modelling parallel berthing by developing a mathematical model and a dynamic programming based memetic algorithm (DPMA) to obtain near-optimal solutions within a reasonable computational time. These methods are then validated using real-world data from an Indian IWT terminal.

2 METHODOLOGY

2.1 Model description

A Mixed Integer Linear Programming (MILP) model is developed, wherein the spatial, temporal, and handling time attributes are incorporated into the constraints, while the performance measure is represented by the objective function. For the *spatial* attribute, we adopt a discrete berth layout. The constraints ensure each barge is assigned to a single berth that meets its compatibility criteria regarding both single and parallel berthing. In terms of *temporal* attribute, dynamic barge arrivals are considered. These constraints prevent overlap, link berthing times to barge arrivals, stipulate sequential berthing, and mandate outer barges to berth later and depart earlier than inner barge.

The *handling time* attribute depends on equipment handling rates, cargo handling capacity, cargo type and quantity, and whether the cargo is stored or transloaded. Berths are equipped with heterogenous handling equipment, from fixed conveyors and pipelines to versatile mobile harbor cranes (MHC) for standard handling, and pneumatic systems for transloading, with each equipment having a specific handling rate. Cargo handling at each berth is limited by available equipment, earmarking certain berths to specific barges. For example, cargo needing a conveyor can only be handled at berths equipped with conveyors. Handling time for each barge is calculated by dividing the cargo quantity by the handling/transloading rate of relevant equipment. For barges with multiple cargos, the handling times are summed over for each cargo type. Parallel berthing constraints allow only transloaded barges to berth in parallel and ensure a maximum of two parallel barges per berth.

In terms of *performance measure*, while terminal operators aim to achieve higher throughput, vessel operators get better satisfaction when waiting times are minimal. In line with these objectives, we focus on minimizing both total completion time and scaled maximum waiting time for barges. This approach also prevents excessive waiting times for some barges, which is often an issue with traditional objectives like minimizing total service time or tardiness alone.

Moreover, we carry out data preprocessing to make the model concise and also add valid inequalities to the formulation to reduce the size of the feasible region and computation time.

2.2 Solution approach

Since the BAP has been proven to be NP-hard (Monaco and Sammarra, 2007), it is difficult to solve the MILP model for large-scale instances with commercial solvers. To address this, we propose a **Dynamic Programming based Memetic Algorithm (DPMA)**, which uses a simplified chromosome structure representing only allocation, while sequencing and parallel berthing are handled by a separate Dynamic Programming (DP) algorithm. Genetic operators generate new solutions, while the local search refines them by exploring their neighborhoods for improvements.

The proposed DPMA is outlined in Algorithm 1. It begins by initializing both the model and algorithm-specific parameters. The chromosome structure comprises of $|B|$ genes, where each gene represents a barge, and its allele corresponds to the assigned berth from its feasible berth list.

The **Initialization** generates an initial population of chromosomes using a combination of random assignments and first-come-first-served (FCFS) heuristics to incorporate high-quality solutions early on. The initial population's **Fitness** values are then calculated based on berth scheduling using a novel **Dynamic Programming** (DP) procedure. The DP considers the sequence of each barge at each berth and incorporates parallel berthing configurations, adjusting based on each barge's arrival time, handling requirements, and feasible berth positions.

The main loop begins with **Selection**, where binary tournament selects chromosomes based on fitness. **Crossover** and **Mutation** operators are applied using a decay-based approach where the number of crossover points and mutated genes decrease over generations, focusing on exploration in early stages and exploitation in later stages. **Elitism** ensures that fittest individuals are retained across generations. A **Local Search** is then performed on a specified portion of the offspring population, using both Variable Neighborhood Descent (VND) and Basic Variable Neighborhood Search (BVNS) to fine-tune solutions. VND iteratively applies increasingly disruptive neighborhood structures, such as swapping barge positions within or across berths, relocating barges, and adjusting berth assignments. BVNS complements this by thoroughly exploring neighborhood structures around the best solutions. **Termination** occurs when stopping criteria is met, such as reaching a maximum generation count (*Maxgen*) or no improvement over several iterations. The best chromosome at termination represents the optimized berth allocation and schedule.

Algorithm 1. DPMA framework

DPMA *(Inputdata, Popsize,* P_c*,* P_m*,* P_{ls}*,* E_r*, Termcriteria)*
in: *Inputdata* - model parameters, *Popsize* - population size, P_c - crossover rate, P_m - mutation rate, E_r - elitism ratio, P_{ls} - local search probability, *Termcriteria* - termination criterion
out: *Bestchrom* - Best berth allocation and schedule

$|Pop| \leftarrow Popsize;\ |Fit| \leftarrow Popsize;\ |Parents| \leftarrow Popsize;\ |Offspring| \leftarrow Popsize$ ⊲ initialization
gen ← 0 ⊲ Start generation counter
Pop_{gen} ← **initialization***(Inputdata, Popsize, FCFS)* ⊲ Initialization (Random + FCFS)
$Schedule_{gen}$ ← ***DP***$(Pop_{gen}, Inputdata)$ ⊲ Initial population DP scheduling
Fit_{gen} ← **Fitness**$(Schedule_{gen}, Inputdata)$ ⊲ Initial population fitness evaluation
while *Termcriteria* != TRUE **do** ⊲ Iterate until stopping criteria is met
  gen ← gen + 1 ⊲ Update generation counter
  $Parents_{gen}$ ← **Selection**(Pop_{gen}, Fit_{gen}) ⊲ Parent selection
  $Offsprings_{gen}$ ← **Crossover**$(Parents_{gen}, P_c)$ ⊲ Crossover
  $Offsprings_{gen}$ ← **Mutation**$(Offsprings_{gen}, P_m)$ ⊲ Mutation
  $Schedule_{gen}$ ← ***DP****(Offspringsgen, Inputdata)* ⊲ Offspring DP scheduling
  Fit_{gen} ← **Fitness**$(Schedule_{gen}, Inputdata)$ ⊲ Offspring fitness evaluation
  $Offsprings_{gen}$ ← **Elitism**$(Pop_{gen}, Offsprings_{gen}, Fit_{gen}, E_r)$ ⊲ Elitism
  $Pop_{(gen+1)}$ ← **Localsearch**$(Offsprings_{gen}, Fit_{gen}, Inputdata, P_{ls})$ ⊲ Local search
end while
$Bestchrom \leftarrow Pop[argmax(Fit_{gen})]$ ⊲ Get the best chromosome from the population
return *Bestchrom* ⊲ Return the best chromosome as solution

3 RESULTS AND DISCUSSION

Numerical experiments are conducted to evaluate the performance of the proposed mathematical model and solution algorithm across various data instances. The MILP model is implemented using the DOcplex Python library within CPLEX 22.1.1, with default parameter settings and a maximum runtime of one hour per instance. All algorithms are executed in MATLAB R2024b, with parameter tuning performed in Minitab 22.1. The experiments are conducted on an Apple M2 Pro 12 Core CPU with 32GB of unified memory. The algorithm-specific parameters are tuned using Taguchi method.

The input data is generated using the terminal and operational data from Haldia Multimodal Terminal (MMT), Inland Waterways Authority of India (IWAI), and other terminal operators in India. Using this data, the problem instances are classified based on the number of barges (|B|) ranging from 8 to 40, and the number of cargo types (|C|) ranging from 4 to 10. This results in 32 unique instances, divided into small-scale (8-20 barges) and large-scale (28-40 barges) categories.

The results show that CPLEX achieves optimal solutions only for small instances within the time limit but fails for larger instances. DPMA consistently reaches optimality (out of 10 runs) in small-scale instances, often matching CPLEX solutions with minimal average gaps (0.12%). For larger instances, DPMA maintains strong performance with an average gap of 0.17% against CPLEX, achieving these solutions in significantly less runtime (average of 35 seconds). DPMA significantly outperforms FCFS heuristics with gaps ranging from 20-30%, revealing the inefficiencies of current terminal practices. DPMA is also compared with DPGA which excludes the local search component. While DPGA performs better than FCFS heuristics, it converges prematurely, especially in larger instances, highlighting the benefits of local search in DPMA. On average, the gaps between DPMA and DPGA are 2.52% for small instances and 5.60% for large instances. The nonparametric Kruskal-Wallis and Mann-Whitney U tests, confirm these performance differences between the algorithms. Overall, the high solution quality, fast convergence, and lower standard deviation of DPMA make it the preferred approach for the BAP, particularly for large-scale applications.

As shown in Figure 1, a comparison between parallel and single berthing reveals that parallel berthing achieves significant objective improvements, averaging over 13% in small instances and 18% in large instances, highlighting its effectiveness in enhancing operational efficiency.

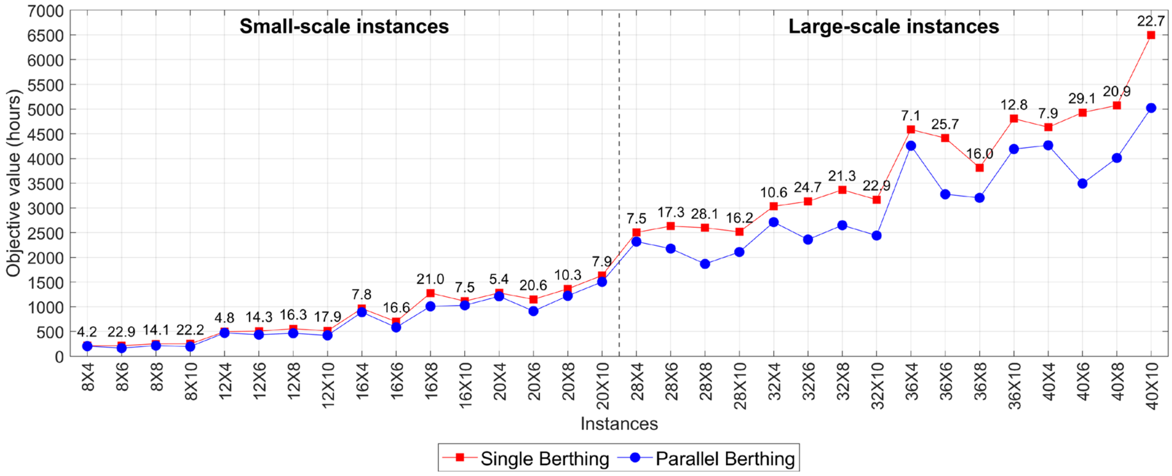

Figure 1 – *Comparison between single and parallel berthing*

Finally, we conduct a sensitivity analysis to examine how congestion levels, berth feasibility, and parallel handling compatibility affect performance metrics. Results show that higher congestion levels increase waiting times but reduce completion times, while multipurpose berths significantly improve flexibility and reduce delays. Increasing compatibility for parallel cargo handling also greatly reduces waiting and completion times, especially when all cargo types can use outer berths. Overall, these findings provide valuable managerial insights for optimizing terminal operations and improving decision making processes.

4 CONCLUSIONS

This paper addresses the BAP for multipurpose IWT terminals, particularly relevant for developing countries like India, where complex operational features include heterogeneous cargo types, direct transloading, heterogenous equipment, multiple cargo on barges, and parallel berthing. A MILP model was formulated to minimize total completion and maximum waiting times. Data preprocessing and valid inequalities helped streamline the model. A DP based memetic algorithm (DPMA) was developed to solve large problem instances. Computational experiments show that DPMA outperforms CPLEX and DPGA in solution quality, consistency, and robustness, especially for large instances, and demonstrates clear advantages over the FCFS heuristics which is the current industrial practice at IWT terminals in India. The integration of parallel berthing significantly improves terminal efficiency by reducing congestion and waiting times, especially in terminals where barge traffic and cargo diversity are high. Sensitivity analysis further reveals how congestion, berth feasibility, and parallel handling compatibility impact performance, providing useful insights.

The proposed approach is expected to result in improved terminal performance and increased competitiveness of the IWT system. The outcomes of this study will be valuable to policymakers, terminal operators, and other stakeholders in the transportation sector, enabling them to make informed decisions in the management of IWT terminals in India and other developing countries.

Future research could explore hybrid berthing layouts, modelling uncertainty in arrival and handling, and integration of BAP with yard allocation.

References

Bierwirth, C. and Meisel, F., 2015. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, **244** (3), pp. 675-689.

Monaco, M.F. and Sammarra, M., 2007. The berth allocation problem: a strong formulation solved by a Lagrangean approach. *Transportation Science*, **41** (2), pp. 265-280.