



## GIẢI THUẬT NÂNG CAO

---

Bài tập lớn

# Xếp lịch cho bến tàu tại Tân Cảng

---

Tutors

Hồ Xuân Long

Trang Hồng Sơn

Huynh Tuong Nguyen

Student name

Giải thuật nâng cao - Nhóm 1

Phan Phước Minh – 2010418

Nguyễn Đắc Hoàng Phú – 2010514

Lê Trọng Phú – 2070424

Trần Bình Phương Nhân – 2170550

Phiên bản 1.0

## Mục lục

<b>1</b>	<b>Giới thiệu</b>	<b>2</b>
<b>2</b>	<b>Bài toán phân bổ tài nguyên và xếp lịch tại bến tàu</b>	<b>2</b>
<b>3</b>	<b>Mô tả bài toán</b>	<b>4</b>
3.1	Biểu đồ không-thời gian . . . . .	5
3.2	Mô hình hóa toán học cho bài toán . . . . .	5
3.2.1	Các thông số . . . . .	6
3.2.2	Biến ra quyết định . . . . .	6
3.2.3	Biến trung gian . . . . .	6
3.2.4	Mô hình toán học . . . . .	6
<b>4</b>	<b>Giải pháp đề xuất</b>	<b>7</b>
4.1	Construction Phase . . . . .	9
4.2	Optimize Phase . . . . .	9
4.2.1	Swap in break method . . . . .	9
4.2.2	Swap all method . . . . .	10
<b>5</b>	<b>Định dạng nhập-xuất và kết quả thực nghiệm</b>	<b>10</b>
5.1	Dữ liệu đầu vào . . . . .	10
<b>6</b>	<b>Kết quả thực nghiệm</b>	<b>12</b>
6.1	Yêu cầu công việc . . . . .	12
6.2	Đánh giá kết quả . . . . .	12
6.2.1	Construction phase . . . . .	13
6.2.2	Optimize phase . . . . .	13
<b>7</b>	<b>Kết luận</b>	<b>14</b>

# Static berth allocation problem in an area of continuous-segments

Ho Xuan Long, Trang Hong Son, Huynh Tuong Nguyen  
Faculty of Computer Science & Engineering,  
Ho Chi Minh city University of Technology, Vietnam  
268 Lý Thường Kiệt, Hồ Chí Minh, Viet Nam  
htnguyen@hcmut.edu.vn

Ngày 21 tháng 11 năm 2022

## Tóm tắt nội dung

In this study, blah blah.

**Keywords:** berth allocation problem; continous berth; static berth allocation problem.

## 1 Giới thiệu

Xuất phát từ những nhu cầu phát sinh trong việc lập kế hoạch sản xuất và xây dựng các hệ thống hỗ trợ quản lý thông qua máy tính, lý thuyết lập lịch là một trong những ngành đã được đào sâu và thu hút nhiều nhà nghiên cứu trên thế giới. Ngày nay, lập lịch là một lĩnh vực quan trọng của ngành tối ưu hóa tổ hợp, và là lĩnh vực liên ngành kết hợp giữa: toán học ứng dụng, khoa học máy tính và khoa học quản lý. Theo một nghĩa rộng, lập lịch là tìm một sắp xếp các nhiệm vụ (hoặc công việc) nhằm thỏa mãn một số ràng buộc về tài nguyên hạn chế và nhằm để đáp ứng một số mục tiêu đề ra (lý thuyết lập lịch được định nghĩa và trình bày chi tiết trong nhiều sách xuất bản trên thế giới [10, 12, 15, 18, 19]).

Lập lịch theo nghĩa rộng là một quá trình hỗ trợ quyết định cách cấp phát tài nguyên để xử lý một tập các tác vụ/công việc sao cho thỏa mãn một số ràng buộc cho sẵn. Nó đóng một vai trò quan trọng trong việc giúp đỡ con người lập kế hoạch sử dụng nguồn tài nguyên hợp lý và có hiệu quả nhất. Và do vậy, nó thường được áp dụng trong các hệ thống quản lý sản xuất công nghiệp và dịch vụ.

Khái niệm về lập lịch không phải là mới: các kim tự tháp Ai Cập đã hơn 3000 năm tuổi, Tôn Tử đã viết về việc xây dựng chiến lược quân sự 2500 năm trước đây, đường sắt xuyên lục địa đã được xây dựng trong khoảng 200 năm, ... Không có bất kỳ công trình nào đã trình bày có thể được thực hiện mà không có ảnh hưởng bởi một hình thức nào đó về lập kế hoạch (hay còn gọi là lập lịch trình - sự hiểu biết các hoạt động và trình tự). Trong khi các nhà quản lý (hoặc các nhà lãnh đạo quân sự, các tổ chức chịu trách nhiệm hoàn thành công trình) phải nhận thức và đánh giá cao của lập kế hoạch (hoặc ít nhất là những người thành công cần phải có)- tuy nhiên có rất ít bằng chứng rõ ràng về một kế hoạch được xây dựng bài bản, chuẩn mực cho đến thế kỷ 20.

Từ những năm 1950, các bài toán về hỗ trợ ra quyết định đã được đào sâu và liên tục cải tiến song hành cùng với ngành khoa học quản lý tại các nước phát triển trên thế giới bởi vì nhu cầu thực tiễn của nó ngày càng tăng và do đó, các bài toán nghiên cứu học thuật ngày càng được mở rộng và sát với thực tế. Trong số những bài toán tối ưu hóa tổ hợp ứng dụng thực tiễn vào thời điểm này, đa phần thường liên quan đến bài toán lập lịch trong thế giới công nghiệp và dùng để quản lý/ tối ưu việc sử dụng các nguồn tài nguyên rất hạn chế. Và từ nền tảng cơ sở đó, các bài toán quản lý trong các lĩnh vực nghiên cứu tối ưu khác như kiến trúc máy tính [17], viễn thông [16], giao thông vận tải [7], hàng không [7, 8], sinh tin học [10], tài chính [11], sức khỏe cộng đồng [13, 14], ... cũng đã được hưởng lợi từ những thành quả mang lại từ các bài toán lập lịch trong sản xuất.

## 2 Bài toán phân bổ tài nguyên và xếp lịch tại bến tàu

Vận tải hàng hải luôn đóng một vai trò quan trọng trong việc trao đổi hàng hóa giữa các châu lục, và việc làm giảm chi phí vận chuyển luôn là một mục tiêu thương mại quan trọng. Để giảm hơn nữa chi phí vận chuyển, các

chủ hàng tìm cách tăng quy mô kinh tế, đóng các tàu container lớn hơn bao giờ hết cho các tuyến đường dài và đòi hỏi các nhà ga có trang thiết bị và công nghệ có thể đáp ứng được (mega-terminals). Hệ thống này được gọi là trung tâm và nói: tàu container biển sâu (tàu mẹ) hoạt động giữa một số lượng hạn chế các bến trung chuyển (trung tâm), và các tàu nhỏ hơn (trung chuyển) liên kết trung tâm với các cổng khác. Sự cần thiết phải quản lý hiệu quả các hoạt động logistic tại các bến container hiện đại, và đặc biệt là tại các trung tâm lớn.

Các bến cảng đóng một vai trò quan trọng trong bối cảnh thương mại thế giới vì chúng là các điểm nối chính chịu trách nhiệm kết nối vận tải đường biển và đường bộ. Có một số vấn đề về tối ưu hóa phát sinh trong các bến cảng và những vấn đề liên quan đến phân bổ bến là một trong những vấn đề quan trọng nhất. Chúng tạo thành cấp độ đầu tiên của các hoạt động lập kế hoạch đầu cuối. Do đó, các quyết định tương ứng ảnh hưởng đến tất cả các hoạt động tiếp theo tại các bến cảng. Các quyết định phân bổ bến thường được tích hợp với các quyết định phân công cầu quay và / hoặc lập lịch trình. Tuy nhiên, chất lượng của các quyết định này bị ảnh hưởng rất nhiều bởi các sự kiện không thể đoán trước có thể xảy ra thường xuyên.

Sự cạnh tranh giữa các bến container đã gia tăng do sự tăng trưởng đáng kể về lượng container chính bằng đường biển các tuyến đường. Trước thách thức gay gắt, để thu hút thêm các hãng tàu, các nhà khai thác cảng container đã cố gắng cung cấp các dịch vụ hậu cần chuyên sâu hơn và trong khi đó, để giảm chi phí bằng cách sử dụng hiệu quả các nguồn lực, bao gồm cả nguồn nhân lực, bến, bãi container và các thiết bị xếp dỡ container khác nhau. Trong số tất cả các nguồn lực, cầu cảng là quan trọng nhất nguồn lực và lập lịch trình bến tốt cải thiện sự hài lòng của khách hàng và tăng thông lượng cảng, dẫn đến doanh thu cao hơn cảng (Kim và Moon, 2003). Hình 1 dưới đây minh họa một tàu cập bến tại Tân Cảng.

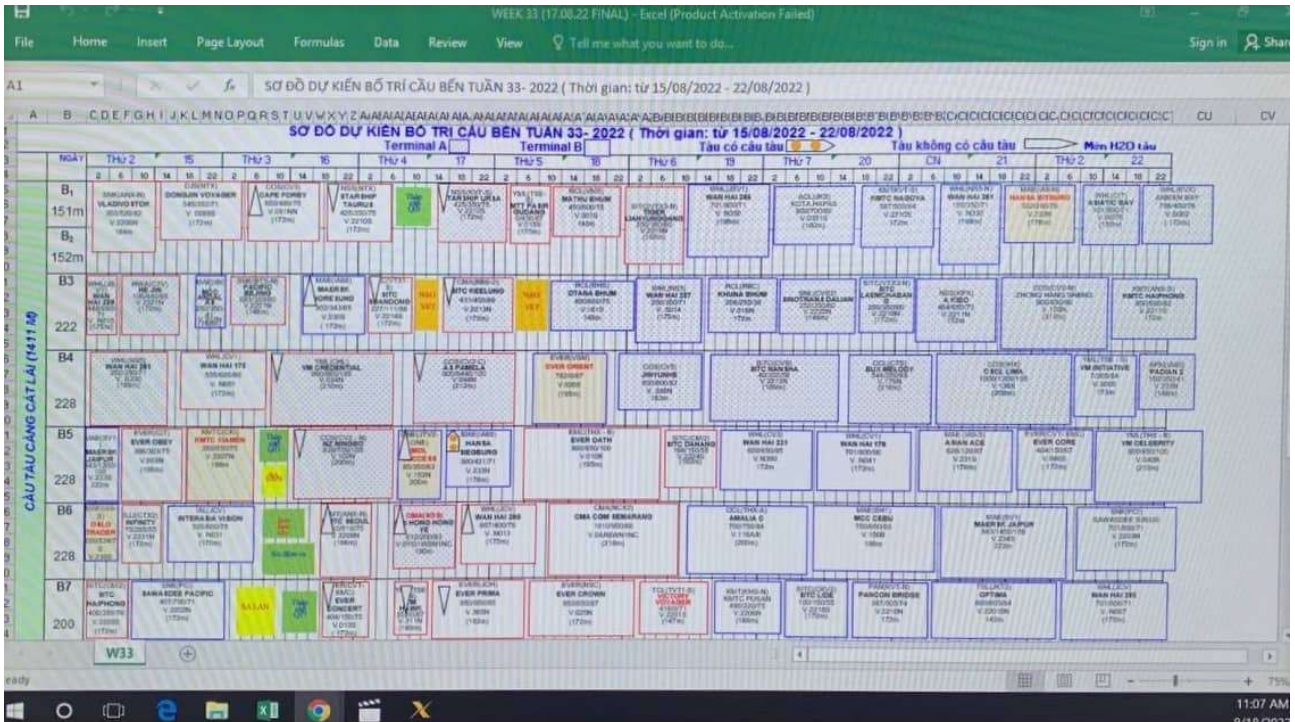


Hình 1: Tàu cập bến tại Tân Cảng

Trong những thập kỷ qua, đã có một số nghiên cứu về cách phân bổ bến cho các tàu đến. Về cơ bản, bài toán phân bổ bến (BAP) có thể được phân loại thành hai lớp: rời rạc (discrete berth allocation problem - BAPD) và liên tục (continuous berth allocation problem- BAPD). Đối với phiên bản rời rạc - BAPD, cầu cảng được xem như một tập hợp hữu hạn các bến. Thông thường, một bến chỉ có thể phục vụ một tàu tại một thời điểm. Ngược lại, mô hình liên tục BAPC cho phép tàu cập bến ở bất kỳ đâu dọc theo cầu cảng để sử dụng đầy đủ tài nguyên cầu cảng. Một tiêu chí khác để phân loại BAP dựa trên việc tất cả các tàu đã đến bến hay chưa trước khi bắt đầu lập kế hoạch cập bến (static/dynamic).

Trong khuôn khổ bài nghiên cứu này, chúng ta sẽ tìm hiểu về bài toán phân bổ tại bến tàu và xếp lịch cập bến cho các tàu. Thông tin đến cảng cũng như cá thông số của các tàu đã được xác định trước. Tại Tân Cảng hiện tại, Hình 2 minh họa ứng dụng dùng để quản lý việc phân bổ tại bến tàu hiện tại và cần dùng rất nhiều nhân lực để tính toán sắp xếp và theo dõi để tránh mâu thuẫn xảy ra giữa các tàu khi cập bến và tối ưu hóa

tài nguyên.



Hình 2: Giao diện ứng dụng quản lý xếp lịch tàu cập bến tại Tân Cảng

Theo [4], vấn đề quan tâm là tìm cách tự động hóa, giảm nhân công và tính toán tối ưu việc phân bổ không gian tại bến cho các tàu để giao nhận và chuyển đổi các container - mà ta gọi là bài toán phân bổ tại bến tàu. Do không gian bến ở tại các bến tàu đều rất hạn chế và hàng nghìn container phải được xử lý mỗi ngày, nên việc phân bổ bến hiệu quả là rất quan trọng để quản lý hiệu quả luồng giao thông container. Một bến lớn điển hình tại các bến container là có thể tiếp nhận nhiều tàu cùng một lúc. Khi không còn chỗ đậu, tàu cần chờ neo đậu ở ngoài biển khơi và chờ đến khi được cấp phát không gian tại bến để cập vào cảng. Để đơn giản, chúng ta gọi tổng thời gian chờ và thời gian xử lý của một tàu là thời gian lưu chuyển của nó. Mục tiêu chính yếu của bài toán là phân bổ không gian bến cho các tàu và lên lịch cho các tàu sao cho tổng thời gian dòng chảy (flowtime) có trọng số được giảm thiểu, trong đó trọng lượng phản ánh tầm quan trọng tương đối của các tàu.

Một số giả định trong mô hình:

1. Các tàu không cần phải chờ khi không gian bến còn chỗ và không thể neo đậu song song về mặt không gian tại bến cảng. Nghĩa là, nếu không gian tại bến tàu đủ lớn thì vẫn có thể có trường hợp xảy ra là hai hay nhiều tàu cập bến cùng một thời điểm và đậu kế cận nhau mà không làm ảnh hưởng gì đến nhau. Tuy nhiên, tại một vị trí của bến tàu và tại một thời điểm thì chỉ có tối đa một tàu được cập bến.
2. Chiều sâu mực nước tại bến cảng đủ điều kiện đảm bảo cho các tàu với mọi kích thước đều có thể cập bến.
3. Một khi tàu được neo đậu, nó sẽ ở nguyên vị trí của nó cho đến khi tất cả các quy trình xử lý container cần thiết được thực hiện.
4. Thời gian xử lý đảo chuyển các container tại các tàu cập bến là có thể xác định một cách tự động trước. Hay nói cách khác, thông số đầu vào có thể chứa thông tin về thời gian xử lý (processing time  $p_i$ ) khi tàu được cập bến.

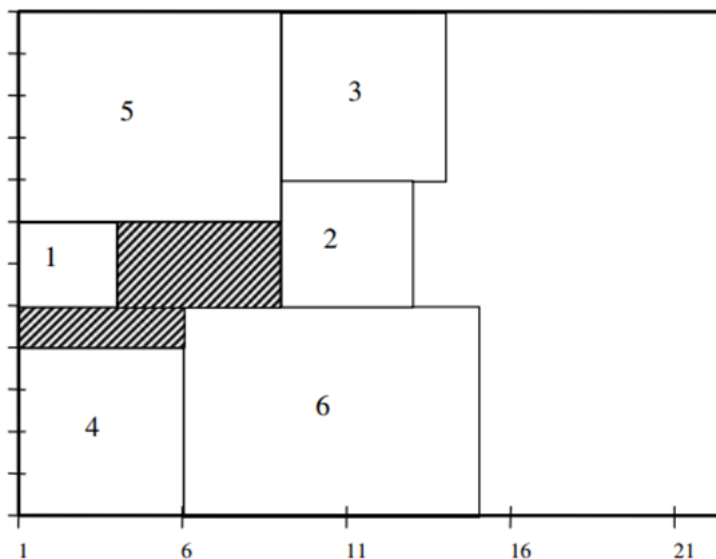
### 3 Mô tả bài toán

Như vậy, bài toán xếp lịch cho bến tàu là bài toán đi tìm một lập lịch thỏa mãn một số ràng buộc về mặt không-thời gian của các tàu với mục tiêu là cực tiểu hóa tổng thời gian dòng chảy có trọng số.



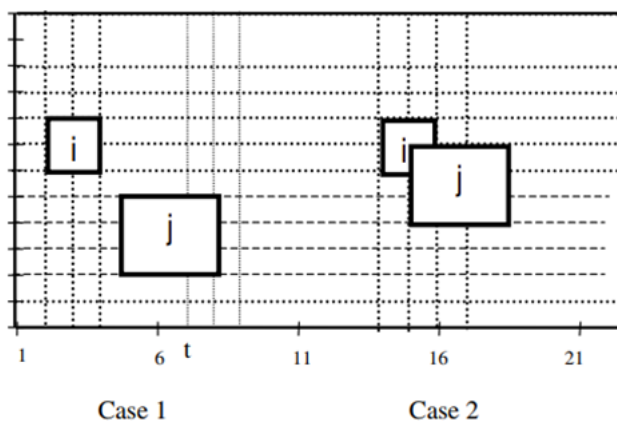
### 3.1 Biểu đồ không-thời gian

Bài toán phân bổ bến có thể được biểu diễn bằng biểu đồ không gian thời gian như Hình 3 trong đó trục hoành và trục tung lần lượt thể hiện các đơn vị thời gian và phần bến. Một tàu có thể được xem như một hình chữ nhật có chiều dài là thời gian xử lý và chiều cao là kích thước tàu. Ta gọi tàu  $i$  đang neo đậu tại đoạn bến tại thời điểm  $u_i$ , và khi đó tàu chiếm các đoạn bến liên tiếp giữa  $v_i$  và  $v_i + s_i - 1$  và từ đơn vị thời gian  $u_i$  đến  $u_i + p_i - 1$  (tham khảo mô tả chi tiết các thông tin tại mục bên dưới).



Hình 3: Minh họa về biểu đồ không-thời gian của một lời giải khả thi

Hình 4 mô tả rõ hơn một cách trực quan giữa 2 trường hợp cho quyết định phân bổ tài nguyên không-thời gian cho tàu  $i$  và tàu  $j$ : (Case 1) là một đề xuất khả thi trong thực tế còn (Case 2) là không khả thi.



Hình 4: Phân biệt lời giải khả thi và không khả thi

### 3.2 Mô hình hóa toán học cho bài toán

Đối với bài toán xếp lịch tại bến cảng với lịch tàu cho trước (trong các phần sau, tên bài toán sẽ được gọi tắt là BAP - static berth allocation problem in a continuous-segments area). Bài nghiên cứu này tuân theo công thức toán học trong [Guan and Cheung, 2004] và [Lee et al., 2010]. Trong công thức này, việc dịch chuyển tàu không được xem xét vì sự gián đoạn của việc xếp dỡ container thường gây tốn kém.

### 3.2.1 Các thông số

- $S$ : chiều dài của bến
- $K$ : số lượng điểm ngắt quãng của bến
- $b_k$ : vị trí ngắt quãng thứ  $k$ ,  $\forall 1 \leq k \leq K$
- $T$ : tổng thời lượng cần xếp lịch bến tàu
- $N$ : số lượng tàu cập bến cần xếp lịch
- $p_i$ : thời gian xử lý đảo chuyển các container của tàu thứ  $i$ ,  $\forall 1 \leq i \leq N$
- $s_i$ : kích thước của tàu thứ  $i$ ,  $\forall 1 \leq i \leq N$
- $a_i$ : thời điểm đến bến của tàu thứ  $i$ ,  $\forall 1 \leq i \leq N$
- $w_i$ : trọng số mô tả mức độ quan trọng của tàu thứ  $i$ ,  $\forall 1 \leq i \leq N$

### 3.2.2 Biến ra quyết định

- $u_i$ : thời điểm được bắt đầu neo đậu của tàu  $i$ ,  $\forall 1 \leq i \leq N$ ,
- $v_i$ : vị trí cập bến của của tàu  $i$ ,  $\forall 1 \leq i \leq N$ ,

### 3.2.3 Biến trung gian

- $c_i$ : thời điểm rời bến của tàu  $i$ ,  $\forall 1 \leq i \leq N$ .
- $\sigma_{ij}$ : 1, nếu tàu  $i$  được kết thúc bên trái của tàu  $j$  trong biểu đồ không-thời gian; và ngược lại thì sẽ bằng 0,  $\forall 1 \leq i, j \leq N, i \neq j$
- $\delta_{ij}$ : 1, nếu tàu  $i$  is được thực hiện bên dưới tàu  $j$  trong biểu đồ không-thời gian; và ngược lại thì sẽ bằng 0,  $\forall 1 \leq i, j \leq N, i \neq j$
- $\gamma_{ik}$ : 1, nếu  $v_i \geq b_k$  (tàu  $i$  cập bến tại vị trí sau ngắt quãng  $k$ ); và ngược lại thì sẽ bằng 0,  $\forall 1 \leq i \leq N, \forall 1 \leq k \leq K$ .

### 3.2.4 Mô hình toán học

Hàm mục tiêu (objective function) là tối thiểu hóa tổng thời lượng chờ có hệ số ưu tiên của các tàu tại bến (minimizing the sum of weighted turnaround time for each incoming vessel). Các ràng buộc (2) và (3) dùng để mô tả định nghĩa của các biến ra quyết định  $\sigma_{ij}$  và  $\delta_{ij}$ . Các ràng buộc (4)–(6) đảm bảo rằng tàu  $i$  và tàu  $j$  không được phép neo đậu chồng chéo nhau kể cả không gian và thời gian. Các ràng buộc (7) biểu diễn mối quan hệ giữa thời điểm rời bến và thời điểm cập bến neo đậu lại của mỗi tàu. Các ràng buộc (8) và (9) định nghĩa miền giá trị của các biến ra quyết định  $u_i$ ,  $v_i$ ;  $\sigma_{ij}$ , and  $\delta_{ij}$ .

Các ràng buộc (10), (11) và (12) trình bày định nghĩa của các biến trung gian  $\gamma_{ik}$ . Các ràng buộc (13) và (14) cho mô tả về các ngắt quãng của bến tàu. Mô tả về mặt toán học cho bài toán CSBAP có thể được phát biểu như sau.

- Hàm mục tiêu (objective function):

$$\min \sum_{i=1}^N w_i(c_i - a_i) \quad (1)$$

- Các ràng buộc (constraints):

- Mối quan hệ giữa 2 tàu  $i$  và  $j$  theo các biến ra quyết định  $\sigma_{ij}$  and  $\delta_{ij}$ :  
Nếu  $\sigma_{ij} = 1$  thì tàu  $i$  phải được neo đậu vào bến trước tàu  $j$ :

$$u_j - u_i - p_i - (\sigma_{ij} - 1).T \geq 0, \quad \forall 1 \leq i, j \leq N, i \neq j \quad (2)$$

- Nếu  $\delta_{ij} = 1$  thì tàu  $i$  phải được neo đậu tại vị trí có giá trị nhỏ hơn vị trí neo đậu của tàu  $j$ :

$$v_j - v_i - s_i - (\delta_{ij} - 1).S \geq 0, \quad \forall 1 \leq i, j \leq N, i \neq j \quad (3)$$

- Tàu  $i$  và tàu  $j$  không được có quyết định chồng chéo nhau trong biểu đồ không-thời gian:

$$\sigma_{ij} + \sigma_{ji} \leq 1, \quad \forall 1 \leq i, j \leq N, i \neq j \quad (4)$$

$$\delta_{ij} + \delta_{ji} \leq 1, \quad \forall 1 \leq i, j \leq N, i \neq j \quad (5)$$

$$\sigma_{ij} + \sigma_{ji} + \delta_{ij} + \delta_{ji} \geq 1, \quad \forall 1 \leq i, j \leq N, i \neq j \quad (6)$$

- Mỗi quan hệ giữa thời điểm rời bến ( $c_i$ ) và thời điểm bắt đầu neo đậu vào bến ( $u_i$ ) của mỗi tàu đều cần tuân thủ:

$$p_i + u_i = c_i, \quad \forall 1 \leq i, j \leq N, i \neq j \quad (7)$$

- Miền giá trị của các biến ra quyết định  $u_i$ ,  $v_i$ ;  $\sigma_{ij}$ , và  $\delta_{ij}$ :

$$a_i \leq u_i \leq (T - p_i), \quad 0 \leq v_i \leq (S - s_i), u_i, \forall v_i \in \mathbb{R}^+, \forall 1 \leq i \leq N \quad (8)$$

$$\sigma_{ij}, \delta_{ij} \in \{0, 1\}, \quad \forall 1 \leq i, j \leq N, i \neq j \quad (9)$$

$$b_k \cdot \gamma_{ik} \leq v_i + s_i, \quad \forall 1 \leq i \leq N, \forall 1 \leq k \leq K \quad (10)$$

- Các ràng buộc của biến trung gian  $\gamma_{ik}$ :

$$b_k - v_i + S \cdot \gamma_{ik} \geq 0, \quad \forall 1 \leq i \leq N, \forall 1 \leq k \leq K \quad (11)$$

$$\gamma_{ik} \in \{0, 1\}, \quad \forall 1 \leq i \leq N, \forall 1 \leq k \leq K \quad (12)$$

- Ràng buộc có các ngắt quãng tại bến:

Nếu  $v_i \geq b_k$  (tàu  $i$  cập bến tại vị trí sau ngắt quãng  $k$ ) thì  $\gamma_{ik}$  phải thỏa mãn

$$v_i - b_k \cdot \gamma_{ik} \geq 0, \quad \forall 1 \leq i \leq N, \forall 1 \leq k \leq K \quad (13)$$

Nếu  $v_i < b_k$  (tàu  $i$  cập bến tại vị trí trước ngắt quãng  $k$ ) thì  $\gamma_{ik}$  phải thỏa mãn sao cho tàu  $i$  khi neo không có chạm vào ngắt quãng  $k$

$$v_i - b_k - (1 - \gamma_{ik}) \cdot S \leq 0, \quad \forall 1 \leq i \leq N, \forall 1 \leq k \leq K \quad (14)$$

## 4 Giải pháp đề xuất

Theo [D-H.Lee et al, 2010] ta có một phương án tìm ra các điểm đặt tàu chấp nhận được dựa theo việc phân tích các điểm vào các Class điểm. Tuy nhiên, chúng ta không thể ứng dụng trực tiếp các làm đó vào đây, vì trên nơi neo đậu tàu, chúng ta có những break không thể đặt tàu chồng qua các điểm đó.

Về phân lớp của mỗi điểm, chúng ta giữ lại định nghĩa của phân lớp của các điểm như trong [D-H.Lee et al, 2010]. Tuy nhiên sau bước khởi tạo các điểm vào các class ban đầu, chúng ta thêm một số các quy tắc chuẩn hóa như sau:

- Đối với Class 2:

- Nếu giá trị của điểm đó là  $[1,0,0,1]$  và điểm đó nằm trên break: Tiến hành nhân đôi điểm đó và thêm vào Class 1 với hai giá trị  $[1,0,0,0]$  và  $[0,0,0,1]$ .
- Nếu giá trị của điểm đó là  $[1,0,1,0]$ : Tiến hành nhân đôi điểm đó và thêm vào Class 1 với hai giá trị là  $[1,0,0,0]$  và  $[0,0,1,0]$ .
- Nếu giá trị của điểm đó là  $[0,1,0,1]$ : Tiến hành nhân đôi điểm đó và thêm vào Class 1 với hai giá trị là  $[0,1,0,0]$  và  $[0,0,0,1]$ .

- Đối với Class 3:

- Các giá trị của điểm đó là  $[1,0,1,1]$  và nằm trên break: Tiến hành thêm điểm đó vào Class 1 với giá trị là  $[1,0,0,0]$ .



- Các giá trị của điểm đó là  $[0,1,1,1]$  và nằm trên break: Tiến hành thêm điểm đó vào Class 1 với giá trị là  $[0,1,0,0]$ .
- Các giá trị của điểm đó là  $[1,1,0,1]$  và nằm trên break: Tiến hành thêm điểm đó vào Class 1 với giá trị là  $[0,0,0,1]$ .
- Các giá trị của điểm đó là  $[1,1,1,0]$  và nằm trên break: Tiến hành thêm điểm đó vào Class 1 với giá trị là  $[0,0,1,0]$ .

For nodes in $I_1$		For nodes in $I_2$	
<b>a</b>	<b><math>b_{I_1}</math></b>	<b><math>a_1, a_2</math></b>	<b><math>b_{I_2}</math></b>
$[0,1,1,1]$	$\Rightarrow$	$[0,0,0,1]$	
$[1,0,1,1]$	$\Rightarrow$	$[1,0,0,0]$	$\Rightarrow$
$[1,1,0,1]$	$\Rightarrow$	$[0,0,0,1]$	$\Rightarrow$
$[1,1,1,0]$	$\Rightarrow$	$[1,0,1,1], [1,1,1,0]$	$[1,0,0,0]$
		$[1,1,0,1], [0,1,1,1]$	$[0,0,0,1]$

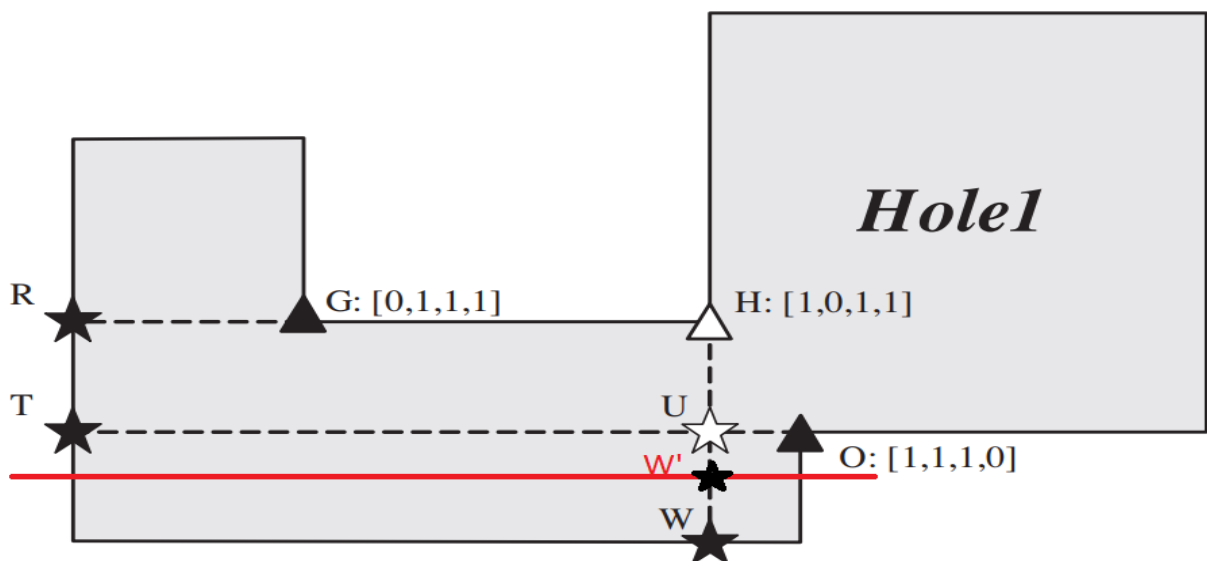
Hình 5: Phân lớp các điểm trong bài báo

Class 2	<b>b</b>	Class 3	<b>b</b>
<b>a</b>		<b>a</b>	
$[1, 1, 0, 0]$	$[1, 0, 0, 0], [0, 1, 0, 0]$	$[1, 1, 1, 0]$	$[1, 0, 0, 0]$
$[1, 0, 1, 0]$	$[1, 0, 0, 0], [0, 0, 1, 0]$	$[1, 1, 0, 1]$	$[1, 0, 0, 0]$
$[1, 0, 0, 1]$	$[1, 0, 0, 0], [0, 0, 0, 1]$	$[1, 0, 1, 1]$	$[1, 0, 0, 0]$
$[0, 1, 1, 0]$	$[0, 1, 0, 0], [0, 0, 1, 0]$	$[0, 1, 1, 1]$	$[0, 1, 0, 0]$
$[0, 1, 0, 1]$	$[0, 1, 0, 0], [0, 0, 0, 1]$		
$[0, 0, 1, 1]$	$[0, 0, 1, 0], [0, 0, 0, 1]$		

Hình 6: Phân lớp các điểm nhóm định nghĩa

Với việc xác định các điểm chiếu ta và các điểm thuộc Class 3 nhưng không nằm trên các đường break ta giữ nguyên các chuẩn hóa như cũ.

Xác định các giao điểm của các điểm thuộc Class 3 nhưng nằm trên break khi mang giá trị  $[1,0,1,1]$  và  $[1,1,0,1]$ : Ta chỉ xét giữa 2 đường break mà điểm đó thuộc về.



Hình 7: Xác định giao điểm với break

Ví dụ như trong hình 7, giả sử có một đường break (màu đỏ) cắt ngang đường UW, khi đó gọi giao điểm là W'. Ta sẽ đưa điểm U về điểm W' thay vì điểm W như cách làm thông thường.

Tới đây, để giải bài toán chúng ta sẽ chia thành hai phase:

- Construction Phase
- Optimize Phase

## 4.1 Construction Phase

Ở giai đoạn này, chúng ta sẽ xây dựng một lời giải ban đầu (initial feasible solution) của bài toán.

Ý tưởng chính: Giải bài toán theo nguyên tắc *First come first serve*. Chúng ta sẽ sắp xếp những con tàu theo thời gian đến của nó vào những vị trí thích hợp.

Cụ thể các bước giải như sau:

- Bước 1: Đọc dữ liệu từ file đầu vào, tiến hành sắp xếp thứ tự các tàu theo thời gian đến, với hai con tàu cùng thời gian đến ta sẽ ưu tiên con tàu có trọng lượng lớn hơn
- Bước 2: Xác định trạng thái hiện tại của các con tàu đã neo đậu trước đó ( $n - 1$  con tàu), tiến hành tìm ra các điểm phù hợp để đậu con tàu tiếp theo thứ  $n$ . Và tính toán chi phí tại các điểm đó
- Bước 3: Lựa chọn vị trí con tàu thứ  $n$  được đặt theo hai phương án:
  - Phương án 1: Roulette wheel selection method
  - Phương án 2: Best-fit. Được hiểu ở đây chúng ta sẽ chọn vị trí có cost ít nhất, trong các trường hợp có cùng cost, ta sẽ chọn đặt con tàu vào vùng mà chênh lệch giữa hai break của vùng đó với kích thước con tàu là nhỏ nhất
- Bước 4: Lặp lại bước 3 cho đến khi hoàn thành tất cả các con tàu.

## 4.2 Optimize Phase

Bằng thực nghiệm cho thấy, phương pháp best-fit đem lại hiệu quả rõ rệt so với Roulette wheel selection method, do đó chúng ta sẽ sử dụng phương pháp best-fit để tiếp tục vào optimize phase.

Mục đích chính của phase này sẽ là tìm kiếm một lời giải tốt hơn cho lời giải hiện tại của chúng ta. Ý tưởng chính: Đổi chỗ các con tàu có khối lượng (weight) lên sớm nhất có thể.

Ở đây nhóm có đề xuất hai phương pháp optimize.

### 4.2.1 Swap in break method

Chi tiết các bước làm như sau:

- Bước 1: Xác định các con tàu ở construction phase thuộc về break nào.
- Bước 2: Sắp xếp các con tàu theo từng nhóm break và giảm dần theo mooring time.
- Bước 3: Với mỗi con tàu, ta duyệt qua các con tàu có mooring-time nhỏ hơn và cùng break. Kiểm tra xem có thể đổi chỗ không và sau khi đổi. Điều kiện để có thể đổi chỗ 2 con tàu:
  - Arrival time của tàu được dời lên sớm phải nhỏ hơn mooring time của tàu bị dời lui sau. Hay nói cách khác, con tàu được dời lên sớm phải đến trước khi con tàu bị đổi được neo đậu.
  - Không gian xung quanh của hai con tàu phải đủ để chứa con tàu còn lại
- Bước 4: Xác định con tàu đổi có cost nhỏ nhất không âm. Tiến hành đổi chỗ con tàu hiện tại chọn với con tàu đã chọn.
- Bước 5: Lặp lại bước 3 với tất cả các con tàu.

#### 4.2.2 Swap all method

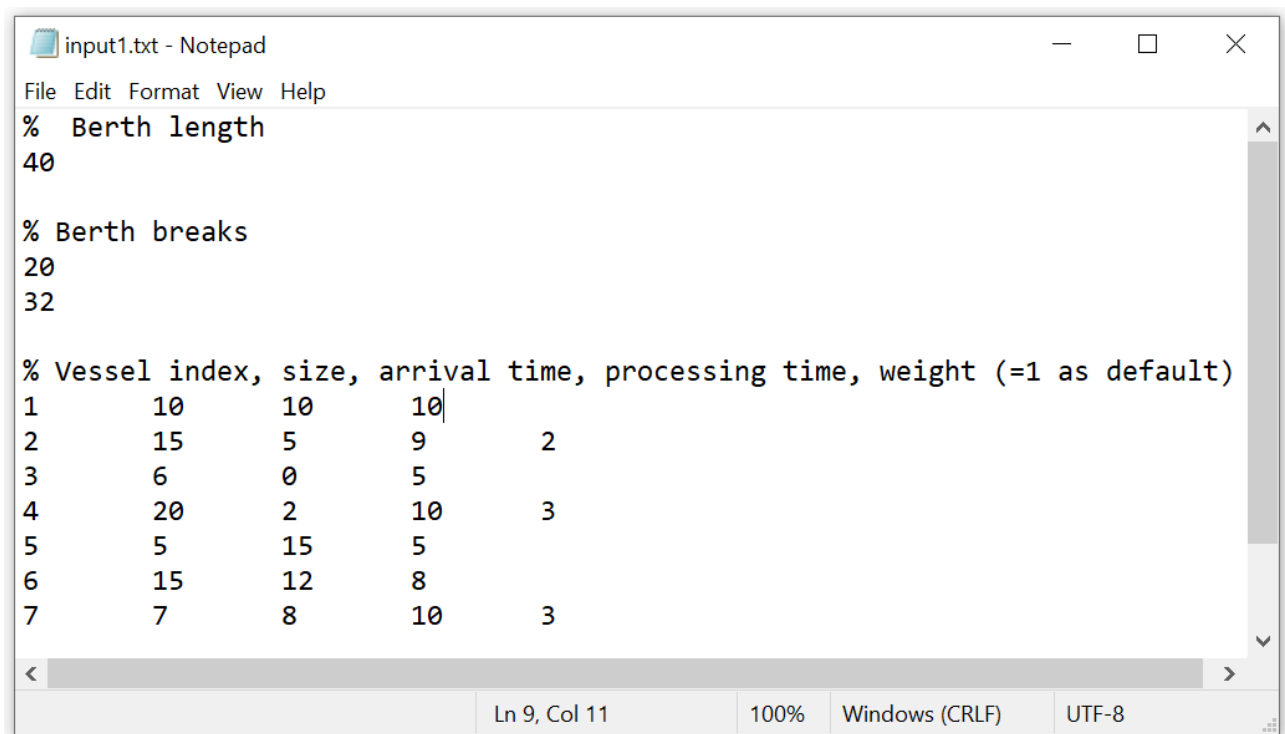
Chi tiết các bước làm như sau:

- Bước 1: Sắp xếp các con tàu giảm dần theo mooring time.
- Bước 2: Với mỗi con tàu, ta duyệt qua các con tàu có mooring-time nhỏ hơn (kể cả không cùng break). Kiểm tra xem có thể đổi chỗ không và sau khi đổi. Điều kiện để có thể đổi chỗ 2 con tàu:
  - Arrival time của tàu được dời lên sớm phải nhỏ hơn mooring time của tàu bị dời lui sau. Hay nói cách khác, con tàu được dời lên sớm phải đến trước khi con tàu bị đổi được neo đậu.
  - Không gian xung quanh của hai con tàu phải đủ để chứa con tàu còn lại
- Bước 3: Xác định con tàu đổi có cost nhỏ nhất không âm. Tiến hành đổi chỗ con tàu hiện tại chọn với con tàu đã chọn.
- Bước 4: Lặp lại bước 2 với tất cả các con tàu.

## 5 Định dạng nhập-xuất và kết quả thực nghiệm

### 5.1 Dữ liệu đầu vào

Hình 5.1 mô tả một mẫu ví dụ đầu vào mà trong đó có cung cấp các thông số cần thiết để tính toán bao gồm: tổng tài nguyên được cấp phát tại bến tàu có 40 đơn vị không gian; bến tàu có 2 điểm ngắt quãng tại vị trí 20 và 32; dự kiến có 7 con tàu cần được xếp lịch và phân bổ vị trí đậu.



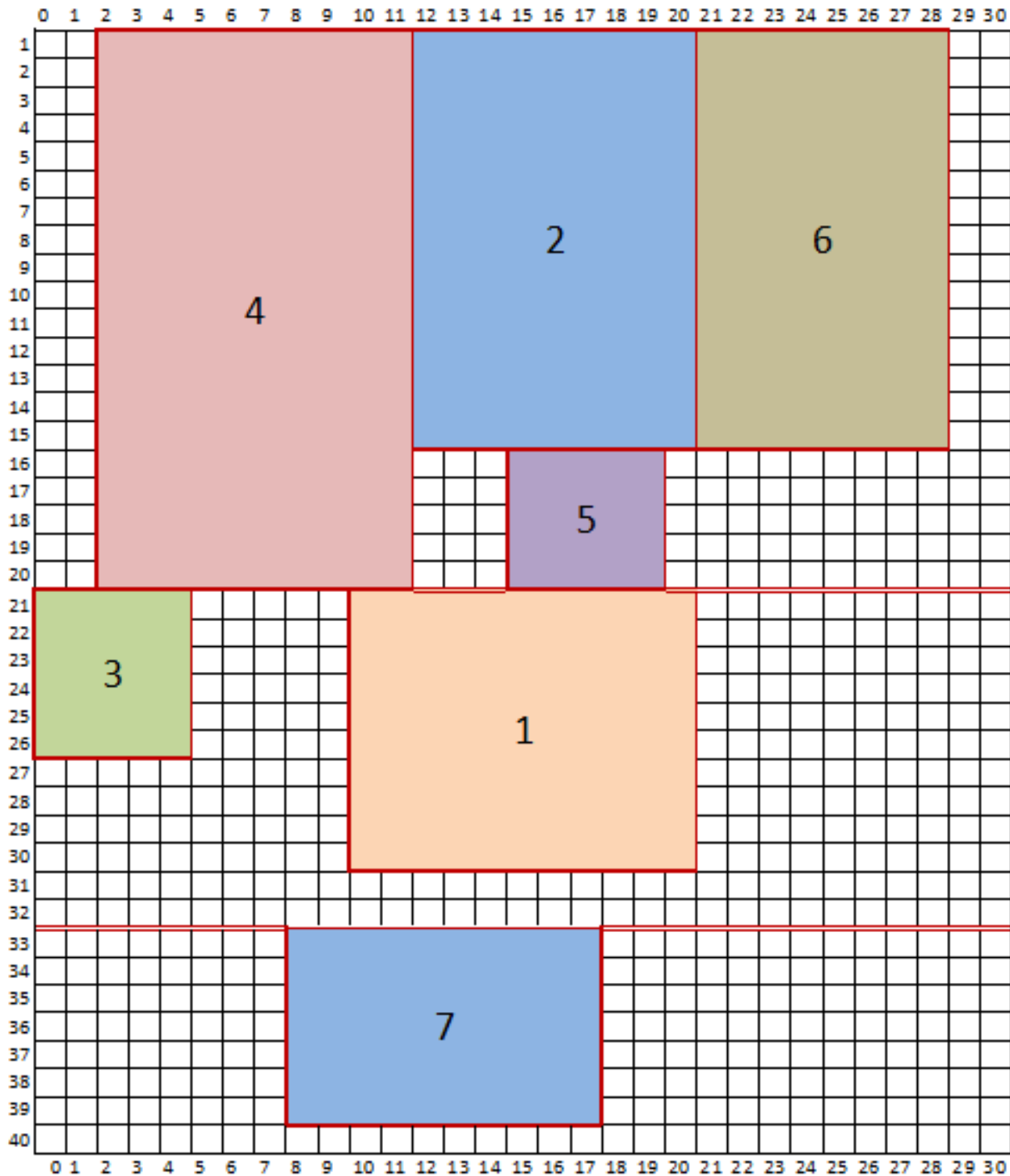
```
input1.txt - Notepad
File Edit Format View Help
% Berth length
40

% Berth breaks
20
32

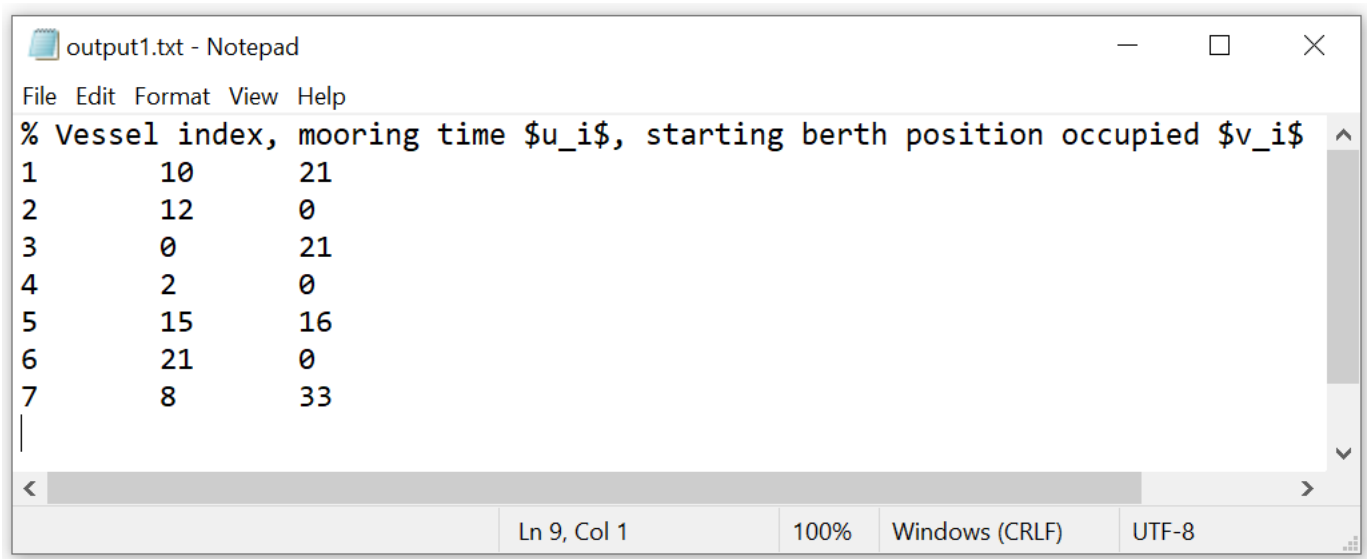
% Vessel index, size, arrival time, processing time, weight (=1 as default)
1      10      10      10
2      15      5       9      2
3       6       0       5
4      20      2      10      3
5       5      15       5
6      15      12       8
7       7       8      10      3
```

Hình 8: Mẫu dữ liệu đầu vào của bài toán nghiên cứu

Một lời giải khả thi có thể được xác định như Hình 5.1 và kết quả cần xuất tương ứng sẽ có định dạng như Hình 5.1. Kết quả thu được từ hàm mục tiêu là 23.



Hình 9: Một lời giải khả thi trên biểu đồ không-thời gian



Hình 10: Mẫu dữ liệu xuất kết quả

## 6 Kết quả thực nghiệm

## 6.1 Yêu cầu công việc

Mỗi nhóm, từ 3 đến 5 sinh viên, đề xuất giải pháp để giải bài toán trên. Nhóm cần nộp báo cáo trình bày về thuật giải đề xuất và kết quả thực nghiệm. Đồng thời, nhóm cũng cần nộp source code, và trình bày công trình của mình trong khoảng 15 minutes. Báo cáo và slide trình bày cần được viết dưới dạng LaTeX. **Hạn chót nộp báo cáo và sản phẩm demo: 12/11/2022.**

## 6.2 Đánh giá kết quả

Cấu hình máy sử dụng:

- CPU: AMD Ryzen 7
- RAM: 8GB
- OS: Window 11

Yêu cầu thuật toán cần xử lý tối đa là **5 phút**, cho mỗi trường hợp cụ thể của bài toán. Dữ liệu kiểm thử sẽ được tạo ngẫu nhiên, gồm 2 mẫu thử (testset) mỗi mẫu sẽ có 5 test case với thông số tương ứng trong hình 12.

### 6.2.1 Construction phase

Testcase			Test Set 1				Test Set 2			
			Roulette wheel	Time	Best Fit	Time	Roulette wheel	Time	Best Fit	Time
Test 1	Berth length	100	31225	0.298s	812	0.04394s	4692	0.069975s	177	0.026019s
	Number of berth break	4								
	Number of vessel	16								
Test 2	Berth length	150	139056	7.49725s	46957	1.312s	233425	3.96673s	38980	1.143s
	Number of berth break	6								
	Number of vessel	75								
Test 3	Berth length	200	175249	13.5981s	62358	2.52999s	364924	9.10627s	183650	6.60275s
	Number of berth break	8								
	Number of vessel	100								
Test 4	Berth length	500	541957	99.8938s	86781	18.8861s	1259249	74.8681s	595978	56.3091s
	Number of berth break	20								
	Number of vessel	200								
Test 5	Berth length	500	3173206	180.222s	1409197	133.112s	2997296	172.24s	1014132	98.1615s
	Number of berth break	30								
	Number of vessel	300								

Hình 11: Kết quả của construction phase

#### Nhận xét:

- Trường hợp *best-fit* cho kết quả tốt hơn khá rõ rệt so với *roulette wheel selection method*. Điều này cũng khá dễ lý giải vì lý do: Trường hợp sử dụng phương pháp chọn điểm ngẫu nhiên, có thể phương pháp đó sẽ chọn vào những điểm làm cho mốc thời gian dài ra một cách đáng kể thay vì đưa ra lời giải con tàu neo đậu vào những chỗ còn đang trống giữa các con tàu.
- Về thời gian chạy, ta có thể thấy không thể khẳng định rằng phương pháp nào có thể đưa đến thời gian chạy.  
Lý do: Tùy thuộc vào từng test case sẽ đưa ra những điểm đặt tàu khác nhau và sinh ra những loại điểm khác nhau nên chưa thể so sánh được.

### 6.2.2 Optimize phase

Testcase			Test Set 1				Test Set 2			
			SwapAll	Time	SwapInBreak	Time	SwapAll	Time	SwapInBreak	Time
Test 1	Berth length	100	528	0.044s	528	0.045995s	177	0.030999s	177	0.028766s
	Number of berth break	4								
	Number of vessel	16								
Test 2	Berth length	150	43369	1.42975s	46000	1.98616s	38555	1.17908s	38700	1.148s
	Number of berth break	6								
	Number of vessel	75								
Test 3	Berth length	200	55773	2.59263s	56559	2.539s	169099	6.83729s	169147	6.70774s
	Number of berth break	8								
	Number of vessel	100								
Test 4	Berth length	1000	78823	18.9721s	84307	19.404s	527259	58.1569s	559353	56.7922s
	Number of berth break	20								
	Number of vessel	200								
Test 5	Berth length	1200	1302636	133.95s	1307161	135.849s	920660	99.0425s	931253	99.9921s
	Number of berth break	40								
	Number of vessel	250								

Hình 12: Kết quả của optimize phase

#### Nhận xét:

- Ở tất cả các testcase đều cho thấy sự cải thiện của thuật toán optimize (mặc dù không quá nhiều). Tuy nhiên việc chênh lệch thời gian so với construction phase không quá đáng kể nên có thể chấp nhận được.



- Có thể thấy được phương pháp SwapAll sẽ cho được kết quả cải thiện hơn so với SwapInBreak.  
Lý do: việc chỉ cải thiện trên mỗi break chỉ giới hạn được các tàu có trọng số lớn di chuyển đến một vị trí nào đó trong break, tuy nhiên khi áp dụng trên toàn bộ khu neo đậu, ta có thể đưa các tàu đó đi lên những vị trí tốt hơn.

## 7 Kết luận

Đây là một bài toán ví dụ trong số các bài toán tối ưu chung quanh chúng ta. nếu chúng ta có thể xác định được các bài toán này, và đề xuất được các thuật giải/giải pháp tìm ra đáp án tốt cho bài toán, điều này sẽ giúp cho các công việc hàng ngày của chúng ta sẽ được thực hiện trôi chảy và hiệu quả hơn. Hy vọng thông qua việc tìm hiểu và giải bài toán này, chúng ta sẽ hiểu hơn về các thuật toán ứng dụng trong công nghiệp cũng như trong các bài thực tế quanh ta; và hy vọng trong một tương lai gần, các bạn có cơ hội và có thể đề xuất các giải pháp tốt cho các bài toán hỗ trợ ra quyết định. Chúc các bạn thành công.

## Tài liệu

- [1] D.-H. Lee, J.H. Chen, J.X. Cao (2010) The continuous berth allocation problem: A greedy randomized adaptive search solution. **Transportation Research Part E** 46, 1017 – 1029.
- [2] G. Giallombardo, L. Moccia, M. Salani, I. Vacca (2010) Modeling and solving the tactical berth allocation problem. **Transportation Research Part B** 44, 232 – 245.
- [3] A. Lim (1998) The berth planning problem. **Operations Research Letters** 22, 105 – 110.
- [4] Y. Guan, R.K. Cheung (2004) The berth allocation problem: models and solution methods **OR Spectrum** 26, 75 – 92.
- [5] C. Bierwirth, F. Meisel (2010) A survey of berth allocation and quay crane scheduling problems in container terminals. **European Journal of Operational Research** 202, 615 – 627.
- [6] J.R. Correcher, T.V.d Bossche, R. Alvarez-Valdes (2019) The berth allocation problem in terminals with irregular layouts. **European Journal of Operational Research** 272(3), 1096 – 1108.
- [7] K. Artiouchine, Ph. Baptiste, J. Mattioli (2008) The K King Problem, an Abstract Model for Computing Aircraft Landing Trajectories: On Modeling a Dynamic Hybrid System with Constraints. *INFORMS Journal on Computing* 20 (2) 222 – 233.
- [8] Ph. Baptiste, R. Sadykov (2010) Time-indexed formulations for scheduling chains on a single machine: An application to airborne radars. *European Journal of Operational Research* 203(2), 476 – 483.
- [9] Ph. Baptiste, M. Flamini, F. Sourd (2008) Lagrangian bounds for just-in-time job-shop scheduling. *Computers & Operations Research* 35(3), 906 – 915.
- [10] J. Blazewicz, P. Formanowicz, M. Kasprzak, P. Schuurman, G.J. Woeginger (2007) A polynomial time equivalence between DNA sequencing and the exact perfect matching problem. *Discrete Optimization* 4(2), 154 – 162.
- [11] E.G. Coffman Jr., D. Matsypura, V. G. Timkovsky (2010) Strategy vs risk in margining portfolios of options. *4OR* 8(4), 375 – 386.
- [12] J. Józefowska (2007) Just-in-time scheduling : models and algorithms for computer and manufacturing systems. Springer.
- [13] Y. Kergosien, C. Lenté, D. Piton, J.-C. Billaut (2011) A tabu search heuristic for the dynamic transportation of patients between care units. *European Journal of Operational Research* 214(2), 442 – 452.
- [14] Y. Kergosien, J.-F. Tournamille, B. Laurence, J.-C. Billaut (2011) Planning and tracking chemotherapy production for cancer treatment: A performing and integrated solution. *International Journal of Medical Informatics* 80(9), 655 – 662.
- [15] J.Y-T. Leung. (2004) Handbook of scheduling : algorithms, models, and performance analysis. Computer and information science series, Chapman and Hall/CRC (ed.), Boca Raton, Florida.
- [16] J.-L. Lutton, D. Nace, J. Carlier (2000) Assigning spare capacities in mesh survivable networks. *Telecommunication Systems* 13(2-4), 441 – 451.

- [17] S.S. Muchnick, Ph.B. Gibbons (2004) Efficient instruction scheduling for a pipelined architecture, ACM SIGPLAN Notices 39(4), 167 – 174.
- [18] M. Pinedo. (2002) Scheduling : theory, algorithms, and systems. 2nd edition, Precentice Hall, Upper Saddle River, New York, USA.
- [19] V. T'kindt, J-C. Billaut. Multicriteria scheduling : theory, models and algorithms. 2nd edition, Springer, 2006.