Trường Đại Học Bách Khoa Tp. Hồ Chí Minh
Khoa Khoa Học & Kỹ Thuật Máy Tính

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
Phan Anh Tú - 1915822
Huỳnh Tuấn Đạt - 1913026
Nguyễn Văn Quốc - 1914864

Phiên bản 1.0

Mục lục

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 20 tháng 11 năm 2022

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.

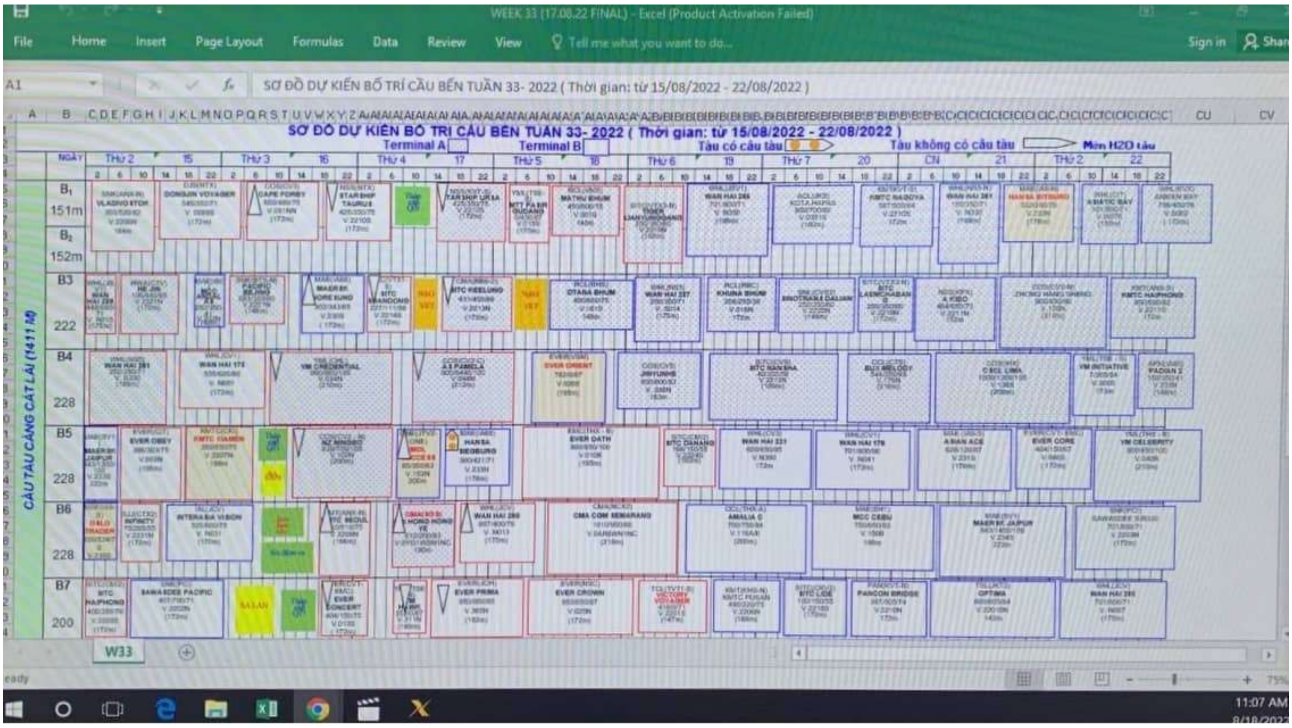
Các bến cảng đóng một vai trò quan trọng trong bối cảnh thương mại thế giới vì chúng là các điểm nối chính chịu trách nhiệm kết nối vận tải đường biển và đường bộ. Có một số vấn đề về tối ưu hóa phát sinh trong các bến cảng và những vấn đề liên quan đến phân bổ bến là một trong những vấn đề quan trọng 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 cẩu 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- BAPC). 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.

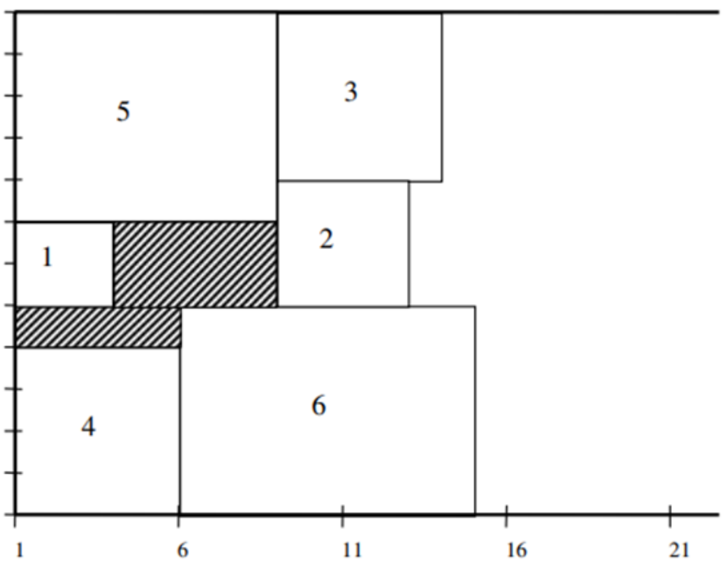
4. Thời gian xử lý đảo chuyển các container tại các tàu cập bến là có thể xác định một cách tự động trước. Hay nói cách khác, thông số đầu vào có thể chứa thông tin về thời gian xử lý (processing time p_i) khi tàu được cập bến.

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 tầu 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).

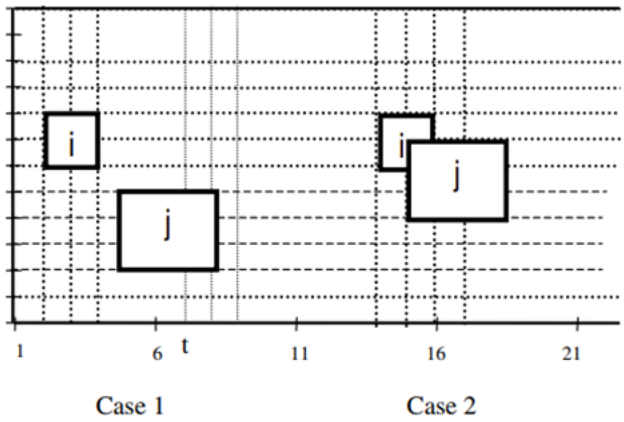


Hình 3: Minh họa về biểu đồ không-thời gian của một lời giải khả thi

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.

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.



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

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, $\forall 1 \leq i \leq N$.
- σ_{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, $\forall 1 \leq i, j \leq N$, $i \neq j$
- δ_{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, $\forall 1 \leq i, j \leq N$, $i \neq j$
- γ_{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 σ_{ij} và δ_{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; σ_{ij}, and δ_{ij}.

Các ràng buộc (10), (11) và (12) trình bày định nghĩa của các biến trung gian γ_{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):

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

- Các ràng buộc (constraints):
  - Mối quan hệ giữa 2 tàu i và j theo các biến ra quyết định σ_{ij} and δ_{ij}:
    Nếu $\sigma_{ij} = 1$ thì tàu i phải được neo đậu vào bến trước tàu j:

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

    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:

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

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

$$\sigma_{ij} + \sigma_{ji} \leq 1, \qquad \forall 1 \leq i,j \leq N, i \neq j \tag{4}$$

$$\delta_{ij} + \delta_{ji} \leq 1, \qquad \forall 1 \leq i,j \leq N, i \neq j \tag{5}$$

$$\sigma_{ij} + \sigma_{ji} + \delta_{ij} + \delta_{ji} \geq 1, \qquad \forall 1 \leq i,j \leq N, i \neq j \tag{6}$$

  - Mối 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:

$$p_i + u_i = c_i, \qquad \forall 1 \leq i,j \leq N, i \neq j \tag{7}$$

  - Miền giá trị của các biến ra quyết định u_i, v_i; σ_{ij}, và δ_{ij}:

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

$$\sigma_{ij}, \delta_{ij} \in \{0,1\}, \qquad \forall 1 \leq i,j \leq N, i \neq j \tag{9}$$

$$b_k.\gamma_{ik} \leq v_i + s_i, \qquad \forall 1 \leq i \leq N, \forall 1 \leq k \leq K \tag{10}$$

  - Các ràng buộc của biến trung gian γ_{ik}:

$$b_k - v_i + S.\gamma_{ik} \geq 0, \qquad \forall 1 \leq i \leq N, \forall 1 \leq k \leq K \tag{11}$$

$$\gamma_{ik} \in \{0,1\}, \qquad \forall 1 \leq i \leq N, \forall 1 \leq k \neq K \tag{12}$$

– 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 γ_{ik} phải thỏa mãn

$$v_i - b_k.\gamma_{ik} \geq 0, \qquad \forall 1 \leq i \leq N, \forall 1 \leq k \leq K \tag{13}$$

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

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

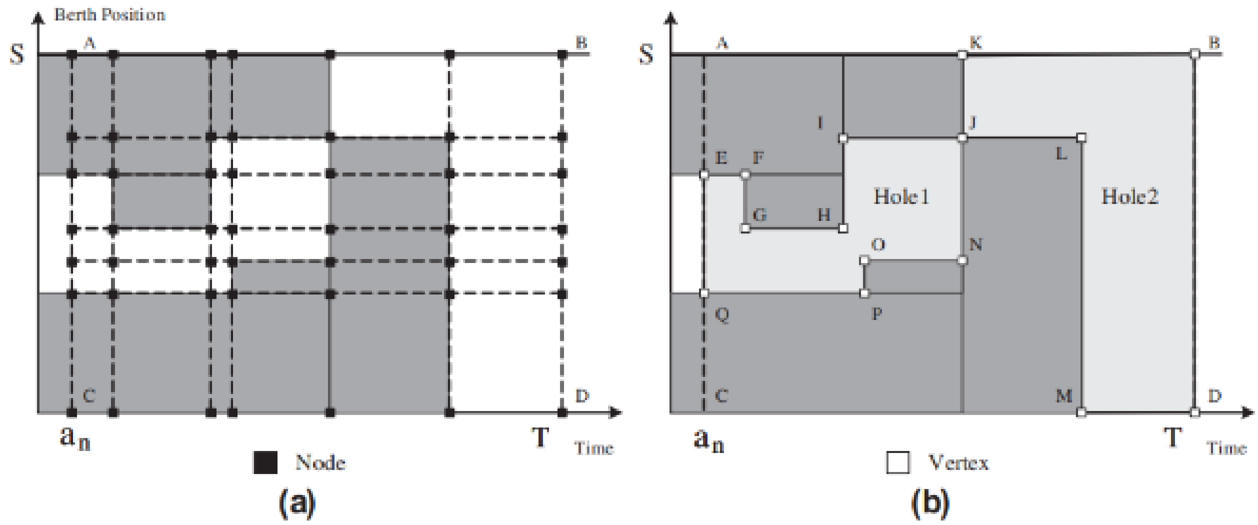
4 Giải pháp đề xuất

Để giải quyết bài toán này, nhóm chúng em đã cải tiến lại **Greedy Randomized Adaptive Search Procedure (GRASP)** tham khỏa trong D.-H.Lee, J.H.Chen, J.X.Cao (2010) sao cho phù hợp với bài toán đặt ra.

4.1 Thuật toán tìm vị trí tàu N

Trước khi vào giải thuật **GRASP**, ta sẽ tìm hiểu về thuật toán tìm tất cả vị trí có thể đặt của tàu thứ N trong biểu đồ không-thời gian khi biết được vị trí của N-1 tàu trước đó. Trước tiên, ta tìm hiểu về các khái niệm sau:

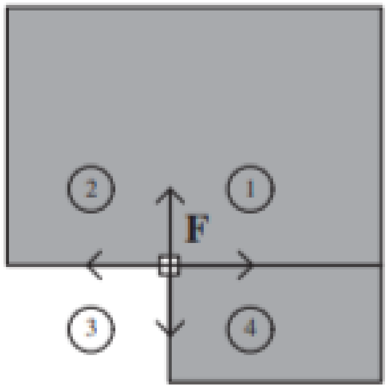
- **Node (Nút)**: Trước tiên, ta xác định một vùng hình chữ nhật trong biểu đồ không-thời gian (tạm gọi là **ABCD**) để thể hiện cho vùng có thể đặt của tàu thứ n dựa vào thời gian đến - a_i và thời gian xử lý - p_i của tàu. Lúc này ta chỉ xét các tàu đã được đóng gói nằm phía trong hoặc một phần nằm phía trong **ABCD**. Mỗi tàu i đã được sắp sẽ được giới hạn bởi 4 đường thẳng (chỉ xét những đường thẳng ở phía trong **ABCD**) $s = v_i; s = v_i + s_i; t = u_i; t = u_i + p_i$. Gộp các đường thẳng này với 4 đường thẳng $t = a_n; t = T; s = 0; s = S$, ta sẽ được tập hợp gồm các đường thẳng. Một **nút** được xác định là giao điểm của 2 đường thẳng bất kỳ trong tập hợp các đường thẳng nói trên. Ở bài toán này, các khoảng break sẽ được xem như là một tàu có size là 0, process time là vô cùng.
- **Hole (Lỗ)**: Là một khu vực khép kín bên trong hình chữ nhật **ABCD** mà đến hiện nay chưa chứa bất kỳ tàu nào cả. Mỗi lỗ sẽ có tập hợp các đỉnh (**Vertex**) và các cạnh (**Edge**) của nó. Ngoài ra đỉnh là các nút đặc biệt ở trong biểu đồ không-thời gian.



Hình 5: Ví dụ về nút, lỗ, cạnh và đỉnh

Tiếp theo ta sẽ đến các bước của giải thuật tìm vị trí có thể đặt của tàu N khi biết vị trí của N-1 tàu trước đó:

1. Đối với mỗi nút được xác định, ta sẽ gán cho nó thành một vector 4 chiều để thể hiện được vùng mà tàu n có để đặt vào. Mỗi nút sẽ sinh ra xung quanh nó 4 góc phần tư được đánh số theo ngược chiều kim đồng hồ như hình vẽ bên dưới.



Hình 6: Ví dụ về vector của một nút

Tiếp theo ta sẽ gán nút đó với mốt vector 4 chiều, mỗi chiều sẽ đại diện cho một gốc phần tư theo thứ tự. Nếu góc phần tư đó có chỗ trống đặt tàu thì sẽ được gán giá trị 1, ngược lại gán cho nó giá trị 0. Ví dụ nút trong hình sẽ được gán cho vector [0, 0, 1, 0]. Ta sẽ chia các nốt thành 5 lớp dựa vào vector mà nó đã được gán. Danh sách 5 lớp được thể hiện bên dưới:

   - **Class 0**: [0, 0, 0, 0];
   - **Class 1**: [1, 0, 0, 0] , [0, 1, 0, 0] , [0, 0, 1, 0] , [0, 0, 0, 1];
   - **Class 2**: [1, 1, 0, 0] , [1, 0, 1, 0] , [1, 0, 0, 1] , [0, 1, 1, 0], [0, 1, 0, 1] , [0, 0, 1, 1];
   - **Class 3**: [1, 1, 1, 0] , [1, 1, 0, 1] , [1, 0, 1, 1] , [0, 1, 1, 1];
   - **Class 4**: [1, 1, 1, 1].

2. Ta chỉ xét đến các nút đặc biệt hay còn gọi là các đỉnh của lỗ. Bằng việc xác định lớp của các đỉnh, ta dễ dàng nhận thấy các đỉnh chỉ thuộc vào một số loại trong 3 lớp dưới đây:

   - **Class 1**: [1, 0, 0, 0] , [0, 1, 0, 0] , [0, 0, 1, 0] , [0, 0, 0, 1];
   - **Class 2**: [1, 0, 1, 0] , [0, 1, 0, 1]
   - **Class 3**: [1, 1, 1, 0] , [1, 1, 0, 1] , [1, 0, 1, 1] , [0, 1, 1, 1];

   Tuy nhiên đối với các đỉnh thuộc class 2, ta có thể tách chúng thành 2 đỉnh của class 1. Ví dụ, với [1, 0, 1, 0] thì nó sẽ được tách thành [1, 0, 0, 0] và [0, 0, 1, 0]. Do vậy ta có thể xem các đỉnh được xét đều thuộc vào hai **Class 1** và **Class 3**

3. Sau khi xác định được vector của các đỉnh và phân chia chúng thành 2 lớp, ta sẽ tiến hành bước tiếp theo. Đối với các đỉnh thuộc **Class 1**, ta chỉ quan tâm đến các đỉnh có vector [1, 0, 0, 0] , [0, 0, 0, 1] và gộp chung lại một tập gọi là **C1**. Đối với các đỉnh của **Class 3**, ta sẽ tiến hành theo các bước dưới đây:

   - Với các đỉnh có vector [1, 1, 1, 0], [0, 1, 1, 1], từ đỉnh đó ta sẽ kẻ một đường thẳng nằm ngang kéo dài về phía bên trái.
   - Với các đỉnh có vector [1, 0, 1, 1], ta sẽ kẻ một đường thẳng đi xuống phía bên dưới, với các đỉnh có vector [1, 1, 0, 1] ta sẽ kẻ một đường thẳng đi lên phía trên.

- Tiếp theo ta sẽ xác định các nút giao và sắp chúng vào hai tập **I1** và **I2**. Tập **I1** là tập các nút giao của đường kéo dài với cạnh của lỗ. Trong khi đó tập **I2** sẽ là tập các nút giao của các đường kéo dài. Sau đó ta sẽ tiến hành cập nhật lại vector của các nút giao dựa vào bảng dưới đây. Trong đó *a, a1, a2* sẽ là các vector điểm đầu của đường kéo dài tương ứng sinh ra điểm giao đó.

For nodes in I_1			For nodes in I_2		
a		$\mathbf{b}_{I_1}$	$\mathbf{a}_1, \mathbf{a}_2$		$\mathbf{b}_{I_2}$
[0,1,1,1]	⇒	[0,0,0,1]			
[1,0,1,1]	⇒	[1,0,0,0]	[1,0,1,1] , [1,1,1,0]	⇒	[1,0,0,0]
[1,1,0,1]	⇒	[0,0,0,1]	[1,1,0,1] , [0,1,1,1]	⇒	[0,0,0,1]
[1,1,1,0]	⇒	[1,0,0,0]			

Hình 7: Bảng cập nhật vector điểm giao

4. Như vậy lúc này, ta đã có được các điểm khả thi có thể đặt tàu n vào. Đó là tập hợp của **C1**; **I1** và **I2** (sau khi đã cập nhật). Tuy nhiên ta phải tiến hành kiểm tra thử độ khả thi của các nút này và tiến hành cập nhật lại tập hợp các điểm khả thi một lần nữa để tiền hành được bước tiếp theo.

Dưới đây là các bước tóm tắt lại thuật toán tìm tập hợp điểm khả thi để đặt tàu N khi N-1 tàu trước đó đã được sắp xếp:

- **Bước 1**: Nhập thông tin vị trí đã được xếp của N-1 tàu trước đó và thông tin của tàu thứ n (a_n, p_n, s_n).
- **Bước 2**: Tiến hành xác định các đỉnh của các lỗ được tạo ra trong biểu đồ không-thời gian. Sau đó phân loại các lớp cho các đỉnh đó. Lưu ý đối với **Class 2** thì sẽ phân tách thành hai đỉnh thuộc **Class 1**. Khi đó các đỉnh chỉ thuộc **Class 1** và **Class 3**.
- **Bước 3**: Với các đỉnh thuộc lớp 1 có vector [1, 0, 0, 0] , [0, 0, 0, 1], ta gộp chung lại thành tập **C1**. Các đỉnh thuộc lớp 3 gộp chung lại thành tập **C3**.
- **Bước 4**: Với các đỉnh thuộc lớp 3 tiến hành xác định điểm giao và cập nhật điểm giao sau đó gộp thành hai tập **I1** và **I2** như đã trình bày ở trên.
- **Bước 5**: Tiến hành kiểm tra tính khả thi và tối ưu của việc đặt tàu n vào các nút trong tập hợp 3 tập **C1**, **I1** và **I2** bằng cách đặt tàu N vào gốc phần tư được cho phép của mỗi nút. Cuối cùng kết luận lại tập hợp điểm khả thi cho việc đặt tàu N.

4.2 Thuật toán GRASP

Thuật toán tìm vị trí của tàu N ở phía trên sẽ được áp dụng vào trong giải thuật GRASP sẽ được trình bày dưới đây. Giải thuật GRASP sẽ gồm hai giai đoạn trong mỗi lần lặp lại của chúng: Construction và Local search. Số lần lặp sẽ được quy định trước thông qua biến *Max_Iterations*. Dưới đây là pseudo code của GRASP

1. **Giai đoạn Construction:**

   Ở giai đoạn này, các tàu sẽ được chèn lặp đi lặp lại cho đến khi ta được một thứ tự hoàn chỉnh của các tàu. Thuật toán để tiến hành chèn được sử dụng là một giải thuật tham lam dựa vào thuật toán tìm vị trí tàu N đã được nếu ở phần trước.

   Dưới đây là các bước của giai đoạn Construction:

```
Procedure GRASP(Max_Iterations, Seed)
    Read_Input();
    for k = 1, . . . , Max_Iterations do
        Solution ← Greedy_Randomized_Construction(Seed);
        Solution ← Local_Search(Solution);
        Update_Solution(Solution,Best_Solution);
    end;
    return Best_Solution;
endGRASP.
```

Hình 8: Pseudo code của thuận giải GRASP

- **Bước 1:** Nhập thông tin đầu vào cho tất cả các tàu (Thời gian đến, thời gian xử lý và chiều dài tàu.) Sau đó sắp xếp theo chiều tăng dần thời gian đến của chúng, ta được một dãy các tàu đặt tên là *Ves_ Seq*.
- **Bước 2:** Lần lượt thêm các tàu theo danh sách *Ves_ Seq*. Ở tàu thứ N, ta sẽ có thông tin vị trí của N-1 tàu trước đó. Sau đó dùng giải thuật tìm kiếm vị trí của tàu N đã được giới thiệu ở trên để tìm ra k vị trí thích hợp đặt tàu thứ N. Gọi tập hợp các vị trí trên là tập hợp K. Với mỗi vị tri k, sau khi đặt tàu thứ N vào thực hiện tính toán chi phí theo công thức (1) và ký hiệu là $Cost_k$. Tiếp theo ta sẽ dùng công thức bên dưới để loại bỏ đi một số giá trị k để cuối cùng ta thu lại được tập các vị trí có thể đặt tàu N ký hiệu là K'

$$\frac{1}{Cost_k} < r * \max_{i=0}^{K}\{\frac{1}{Cost_i}\} \tag{15}$$

  Trong đó $0 \le r \le 1$ và r là hằng số
- **Bước 3:** Với mỗi vị trí k' trong K', tính toán xác suất theo công thức dưới đây:

$$Pr_{k'} = \frac{\frac{1}{Cost_{k'}}}{\sum_{i=1}^{K'} \frac{1}{Cost_i}} \tag{16}$$

- **Bước 4:** Thực hiện chọn ngẫu nhiên một vị trí k' theo phân phối xác xuất đã tính ở công thức (16). Ở bước này, ta thường sử dụng phương pháp **roulette wheel selection** để thực hiện chọn.
- **Bước 5:** Thực hiện lặp lại bước 2-4 đến khi tất cả các tàu đên đều được sắp xếp

Sau khi thực hiện xong Giai đoạn Construction, ta sẽ được thông tin về vị trí của tất cả các tàu đã được sắp vào cảng để tiếp tục thực hiện giai đoạn tiếp theo.

2. **Giai đoạn Local search:**

Ở giai đoạn này ta cũng sẽ thực hiện tuần tự lặp lại 2 quá trình. Quá trình 1 là hoán đổi vị trí của hai tàu bất kì liền kề nhau. Giả sử danh sách lúc đầu là $i_1, i_2, ..., i_m, i_{m+1}, ..., i_N$. Sau khi hoán đổi ta sẽ có một danh sách thứ tự chèn tàu mới là $i_1, i_2, ..., i_{m+1}, i_m, ..., i_N$.

Tiếp đến ra sẽ qua quá trình hai là thực hiện một giải thuật A-star để đánh giá tính hiệu quả của danh sách mới đó. Ta sẽ chia danh sách đó ra thành 2 danh sách con tại điểm break m: $i_1, i_2, ..., i_m$ và $i_{m+1}, ..., i_N$. Ta sẽ giữ nguyên vị trí của m tàu đầu tiên và thực hiện A-star cho N-m tàu ở danh sách sau.

Đối với mỗi tàu j ($m + 1 \le j \le N$) được xếp vào, ta sẽ có tập K bao gồm các vị trí có thể đặt của tàu j. Với mỗi vị trí k trong tập K, ta sẽ tiến hành tính giá trị theo hàm h(j,k). Vị trí k sao cho nó giá trị của hàm h(j,k) sẽ lớn nhất trong tập K thì vị trí k đó sẽ được chọn là vị trí đặt tàu thứ j. Lặp lại quá trình đến khi tàu N được xếp.

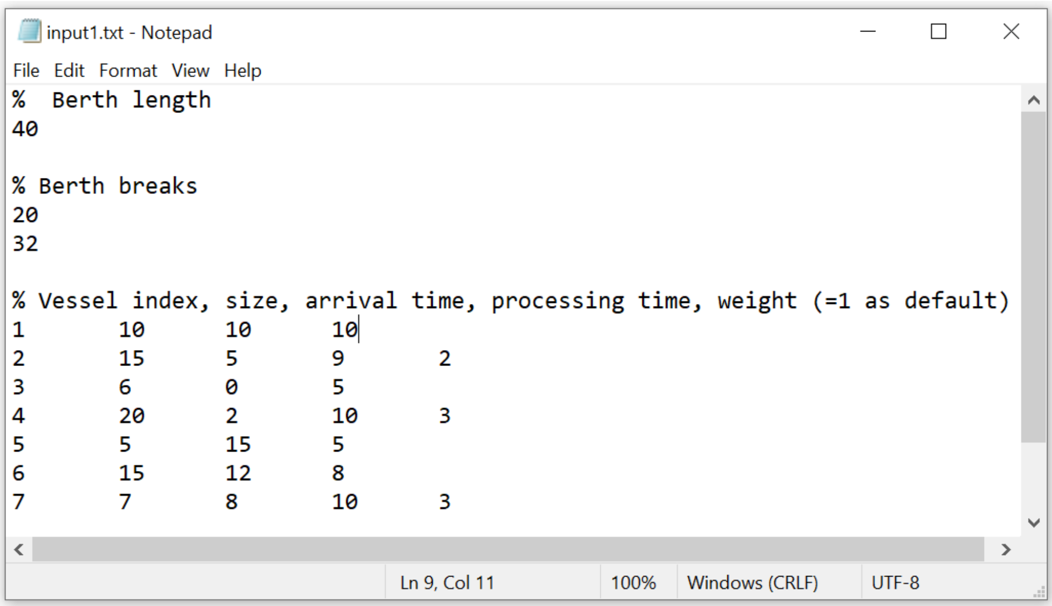
Các bước của giai đoạn Local search sẽ được diễn đạt lại dưới đây:

- **Bước 1**: Hoán đổi lại vị trí của hai tàu và tính toán chi phí của danh sách đã được hoán đổi. Sau đó đánh giá xem danh sách vừa hoán đổi đó có hiệu quả không.
- **Bước 2**: Lặp lại bước 1 L1 lần sau đó chọn ra trường hợp hiệu quả nhất.

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.

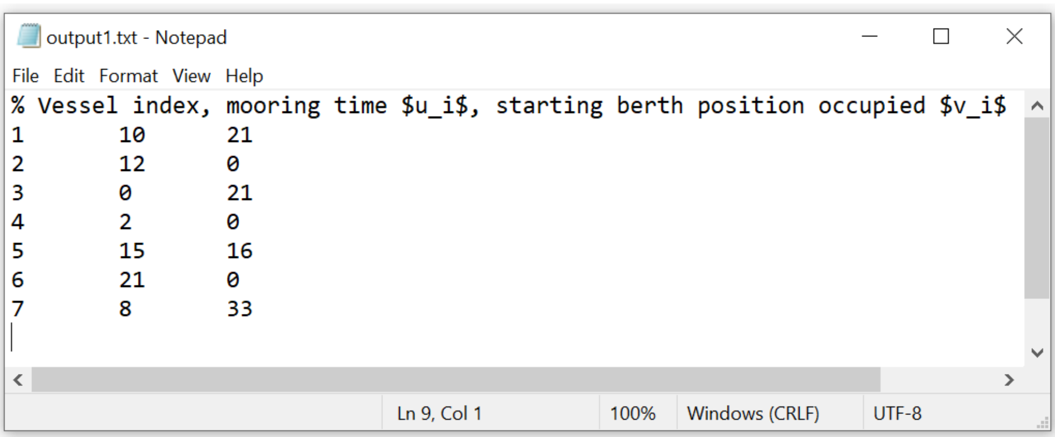



```
%  Berth length
40

% 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
```

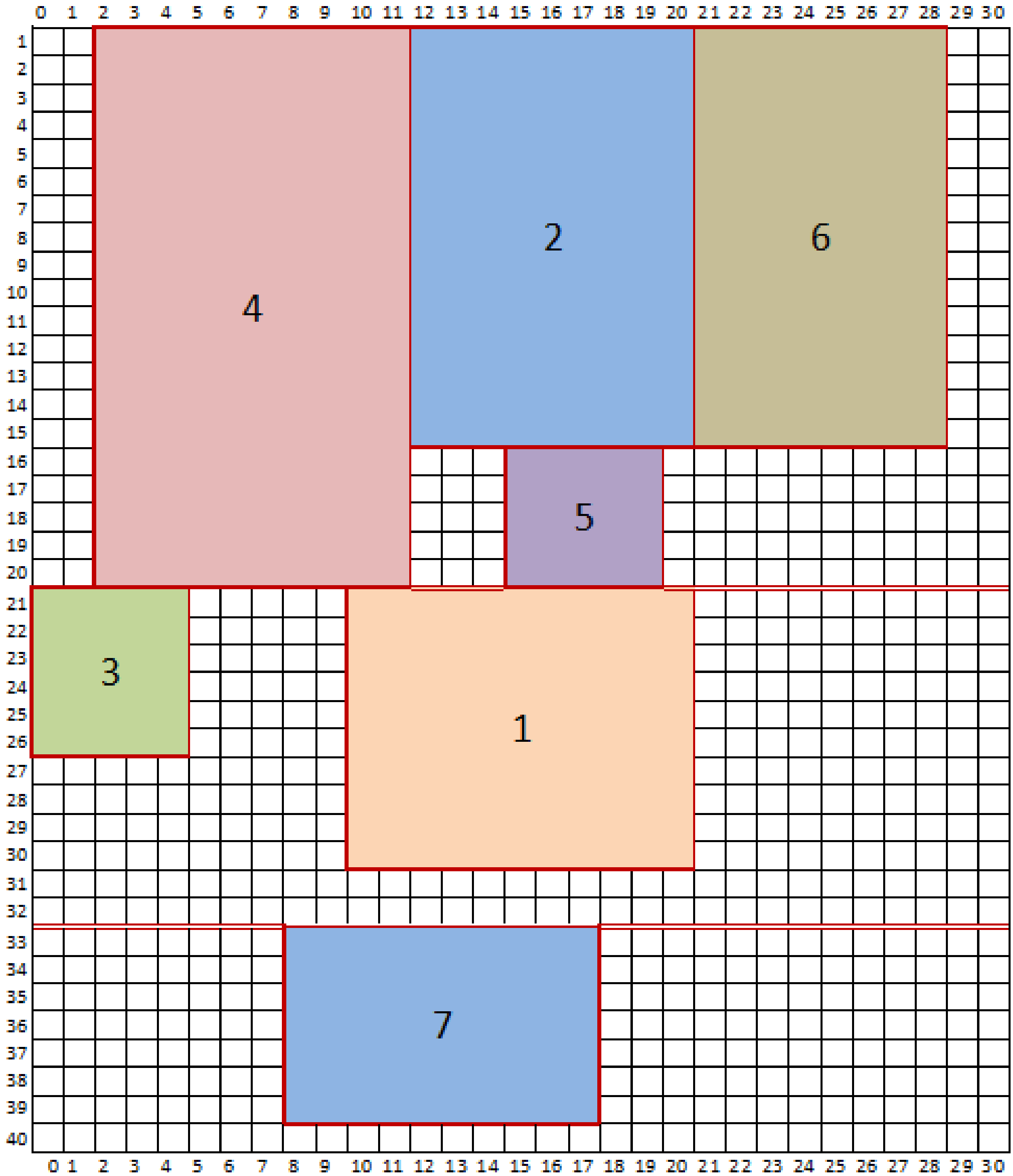
Hình 9: Mẫu dữ liệu đầu vào của bài toán nghiên cứu




```
% Vessel index, mooring time $u_i$, starting berth position occupied $v_i$
1       10      21
2       12      0
3       0       21
4       2       0
5       15      16
6       21      0
7       8       33
```

Hình 10: Mẫu dữ liệu xuất kết quả

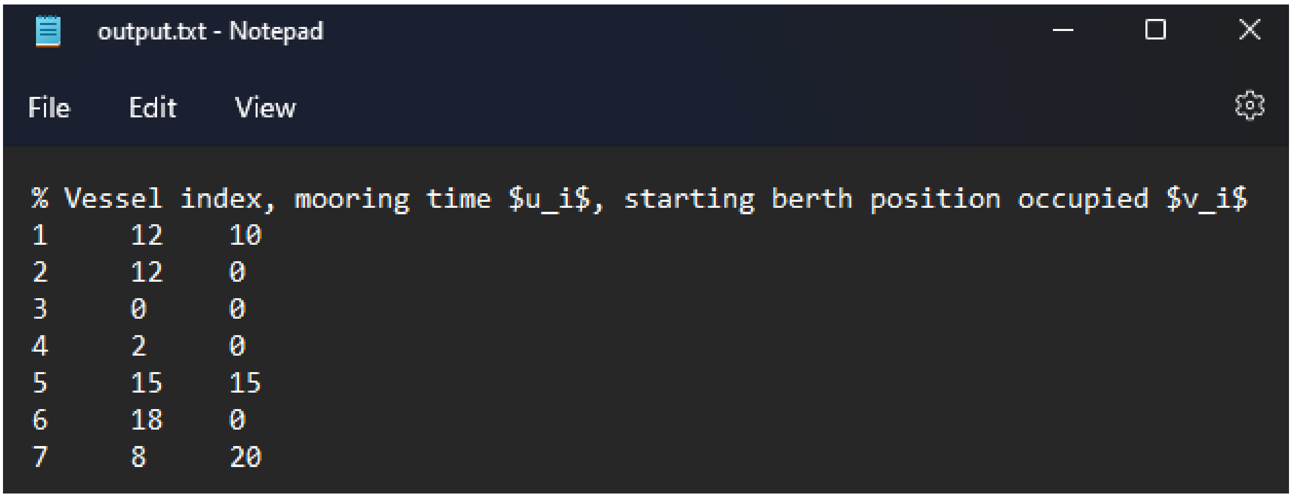
Một lời giải khả thi có thể được xác định như Hình 5.2 và kết quả cần xuất tương ứng sẽ có định dạng như Hình 5.1. Kết quả thu được từ hàm mục tiêu là 23.



Hình 11: Một lời giải khả thi trên biểu đồ không-thời gian

5.2 Kết quả thực nghiệm và đánh giá

Sau khi thực hiện chạy thuật toán cho dữ liệu đầu vào như trên ta được kết quả dưới đây.



```
output.txt - Notepad
File Edit View

% Vessel index, mooring time $u_i$, starting berth position occupied $v_i$
1	12	10
2	12	0
3	0	0
4	2	0
5	15	15
6	18	0
7	8	20
```

Hình 12: Kết quả

Đánh giá kết quả: Với đầu vào dữ liệu gồm 5 tàu thì thời gian chạy trung bình chạy là 17s $\pm$ 4s

6 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 ban 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.