

ĐIỀU ĐỘ ĐỘNG TÀU VÀO BẾN CẢNG CONTAINER BẰNG GIẢI THUẬT LOCAL SEARCH

DYNAMIC SCHEDULING OF SHIPS AT CONTAINER PORT BERTHS USING
LOCAL SEARCH ALGORITHM

Hoàng Trọng Trần Huy

Khoa Công nghệ Cơ khí, Trường Đại học Công Thương Thành Phố Hồ Chí Minh

Email: tantranhaico@gmail.com

Hoặc: huyhtt@huit.edu.vn

TÓM TẮT

Bài báo này tập trung vào việc nghiên cứu bài toán điều độ tàu vào bến cảng container trong trường hợp động, với mô hình không gian cầu cảng hỗn hợp, kết hợp giữa các đặc điểm của cầu cảng rời rạc và liên tục. Bài toán được thiết lập dựa trên nền tảng của bài toán cutting stock 2D, một bài toán NP-Hard, và phù hợp với hoạt động tại các cảng container ở Việt Nam, nơi không gian cầu cảng thường hạn chế. Nghiên cứu đề xuất một giải thuật meta-heuristic để giải quyết bài toán với kích thước tương đương thực tế trong thời gian tính toán chấp nhận được. Tính hiệu quả của phương pháp được kiểm chứng thông qua các thí nghiệm số. Kết quả nghiên cứu nhằm tối ưu hóa việc sử dụng không gian cầu cảng, giảm thiểu thời gian quay vòng tàu và hỗ trợ các nhà khai thác cảng trong việc lập lịch trình tàu một cách hiệu quả và linh hoạt.

Từ khóa: Cầu cảng; Tối ưu hóa; Meta-heuristics; Điều độ; Tàu container.

ABSTRACT

This paper focuses on studying the problem of ship scheduling at container port berths in dynamic conditions, utilizing a mixed berth space model that combines the characteristics of discrete and continuous berths. The problem is formulated based on the 2D cutting stock problem, an NP-Hard problem, and aligns with operations at container ports in Vietnam, where berth space is often limited. The research proposes a meta-heuristic algorithm to solve the problem at realistic scales within acceptable computation times. The efficiency of the method is verified through numerical experiments. The research aims to optimize berth space usage, minimize ship turnaround time, and assist port operators in scheduling ships effectively and flexibly.

Keywords: Berth; Optimization; Meta-heuristics; Scheduling; Container vessels.

1. GIỚI THIỆU

Bài toán phân bổ bến (Berth Allocation Problem - BAP), còn được biết đến là bài toán lập lịch bến, là một thách thức hoạt động quan trọng trong các cảng container. Vấn đề này liên quan đến việc chỉ định không gian cầu cảng cho các tàu theo thời gian để tạo điều kiện thuận lợi cho hoạt động bốc xếp hàng hóa. Việc phân bổ bến hiệu quả có vai trò tối quan trọng trong việc tối đa hóa năng suất của cảng, giảm thiểu thời gian quay vòng của tàu và giảm tắc nghẽn. Chi phí xây dựng và mở rộng các bến là rất lớn, do đó việc sử dụng tối ưu các nguồn lực hiện có trở nên đặc biệt quan trọng. Bản chất cơ bản của BAP là một bài toán NP-completed, điều này cho thấy việc tìm ra giải pháp tối ưu cho các trường hợp quy mô lớn trong thực tế là rất khó khăn về mặt tính toán. Điều này giải thích cho khối lượng lớn các nghiên cứu dành cho việc phát triển các phương pháp heuristic và metaheuristic hiệu quả.

Sự gia tăng không ngừng của khối lượng giao thông container và không gian bến có hạn càng làm nổi bật tầm quan trọng của các chiến lược phân bổ bến hiệu quả đối với sự bền vững và tăng trưởng của các cảng. Các giải pháp BAP hiệu quả có mối tương quan trực tiếp với khả năng xử lý khối lượng thương mại ngày càng tăng của một cảng mà không cần mở rộng cơ sở hạ tầng đáng kể, nhấn mạnh ý nghĩa thực tiễn và kinh tế của lĩnh vực nghiên cứu này.

Lĩnh vực nghiên cứu về điều độ tàu vào bến trong một cảng đã được nhiều tác giả trên thế giới nghiên cứu như [1], [2]. Nếu dựa trên thuộc tính không gian, bài toán có thể được phân thành các cách bố trí bến rời rạc, bố trí bến liên tục, hoặc bố trí bến hỗn hợp. Bố trí bến rời rạc [3] là cầu cảng được chia thành một số hữu hạn các bến, mỗi bến có thể tiếp nhận một tàu tại một

thời điểm. Đối với cách bố trí bến liên tục [4], tàu có thể cập bến ở bất kỳ vị trí nào dọc theo cầu cảng trong phạm vi ranh giới của nó. Phương pháp bố trí bến hỗn hợp [5] là sự kết hợp các khía cạnh của bố trí rời rạc và liên tục, cho phép sắp xếp bến linh hoạt hơn (ví dụ: các tàu nhỏ dùng chung một bến, các tàu lớn chiếm nhiều bến). Việc lựa chọn bố trí bến có ảnh hưởng đáng kể đến độ phức tạp của BAP và các loại mô hình toán học và thuật toán tối ưu hóa phù hợp nhất. Bố trí liên tục, mặc dù có khả năng sử dụng không gian tốt hơn, thường phức tạp hơn về mặt tính toán so với bố trí rời rạc. Điều này cho thấy sự đánh đổi giữa tính thực tế và lợi ích tiềm năng về hiệu quả của các mô hình liên tục so với sự đơn giản tương đối và khả năng giải quyết bằng máy tính của các mô hình rời rạc, ảnh hưởng đến trọng tâm của các dòng nghiên cứu khác nhau. Ngoài ra, nếu dựa trên thời gian tàu đến, bài toán có thể chia thành: tàu đến tĩnh [6], [7] là tất cả các tàu cần được phục vụ đã có mặt tại cảng khi bắt đầu lập lịch. Một khía cạnh khác thời gian tàu đến có thể xem là tàu đến động [8] có nghĩa là tàu đến theo thời gian và thời gian đến của chúng có thể được biết trước hoặc không chắc chắn. Phần lớn các nghiên cứu đã công bố xem xét trường hợp động. Một cách tiếp cận khác thời gian tàu đến không chắc chắn là thời gian đến không được biết chính xác và có thể tuân theo một số phân phối xác suất nhất định. Các kịch bản tàu đến động thể hiện hoạt động thực tế của cảng một cách chân thực hơn và do đó thu hút sự chú ý đáng kể của giới nghiên cứu. Sự phức tạp gia tăng của việc tàu đến động đòi hỏi các phương pháp lập lịch phức tạp hơn so với các kịch bản tĩnh. Điều này nhấn mạnh tính phù hợp thực tế của các nghiên cứu tập trung vào BAP động, vì nó nhằm giải quyết các vấn đề mà các nhà khai thác cảng phải đối mặt hàng ngày. Nhu cầu về các giải pháp mạnh mẽ có thể xử lý tính chất thay đổi theo thời gian của việc tàu đến là rất quan trọng.



Trong nghiên cứu này, tác giả sẽ nghiên cứu bài toán điều độ tàu vào bãi khi xét bài toán động, và đối với cách chia cầu cảng là hỗn hợp. Bài toán được đề cập đến trong nghiên cứu này phù hợp với hoạt động của tàu ở bến cảng của Việt Nam, do bến cảng thông thường có chiều dài cầu cảng hạn chế, và thời gian đến của tàu có thể linh hoạt trong lúc lập lịch trình.

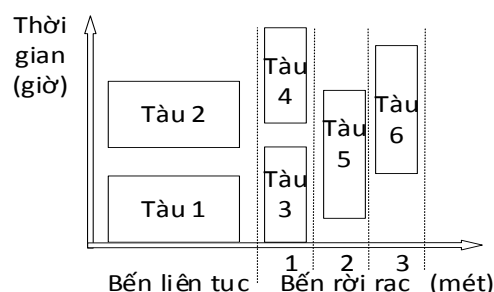
Trong phần kế tiếp, một mô hình toán sẽ được thiết lập. Phần 3 của bài báo giải thuật meta-heuristics được đề xuất để giải bài toán với kích thước tương đương trong thực tế với thời gian tính toán chấp nhận được. Phần 4 chính là phần thí nghiệm số để kiểm chứng mức độ hiệu quả của giải thuật được đề xuất. Cuối cùng là kết luận của bài báo.

2. BÀI TOÁN ĐIỀU ĐỘ TÀU VÀO CẦU CẢNG VÀ MÔ HÌNH TOÁN

2.1. Mô tả bài toán

Trong phần này, bài toán điều độ tàu vào cầu cảng được miêu tả chi tiết. Trong bài nghiên cứu này, bài toán điều độ sẽ được mô hình dựa trên nền tảng của bài toán cutting stock 2D, đây là một bài toán NP-Hard, tức là không tồn tại một phương pháp giải xác định nào có thể giải bài toán trong thời gian tính toán tuân theo hàm đa thức khi kích thước bài toán thay đổi. Hai chiều của bài toán cutting stock được tương ứng với chiều dài và thời gian phục vụ tàu trong cảng. Thêm vào đó, bài toán này dựa trên bài toán điều độ động, tức là thời gian sẵn sàng của các tàu khác không khi lịch trình được lên. Và mô hình không gian cảng sẽ kết hợp giữa mô hình cảng rời rạc và cảng liên tục. Mô hình hybrid này cho phép tận dụng ưu điểm của cả hai cách tiếp cận: sự linh hoạt của mô hình liên tục để tối ưu hóa việc sử dụng không gian và khả năng mô hình hóa các đặc điểm

cụ thể của từng vị trí bến (như loại cần cẩu, độ sâu) thông qua các đoạn rời rạc. Hình 1 thể hiện bài toán điều độ tàu vào cảng với bến cảng dạng hỗn hợp bao gồm bến liên tục, các tàu có thể neo đậu trong khoảng không gian bến, chỉ cần đảm bảo khoảng cách an toàn, và các bến rời rạc, tàu được neo đậu trong từng bến cảng được chia sẵn, chiều dài tàu phải bé hơn chiều dài của bến.



Hình 1. Bến tàu hỗn hợp

2.2. Mô hình toán bài toán điều độ tàu vào cầu cảng

Bài toán này với mục tiêu tối ưu hóa việc sắp xếp các tàu container vào cầu cảng để giảm thiểu tổng thời gian phục vụ của các tàu. Sau đây là chú thích các ký hiệu của mô hình toán:

Tập hợp:	
$V=1,2,3,\dots,N$	Tập hợp các tàu container được điều độ
$K=1,2,3,\dots,M$	Tập hợp các vị trí cầu cảng rời rạc
Tham số của tàu:	
a_i	Thời gian tàu i tới cảng (ready time), với $i \in V$
p_i	Thời gian phục vụ tàu i , với $i \in V$
l_i	Chiều dài của tàu i , với $i \in V$
s_i	Thời điểm mong muốn bắt đầu phục vụ tàu i , với $i \in V$

e_i	Thời điểm mong muốn kết thúc phục vụ tàu i , với $i \in V$
Tham số cầu cảng:	
$L_{\max}$	Tổng chiều dài của cầu cảng liên tục
L_k	Chiều dài của vị trí cầu cảng rời rạc k , với $k \in K$
Q_k^s	Vị trí bắt đầu của cảng rời rạc k , với $k \in K$
Q_k^e	Vị trí kết thúc của cảng rời rạc k , với $k \in K$
H	Tổng thời gian lập kế hoạch
Biến quyết định:	
x_i	Vị trí cập cảng (theo chiều dọc cầu cảng) của mũi tàu i , với $i \in V$
t_i	Thời gian bắt đầu phục vụ tàu i , với $i \in V$
z_{ik}	Biến nhị phân, có giá trị 1 khi tàu i được phân vào vị trí cầu cảng rời rạc k , giá trị 0 nếu ngược lại
w_i	Biến nhị phân, giá trị 1 nếu tàu i được phân vào cầu cảng liên tục, giá trị 0 nếu ngược lại
$\alpha_{ij}, \beta_{ij}, \gamma_{ij}, \delta_{ij}$	Biến nhị phân phụ trợ, có giá trị 1 khi tàu i và tàu j có khả năng chồng lấn về thời gian cũng như không gian, giá trị 0 nếu ngược lại

Hàm mục tiêu:

$$\text{Min } C_{\max} \quad (2.1)$$

Ràng buộc:

$$C_{\max} \geq t_i + p_i, \forall i \in V \quad (2.2)$$

$$t_i \geq a_i, \forall i \in V \quad (2.3)$$

$$t_i + p_i \leq H, \forall i \in V \quad (2.4)$$

$$t_i + p_i \leq t_j + M\alpha_{ij}, \forall i, j \in V, i \neq j \quad (2.5)$$

$$t_j + p_j \leq t_i + M\beta_{ij}, \forall i, j \in V, i \neq j \quad (2.6)$$

$$x_i + l_i \leq x_j + M\gamma_{ij}, \forall i, j \in V, i \neq j \quad (2.7)$$

$$x_j + l_j \leq x_i + M\delta_{ij}, \forall i, j \in V, i \neq j \quad (2.8)$$

$$\alpha_{ij} + \beta_{ij} + \gamma_{ij} + \delta_{ij} \leq 3, \forall i, j \in V, i \neq j \quad (2.9)$$

$$w_i + \sum_{k \in K} z_{ik} = 1, \forall i \in V \quad (2.10)$$

$$x_i + l_i \leq Q_k^s + M(1 - w_i), \forall i \in V, k \in K \quad (2.11)$$

$$Q_k^s - x_i \leq M(1 - z_{ik}), \forall i \in V, k \in K \quad (2.12)$$

$$x_i + l_i - Q_k^e \leq M(1 - z_{ik}), \forall i \in V, k \in K \quad (2.13)$$

$$l_i \leq L_k z_{ik} + M(1 - z_{ik}), \forall i \in V, k \in K \quad (2.14)$$

Hàm mục tiêu trong mô hình là tối thiểu thời gian hoàn thành của tất cả các tàu. Ràng buộc 2.2 thể hiện định nghĩa của thời gian hoàn thành của tàu cuối cùng. Ràng buộc 2.3 thể hiện bài toán động, tàu không thể được phục vụ trước khi nó đến bến cảng. Các ràng buộc từ 2.5 đến 2.9 nhằm tránh sự chồng lấn của các tàu về không gian và thời gian. Các ràng buộc từ 2.10 đến 2.14 là các ràng buộc về việc phân bổ cầu cảng cho tàu dựa trên việc phân bổ vào bến liên tục hay bến rời rạc, và các giới hạn liên quan đến chiều dài của tàu so với cầu cảng. Vì bài toán được mô hình dựa trên bài toán cutting stock 2D và đây là một bài toán NP-hard nên không có phương pháp xác định nào có thể giải được trong thời gian tính toán theo hàm đa thức. Phần kế tiếp sẽ đề xuất giải thuật local 

search để giải bài toán trong thời gian tính toán hợp lý.

3. GIẢI THUẬT LOCAL SEARCH

Giải thuật được đề xuất là một phương pháp tìm kiếm cục bộ (Local Search), một dạng metaheuristic, nhằm giải quyết Bài toán Điều độ Bến bãi Động (Dynamic Berth Allocation Problem - DBAP). Mục tiêu chính là tìm kiếm một lịch trình tối ưu cho việc phân bổ các tàu vào các bến cảng có sẵn sao cho tối thiểu hóa một hàm mục tiêu, cụ thể trong trường hợp này là makespan (thời điểm hoàn thành của con tàu cuối cùng).

1) Cấu trúc biểu diễn lời giải

Một lịch trình (lời giải) được biểu diễn dưới dạng một mảng số nguyên ($\text{int}[]$). Cấu trúc này có hai thành phần chính:

- ID Tàu: Các số nguyên dương đại diện cho mã định danh (ID) của mỗi con tàu.
- Dấu phân cách (Separator): Số 0 được sử dụng như một dấu hiệu để phân tách hàng đợi của các bến.

Ví dụ: Lời giải $\{8, 11, 0, 14, 48, 0, \dots\}$ có nghĩa là:

- Bến 1 sẽ phục vụ Tàu 8, sau đó đến Tàu 11.
- Bến 2 sẽ phục vụ Tàu 14, sau đó đến Tàu 48.
- Và cứ tiếp tục như vậy cho các bến còn lại.

2) Hàm mục tiêu

Hàm mục tiêu được sử dụng để đánh giá chất lượng của một lời giải là makespan. Giá trị này được tính bằng cách mô phỏng toàn bộ lịch trình. Giải thuật duyệt qua mảng lời giải, với mỗi tàu, nó tính toán thời gian bắt đầu phục vụ sớm nhất bằng cách lấy giá trị lớn

hơn giữa thời gian tàu đến và thời gian bến sẵn sàng. Thời gian hoàn thành của tàu được tính bằng thời gian bắt đầu + thời gian xử lý. Thời gian này sau đó sẽ cập nhật lại thời gian bến sẵn sàng cho tàu tiếp theo trong hàng đợi.

3) Quy trình của Giải thuật Local Search

Giải thuật hoạt động theo một chiến lược tìm kiếm lặp đi lặp lại để cải thiện dần lời giải. Đây là một biến thể của Iterated Local Search (Tìm kiếm Cục bộ Lặp lại).

Bước 1: Khởi tạo

- Giải thuật bắt đầu bằng cách tạo ra một lời giải ban đầu hoàn toàn ngẫu nhiên nhưng hợp lệ (chứa đủ ID tất cả các tàu và số lượng đầu 0 cần thiết).

Bước 2: Tìm kiếm lân cận

- Từ lời giải hiện tại, giải thuật tạo ra các “lời giải lân cận” bằng cách thực hiện những thay đổi nhỏ. Có hai phép toán tạo lân cận chính:

- Hoán vị: Chọn ngẫu nhiên hai vị trí trong mảng và hoán đổi giá trị của chúng.

- Di chuyển: Chọn ngẫu nhiên một tàu và di chuyển nó đến một vị trí mới trong mảng.

Bước 3: Chiến lược chấp nhận và thoát khỏi tối ưu cục bộ

- Trong mỗi vòng lặp, giải thuật sẽ thử một số lượng lân cận nhất định.

• Nếu tìm thấy lân cận tốt hơn (có makespan nhỏ hơn) lời giải hiện tại, nó sẽ di chuyển đến lân cận đó.

- Nếu không tìm thấy lân cận nào tốt

hơn, tức là đang bị kẹt tại một điểm tối ưu cục bộ, giải thuật sẽ thực hiện một bước đi ngẫu nhiên (bằng cách hoán vị hoặc di chuyển) để nhảy đến một vùng không gian tìm kiếm khác. Đây là cơ chế then chốt giúp thuật toán thoát khỏi các điểm tối ưu cục bộ và có cơ hội tìm ra lời giải toàn cục tốt hơn.

- Trong suốt quá trình, giải thuật luôn lưu lại lời giải tốt nhất đã từng tìm thấy.

Bước 4: Kết thúc

- Quá trình tìm kiếm sẽ dừng lại sau khi đạt đến số vòng lặp tối đa. Lời giải tốt nhất tìm được trong toàn bộ quá trình sẽ là kết quả cuối cùng.

4) Thực nghiệm số

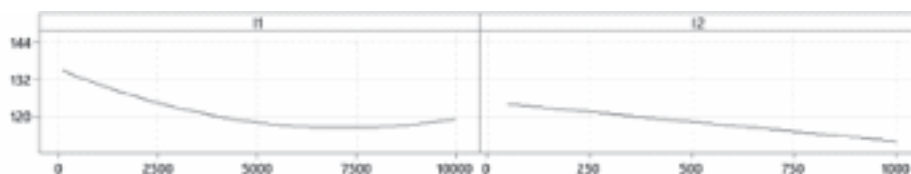
Để tìm ra bộ tham số hiệu quả nhất cho giải thuật Local Search, một loạt các thí nghiệm đã được tiến hành theo phương pháp Thiết kế Thí nghiệm (Design of Experiments - DOE). Hai tham số chính được khảo sát là: I1: Số vòng lặp tối đa (maxIterations) và I2: Số lượng lân cận được tạo và đánh giá trong mỗi vòng lặp (neighborsToTry). Với mục tiêu là tìm ra cặp (I1, I2) cho giá trị Response (đại diện cho makespan) nhỏ nhất. Dữ liệu thu thập được từ 40 lần chạy thí nghiệm đã được tổng hợp và phân tích. Bảng dưới đây tóm tắt các chỉ số thống kê quan trọng cho từng bộ tham số.

Bảng 1. Thực nghiệm xác định tham số, trên phần mềm Minitab

STT	I1	I2	Makespan	STT	I1	I2	Makespan
1	10000	1000	117	21	100	50	142
2	5050	525	116	22	5050	525	118
3	100	50	137	23	10000	1000	118
4	5050	525	117	24	100	50	174
5	10000	50	121	25	5050	525	118
6	10000	1000	115	26	10000	50	122
7	100	1000	130	27	100	1000	126
8	5050	525	118	28	100	50	147
9	100	50	131	29	5050	525	117
10	10000	50	122	30	10000	50	122
11	100	1000	122	31	100	1000	125
12	10000	1000	117	32	10000	1000	116
13	100	1000	124	33	5050	525	118
14	5050	525	117	34	5050	525	118
15	100	50	140	35	100	525	127
16	10000	50	122	36	5050	525	118
17	10000	1000	117	37	5050	525	117
18	100	1000	128	38	10000	525	118
19	5050	525	118	39	5050	50	125
20	10000	50	123	40	5050	1000	117



Qua quá trình phân tích quy hoạch thực nghiệm, điểm tối ưu đối với I1 là 6200 và I2 là 1000.



Hình 2. Đồ thị các nhân tố chính khi quy hoạch

Trong bảng 2 là 15 trường hợp kích thước khác nhau của bài toán với kích thước tương đương trong thực tế, giải thuật được thực hiện trên máy tính Windows 11, với ngôn ngữ lập trình Java, CPU Core™ i7 – 10750H. Ứng với mỗi bài toán, việc tính toán bằng giải thuật được lập lại 20 lần để thống kê. Kết quả cho thấy giải thuật đạt độ ổn định do độ lệch chuẩn nhỏ và thời gian tính toán chấp nhận được.

Bảng 2. Chạy thực nghiệm số

STT	Số tàu	Số bến	Giá trị makespan ban đầu trung bình	Giá trị makespan sau khi giải trung bình	Độ lệch chuẩn của makespan sau khi giải	Trung bình thời gian tính toán (giây)
1	10	4	91	31	0.0	3.12
2	15	4	109	53	0.0	4.74
3	20	8	101	43	0.0	5.44
4	25	8	155	46	0.5	7.46
5	30	8	163	56	0.3	7.58
6	35	8	170	62	0.4	8.90
7	40	10	207	58	0.5	9.22
8	45	10	189	64	0.6	10.39
9	50	10	244	70	0.6	12.07
10	55	10	243	77	0.6	12.65
11	60	10	245	82	0.7	12.96
12	65	10	253	89	0.6	13.58
13	70	12	277	81	0.6	14.65
14	75	12	226	87	0.7	15.92
15	80	12	299	94	0.7	16.72

5. KẾT LUẬN

Bài báo đã nghiên cứu về việc điều độ tàu vào bến, với giả thuyết bến hỗn hợp, bao gồm phần bến liên tục và bến rời rạc. Mô hình toán đã được lập và một giải thuật meta-heuristics được đề xuất để giải quyết bài toán với thời gian tính toán hợp lý. Phương pháp quy hoạch thực nghiệm đã được sử dụng để tìm được tham số tối ưu cho giải thuật là số vòng lặp để tìm kiếm nghiệm là 6200 và số lần cần tìm kiếm lời giải lân cận trước khi cập nhật lời giải hiện hành nên là 1000 lần. Các vấn đề với kích thước khác nhau đã được đưa ra để kiểm chứng độ ổn định và tốc độ tính toán của giải thuật. ❖

Ngày nhận bài: **16/5/2025**

Ngày phản biện: **09/6/2025**

Tài liệu tham khảo:

- [1]. M. M. Golias, M. Boile, and S. Theofanis, “A lamda-optimal based heuristic for the berth scheduling problem”. *Transportation Research Part C*, vol. 18, pp. 794-806, 2010.
- [2]. K. H. Kim and K. T. Park, “A note on a dynamic space-allocation method for outbound containers”. *European Journal of Operational Research*, vol. 148, pp. 92-101, 2003.
- [3]. K. Buhrkal, S. Zuglian, S. Ropke, J. Larsen, and R. M. Lusby, “Models for the Discrete Berth Allocation Problem: A Computational Comparison”. *DTU Management*, Rep. 14, 2009.
- [4]. P. Frojan, J. F. Correcher, R. Alvarez-Valdes, G. Koulouris, and J. M. Tamarit, “The continuous Berth Allocation Problem in a container terminal with multiple quays”. *Expert Systems with Applications*, 2015.
- [5]. E. T. Bacalhau, L. Casacio, and A. T. de Azevedo, “New hybrid genetic algorithms to solve dynamic berth allocation problem”. *Expert Systems with Applications*, vol. 167, p. 114198, 2021.
- [6]. D. Chang, Z. Jiang, W. Yan, and J. He, “Integrating berth allocation and quay crane assignments. *Transportation Research Part E*, vol. 46, pp. 975-990, 2010.
- [7]. A. Imai, E. Nishimura, and S. Papadimitriou, “Marine container terminal configurations for efficient handling of mega-containerships”. *Transportation Research Part E: Logistics and Transportation Review*, vol. 49, pp. 141-158, Jan. 2013.
- [8]. B. Lu, S. Hao, M. Zhao, and H. Wang, “Dynamic berth allocation problem for a novel hybrid berth system in container terminal”. *SSRN*, 2022. [Online]. Available: <https://ssrn.com/abstract=4205029>