

# Sử dụng mạng Petri và thuật toán tìm kiếm heuristic để lập tiến trình sản xuất cho hệ thống sản xuất linh hoạt

## Using Petri Nets and Heuristic Search Algorithms for Flexible Manufacturing System Scheduling

Phạm Trường Tùng<sup>1</sup>, Phạm Đăng Phước<sup>1</sup>, Lưu Đức Bình<sup>2</sup>

<sup>1</sup>ĐH Phạm Văn Đồng, <sup>2</sup>ĐH Bách khoa - ĐH Đà Nẵng

Email: pttung@pdu.edu.vn, pphamdang@yahoo.com, ldbinh@dut.edu.vn

### Abstract

Petri nets have been extensively used to model the manufacturing system due to their power to describe complex systems with their simple tools. However, the “state explosion” has prevented their use for solving real scheduling problems. This paper proposes an algorithm that combines the BFS with heuristic functions and the limited beam expansion to search on reachability graph for FMS scheduling.

### Keywords

Petri nets, manufacturing system, scheduling, heuristic search, BFS search, reachability graph

### Tóm tắt<sup>1</sup>

Mạng Petri là một công cụ hữu dụng cho việc mô hình hóa các hệ thống sản xuất với khả năng mô tả các hệ thống phức tạp bằng các công cụ đơn giản. Tuy nhiên, vấn đề bùng nổ trạng thái đã gây hạn chế cho việc giải bài toán về lập kế hoạch sản xuất. Bài báo đề xuất sử dụng thuật toán BFS kết hợp với hàm đánh giá heuristic và phương pháp giới hạn mức độ triển khai tìm kiếm trên cây trạng thái để lập kế hoạch sản xuất cho hệ thống sản xuất linh hoạt.

### Chữ viết tắt

PN	PetriNet – Mạng Petri
DES	Discrete event system - hệ sự kiện rời rạc
FMS	Flexible manufacturing system - hệ thống sản xuất linh hoạt
BFS	Best First Search
bw	Beam width

### 1. Phần mở đầu

Lý thuyết mạng Petri được đề xuất lần đầu tiên vào năm 1962 bởi Carl Adam Petri trong luận án tiến sĩ “Giao tiếp với các thiết bị tự động” tại Leipzig. Mạng Petri là một công cụ đồ hình và toán học được sử dụng rất hữu hiệu trong việc mô tả, phân tích các sự kiện rời rạc, đặc biệt là đối với hệ thống sản xuất tự động bởi nó tương thích với các tính chất tự nhiên của hệ thống sản xuất như tính không đồng bộ (asynchronous), tính đồng thời (concurrent), không xác định (nondeterministic).

Đến nay, đã có nhiều công trình sử dụng mạng Petri để mô tả, phân tích hệ thống sản xuất tự động như Chen YuFeng và Li ZhiWu cùng các cộng sự trong [1] đã sử dụng mạng Petri để mô hình hóa hệ

thống FMS và nghiên cứu phục hồi hoạt động của hệ thống từ trạng thái deadlock; Tüysüz Fatih và Kahraman Cengiz trong [15] sử dụng mạng Petri xác suất để mô hình hóa một cell trong hệ thống FMS với các tham số mờ; Lee Jongkun và Korbaa Ouajdi trong [6] đã sử dụng mạng Petri để mô hình hóa và lập kế hoạch hoạt động cho một hệ thống FMS; Man-zhi Liu và Mei-hua Zhou cùng các cộng sự trong [8] sử dụng mạng Petri hướng đối tượng (object-oriented Petri net) để mô hình hóa và nghiên cứu một dây chuyền cung cấp than đá trong sản xuất tại công ty YanKuang – Trung Quốc; Savsar Mehmet và Aldaihani Majid trong [14] sử dụng mạng Petri để mô hình hóa và phân tích sự cố trong dây chuyền FMS; Phạm Trường Tùng và Phạm Đăng Phước trong [12] sử dụng mạng Petri để mô hình hóa và phân tích hoạt động của một hệ thống FMS có 2 máy CNC và một robot; Dotoli Mariagrazia và Epicoco Nicola cùng các cộng sự trong [2] sử dụng mạng Petri thời gian để mô hình hóa các sự kiện rời rạc của hệ thống vận tải đa phương thức (Intermodal Freight Transport Terminals).

Việc sử dụng mạng Petri để mô hình hóa các sự kiện gián đoạn (DES) được xem là một công cụ hữu hiệu, và đơn giản. Tuy nhiên, một nhược điểm rất lớn của mạng Petri khi sử dụng để phân tích trên mô hình trên cây trạng thái (reachability graph) là sự “bùng nổ trạng thái” (state explosion). Với một mô hình mạng Petri, có thể nói việc xây dựng cây trạng thái là việc dễ dàng, tuy nhiên ta sẽ gặp vấn đề là cây trạng thái có thể rất lớn cho một mô hình mạng Petri nhỏ.

Việc xây dựng và tìm kiếm một trạng thái của mạng Petri bằng cách duyệt hết trên cây trạng thái là một bài toán không hề đơn giản, tốn rất nhiều tài nguyên máy và thời gian duyệt. Do đó để tìm được chuỗi thông tốt nhất đến trạng thái yêu cầu, đã có rất nhiều các công trình nghiên cứu áp dụng các thuật toán tìm kiếm trên cây trạng thái hoặc các giải thuật mô phỏng khác.

<sup>1</sup> Ngày nhận bài: 15/12/2017; Ngày nhận bản sửa: 02/05/2018; Ngày chấp nhận: 12/05/2018; Phân biên: Đào Văn Hiệp, Đỗ Trung Hải

Mejía Gonzalo và Niño Karen trong [9] đề xuất một thuật toán HFBS (Hybrid Filtered Beam Search) trên mạng Petri thời gian bằng cách đa dạng/tăng cường chiến lược tìm kiếm trên cây trạng thái để lập kế hoạch giải phóng hệ thống FMS khỏi trạng thái chết (deadlock) trong quá trình hoạt động; Li Xiaoling và Xing Keyi cùng các cộng sự trong [7] đã đề xuất thuật toán lập kế hoạch hoạt động cho hệ thống FMS trên cơ sở mạng Petri và giải thuật di truyền trên bằng cách xem xét chuỗi thông và tiến trình thông của mạng Petri – tuy nhiên phương pháp này mô tả các toán tử di truyền bằng công cụ toán học là tương đối phức tạp, thời gian mô phỏng để đạt kết quả còn chậm; Huang Bo và Jiang Rongxi cùng các cộng sự trong [4] sử dụng các hàm admissible heuristic và nonadmissible heuristic cho thuật toán A\* để xác định chiến lược tìm kiếm trên cơ sở mạng Petri P-thời gian của hệ thống FMS; Kim Young Woo và Suzuki Tatsuya cùng các cộng sự trong [5] sử dụng mạng Petri thời gian màu (colored timed Petri net) để mô tả hệ thống FMS, phương pháp này giúp việc xác định chuỗi thông dễ dàng hơn, tuy nhiên việc mô tả phức tạp hơn; Reyes A. và Yu H. cùng các cộng sự trong [13] sử dụng phương pháp tìm kiếm trên cây trạng thái của mạng Petri tích hợp và giải thuật tìm kiếm heuristic lai để xác định chuỗi thông của mạng Petri, phương pháp này có ưu điểm là tốc độ tìm kiếm chuỗi thông nhanh, tuy nhiên việc mô tả bằng cách kết hợp các luật heuristic làm cho giải thuật trở nên phức tạp.

Bài báo này đề xuất một mô hình mạng Petri cho hệ thống sản xuất tự động, đề xuất một phương pháp kết hợp thuật toán tìm kiếm BFS (Best First Search) và hàm đánh giá heuristic để xây dựng cây trạng thái có giới hạn và tìm kiếm chiến lược hoạt động cho hệ thống sản xuất tự động.

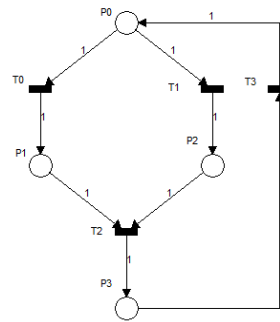
## 2. Xác định chiến lược hoạt động cho hệ thống sản xuất tự động

### 2.1 Mô hình mạng Petri và mô tả hệ thống sản xuất tự động bằng mạng Petri

#### a) Các định nghĩa mạng Petri [3]

**Định nghĩa 1:** Mạng Petri là tập hợp gồm 4 thành phần, ký hiệu  $N = \langle P, T, Pre, Post \rangle$ . Trong đó:

- $P = (p_1, p_2, \dots, p_n)$  là một tập hợp không rỗng chứa hữu hạn các vị trí (place).
- $T = (t_1, t_2, \dots, t_m)$  là một tập hợp không rỗng chứa hữu hạn các chuyển tiếp (transition).
- $Pre: P \times T \rightarrow N$  là hàm trọng lượng các cung từ vị trí đến chuyển tiếp.
- $Post: T \times P \rightarrow N$  là hàm trọng lượng các cung từ chuyển tiếp đến vị trí.
- $Pre, Post$  là các ma trận có số hàng bằng với số phần tử của tập  $P$  và số cột bằng số phần tử của tập  $T$ . Nếu  $pre(p_i, t_j) = W(p_i, t_j) \neq 0$  thì sẽ tồn tại một cung có trọng lượng  $W(p_i, t_j)$  nối từ vị trí  $p_i$  đến chuyển tiếp  $t_j$ ; Nếu  $Post(t_i, p_j) = W(t_i, p_j) \neq 0$  thì sẽ tồn tại một cung có trọng lượng  $W(t_i, p_j)$  nối từ chuyển tiếp  $t_i$  đến vị trí  $p_j$ .



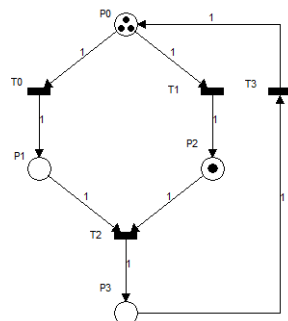
H.1 Một mô hình mạng Petri

Trong H. 1 biểu diễn một mạng Petri có  $P = (p_0, p_1, p_2, p_3)$ ;  $T = (t_0, t_1, t_2, t_3)$  và các ma trận

$$Pre = \begin{matrix} & t_0 & t_1 & t_2 & t_3 \\ p_0 & 1 & 1 & 0 & 0 \\ p_1 & 0 & 0 & 1 & 0 \\ p_2 & 0 & 0 & 1 & 0 \\ p_3 & 0 & 0 & 0 & 1 \end{matrix} \quad Post = \begin{matrix} & t_0 & t_1 & t_2 & t_3 \\ p_0 & 0 & 0 & 0 & 1 \\ p_1 & 1 & 0 & 0 & 0 \\ p_2 & 0 & 1 & 0 & 0 \\ p_3 & 0 & 0 & 1 & 0 \end{matrix}$$

Một mạng Petri được mô tả như trên là mạng Petri tĩnh. Để mô tả hoạt động của mạng Petri, người ta sử dụng mạng Petri đánh dấu.

**Định nghĩa 2:** Một đánh dấu  $M$  của Mạng Petri  $N$  là một phép chiếu từ  $P$  vào  $N$ , được thể hiện bằng một số nguyên không âm (gán số token) vào mỗi vị trí mạng. Mạng Petri được đánh dấu là một cặp  $\langle N, M_0 \rangle$  với  $N$  là mạng Petri và  $M_0$  là một đánh dấu ban đầu của  $N$ .



H.2 Mạng Petri đánh dấu

Trong H. 2 mô tả mạng Petri đánh dấu trên cơ sở mạng Petri ở hình 1. Đánh dấu  $M_0$  của mạng Petri như sau:

$$M_0^T = [3 \quad 0 \quad 1 \quad 0]$$

**Định nghĩa 3:** Một đánh dấu của mạng sẽ được phát triển dựa trên quy tắc thông như sau:

- Một chuyển tiếp  $t$  được gọi là thông (tích cực) nếu như tất cả các vị trí đầu vào của nó có số token lớn hơn hoặc bằng trọng lượng cung nối giữa vị trí và chuyển tiếp tương ứng.

$$M \geq Pre(t) \tag{1}$$

- Khi một chuyển tiếp thông, nó sẽ lấy đi (thêm vào) vị trí đầu vào (đầu ra) số lượng token bằng với trọng lượng cung nối giữa vị trí (chuyển tiếp) và chuyển tiếp (vị trí) tương ứng. Ký hiệu  $(M_2[t$

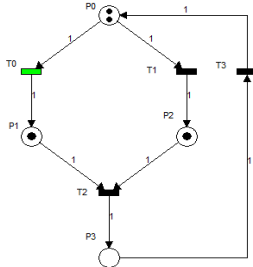
$>M_1$ ) là hệ thống chuyển tiếp từ đánh dấu  $M_1$  sang đánh dấu  $M_2$  khi chuyển tiếp  $t$  thông. Khi đó:

$$M_2 = M_1 + \text{Post}(t) - \text{Pre}(t) = M_1 + C(t) \quad (2)$$

Với  $C(t)$  được gọi là ma trận tiến.

- Gọi  $M_k$  là trạng thái đạt được từ trạng thái  $M_0$  sau chuỗi các chuyển tiếp  $\sigma = t_1 t_2 \dots$ . Khi đó:

$$M_k = M_0 + C \cdot \sigma \quad \sigma \geq 0 \quad (3)$$



H.3 Trạng thái  $M_1$  sau khi chuyển tiếp  $t_0$  thông

Mạng Petri được mô tả như các định nghĩa trên là mạng Petri cơ bản. Để mô hình hóa các hệ thống thời gian, ta định nghĩa một mạng Petri thời gian như sau:

**Định nghĩa 4:** Một mạng Petri thời gian chuyển tiếp (transition timed Petri net (t-TPN)) là một cặp  $\langle N, Z \rangle$  với  $N = \langle P, T, \text{Pre}, \text{Post} \rangle$  và  $Z$  là một hàm số thực không âm với  $z_i$  tương ứng với mỗi chuyển tiếp trong mạng sao cho  $Z: T \rightarrow R^+$ ;  $z_i = Z(t_i)$  được gọi là thời gian từ khi chuyển tiếp  $t_i$  tích cực cho đến khi thông. Luật chuyển tiếp của t-TPN được định nghĩa giống như với mạng Petri thông thường.

Một mạng Petri thời gian vị trí (place timed Petri net (p-TPN)) là một cặp  $\langle N, R \rangle$  với  $N = \langle P, T, \text{Pre}, \text{Post} \rangle$  và  $R$  là một hàm số thực không âm với  $r_i$  tương ứng với mỗi vị trí trong mạng sao cho  $R: P \rightarrow R^+$ ;  $r_i = Z(t_i)$  được gọi là thời gian tồn tại của một token tại vị trí  $p_i$ .

**b) Mô tả hệ thống sản xuất tự động bằng mạng Petri**

Hệ thống sản xuất tự động có 3 máy CNC gồm một máy phay CNC, một máy tiện CNC và một máy khoan CNC. Hệ thống sử dụng 1 robot để phục vụ cấp phôi cho máy và lấy sản phẩm sau khi gia công xong.

Hệ thống sản xuất đồng thời 2 loại chi tiết với các trình tự nguyên công khác nhau:

- Chi tiết 1 có thể thực hiện các nguyên công theo thứ tự gồm: Phay - Tiện - Khoan hoặc có thể theo thứ tự Phay - Khoan - Tiện với thời gian khác nhau.
- Chi tiết 2 có thể thực hiện các nguyên công theo thứ tự: Tiện - Phay - Khoan; Phay - Tiện - Khoan hoặc Phay - Khoan - Tiện.

**B.1 Thời gian thực hiện các nguyên công**

Chi tiết	Phay	Khoan	Tiện
1	10	4	6
2	7	6	8

Giả định rằng thời gian phục vụ của robot được tính vào thời gian thực hiện nguyên công ở B. 1. Ta mô tả hệ thống bằng mạng Petri thời gian chuyển tiếp ở H. 4. Các vị trí của mạng Petri mô tả hệ thống được trình bày ở B. 2, B. 3, B. 4.

**B.2 Các vị trí của PN mô tả các nguyên công trên các máy**

Chi tiết	Phay	Khoan	Tiện
1	$p_1, p_4$	$p_2, p_6$	$p_3, p_5$
2	$p_{32}, p_{34}, p_{42}$	$p_{33}, p_{36}, p_{43}$	$p_{31}, p_{35}, p_{44}$

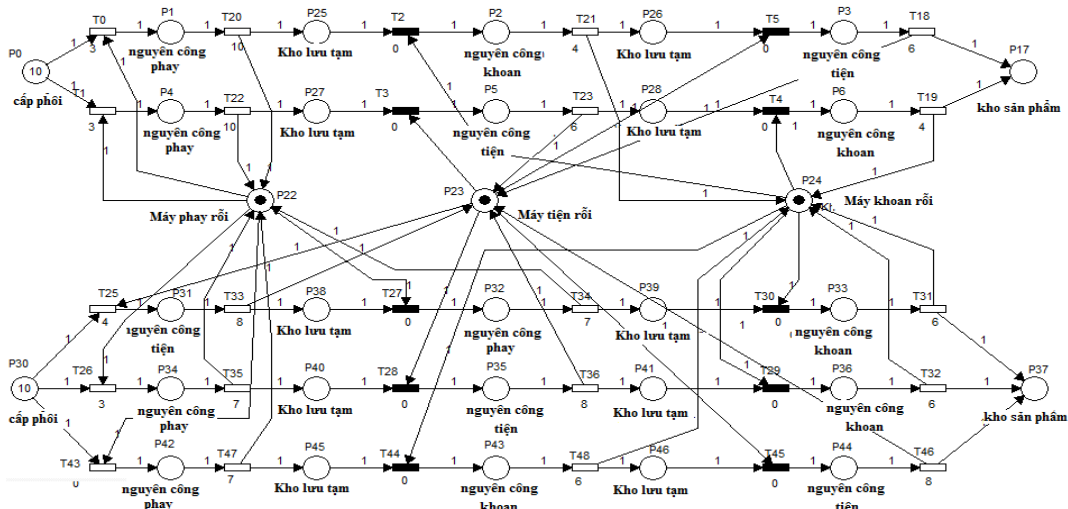
**B.3 Các vị trí mô tả trạng thái rỗi của các máy**

Phay	Khoan	Tiện
$p_{22}$	$p_{24}$	$p_{23}$

**B.4 Các vị trí mô tả kho cấp phôi**

Chi tiết	Cấp phôi	Kho lưu tạm máy Phay	Kho lưu tạm máy Khoan	Kho lưu tạm máy Tiện	Hoàn thành
1	$p_0$	$p_{25}, p_{27}$	$p_{26}$	$p_{28}$	$p_{17}$
2	$p_{30}$	$p_{39}, p_{40}, p_{45}$	$p_{46}$	$p_{38}, p_{41}$	$p_{37}$

Các chuyển tiếp sau các vị trí thể hiện nguyên công mô tả thời gian thực hiện các nguyên công đó.



H.4 Mạng Petri cho hệ thống sản xuất tự động



Nếu tất cả các phần tử của vector  $\gamma$  không âm thì từ đánh dấu  $M_k$  có thể đến được đánh dấu  $M_{end}$ . (Dấu “\” là phép chia ngược trong Matlab).

- Tính bao tại một vị trí trong mạng (gọi là  $h_3$ ). Tức là nếu sau khi thông mà số token tại một vị trí nào đó lớn hơn giá trị cho trước thì  $h_3 = 0$ .
- Tính lặp ( $h_4$ ) tức là xuất phát từ đánh dấu  $M_k$  nhưng nếu đánh dấu  $M_{k+1} = M_{i(i=1..k)}$  thì  $h_4 = 0$ . Điều này sử dụng để tránh cho giải thuật rơi vào vòng lặp vô hạn.
- Ước lượng trạng thái nào gần đích hơn (gọi là  $h_5$ ).  

$$h_5 = \text{Sum}(C \setminus (M_{end} - M_k)) \quad (9)$$
 Với hàm Sum là hàm tính tổng tất cả các phần tử của một vector.
- Tổng thời gian thông từ  $M_0$  đến  $M_k$  (gọi là  $h_6$ ).  
 Như vậy hàm heuristic được xác định như sau:

$$h = h_2 h_3 h_4 \left( \alpha h_1 + \frac{\beta}{h_5} + \lambda h_6 \right) \quad (10)$$

Với  $\alpha, \beta, \lambda$  là các hệ số. Như vậy, mô tả một trạng thái tại “thời điểm”  $k$  bao gồm các yếu tố sau:

- Đánh dấu  $M_k$  ( $M_k$  là một vector là phần tử của ma trận  $M_k^j$ ).
  - Vector thời gian chuyển tiếp  $T_{remaining(k)}$ .
  - Tổng thời gian chuyển tiếp từ  $M_0$  đến  $M_k$ .
  - Hàm đánh giá heuristic.
- Khi xét phép thông để chuyển từ đánh dấu  $M_k$  sang đánh dấu  $M_{k+1}$  ta sẽ được kết quả sau:
- Ma trận  $M_{k+1}^k$ .
  - Ma trận thời gian đếm ngược  $T_{remaining}^k$ .
  - Vector thời gian chuyển tiếp  $T_{sum\_time}^k$ .
  - Vector hàm đánh giá heuristic  $h_{k+1}^k$ .

*b) Bài toán xây dựng trạng thái của mạng Petri ở bước k+1 từ một trạng thái k*

Giả sử ta cần xây dựng tất cả các trạng thái có thể có ở bước k+1 xuất phát từ một trạng thái k của mạng Petri.

Tại trạng thái ở bước thứ k, ta có các mô tả sau:

- Vector đánh dấu  $M_i$ .
- Vector thời gian chuyển tiếp  $T_{remaining(i)}$ .
- Vector chuỗi thông  $RM_i$
- Tổng thời gian chuyển tiếp từ  $M_0$  đến  $M_i$  là  $T_{sumtime(i)}$ .
- Hàm đánh giá heuristic  $h_i$ .

Tại trạng thái ở bước thứ k+1:

- Tính ma trận thông  $\sigma$  theo (5).
- Xác định ma trận đánh dấu theo (6)
- Xác định ma trận thời gian chuyển tiếp:
  - + Xác định vector  $T_{rm}$

$$T_{rm} = T_{remaining(i)} \cdot \sigma \quad (11)$$

- + Xác định ma trận  $T_{rme}$ :

$$T_{rme} = \begin{bmatrix} T_{rm} \\ T_{rm} \\ \dots \\ T_{rm} \end{bmatrix} \quad (12)$$

+ Ma trận T chứa các vector  $T_{remaining}$  được định nghĩa ở định nghĩa 7 được tính bằng

$$T = T_{rme} - T_{rme}^T \quad (13)$$

- Tổng thời gian chuyển tiếp:

$$T_{sumtime} = T_{sumtime(i)} + T_{rm} \cdot \sigma \quad (14)$$

- Ma trận chuỗi thông chứa các vector chuỗi thông:

$$RM_{(k+1)} = \begin{bmatrix} RM_i & RM_i & RM_i & RM_i \\ 1.t_{e1} & 2.t_{e2} & \dots & n.t_{en} \end{bmatrix} \quad (15)$$

- Xác định hàm đánh giá heuristic theo (10)

*c) Thuật toán BFS kết hợp hàm heuristic xây dựng và tìm kiếm chuỗi thông*

Dưới đây là các bước thực hiện trong thuật toán BFS kết hợp hàm heuristic để xây dựng và tìm kiếm chuỗi thông cho mạng Petri

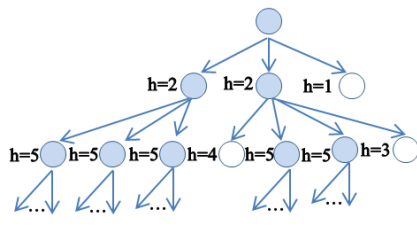
- **Bước 1:** Lập một danh sách OPEN chứa trạng thái đầu tiên.
- **Bước 2:** Nếu OPEN =  $\emptyset$  hoặc một phần tử trong OPEN có đánh dấu bằng  $M_{end}$  thì thoát khỏi thuật toán.
- **Bước 3:** Duyệt trong danh sách OPEN các danh sách có giá trị hàm heuristic  $h = h_{max}$  (với  $h_{max}$  là giá trị lớn nhất của hàm heuristic trong danh sách OPEN).
- **Bước 4:** Với mỗi một phần tử trong danh sách ở bước 3.
  - + **Bước 4.1:** Xây dựng các trạng thái tiếp theo
  - + **Bước 4.2:** Xóa ra khỏi danh sách OPEN.
  - + **Bước 4.3:** Lập lại cho đến khi hết danh sách ở bước 3.
- **Bước 5:** Thêm tất cả các trạng thái đã xây dựng ở bước 4 vào OPEN.
- **Bước 6:** Sắp xếp lại OPEN theo hàm heuristic
- **Bước 7:** Lập lại từ bước 2.

Bước k-1

Bước k

Bước k+1

Bước k+2



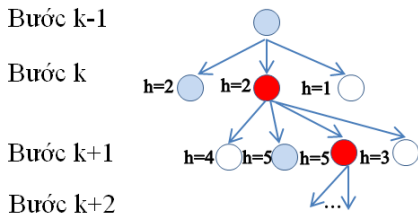
(h là hàm đánh giá heuristic)

**H. 6** Mô tả thuật toán BFS

*d) Tránh sự bùng nổ trạng thái bằng phương pháp chọn nhánh ngẫu nhiên*

Với thuật toán tìm kiếm trên, tại mỗi bước khai triển, thuật toán sẽ xây dựng tất cả các trạng thái trong danh sách OPEN thỏa mãn điều kiện tại bước 3. Phương pháp như vậy giúp ta xét được tất cả các trạng thái tốt nhất tại mỗi bước. Tuy nhiên, có khả năng số danh sách lớn, hoặc số chuyển tiếp tích cực tại mỗi danh sách nhiều sẽ xảy ra hiện tượng bùng nổ trạng thái, khi đó việc tìm kiếm cần rất nhiều thời gian và tài nguyên tính toán.

Để tránh hiện tượng này, chúng tôi đề xuất phương pháp chọn nhánh ngẫu nhiên. Tức là tại bước 3 của thuật toán, thay vì duyệt tất cả các danh sách trong OPEN thì ta chỉ chọn ngẫu nhiên một đối tượng trong danh sách có giá trị hàm heuristic lớn nhất để khai triển tiếp. Việc thực hiện như vậy sẽ tiết kiệm thời gian xử lý. Tuy nhiên, chưa hẳn kết quả tìm kiếm là kết quả tối ưu mà chỉ là kết quả tốt ở mức chấp nhận được. Điều này có thể chấp nhận do bản thân thuật toán BFS chưa phải là một thuật toán tìm kiếm tối ưu.

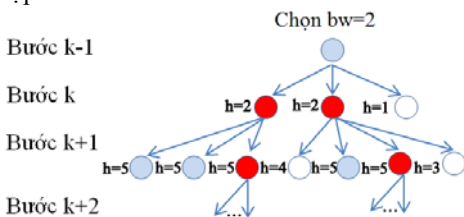


(h là hàm đánh giá heuristic)

**H. 7** Thuật toán BFS kết hợp hàm heuristic

*e) Tránh sự bùng nổ trạng thái bằng phương pháp giới hạn nhánh khai triển*

Việc chọn ngẫu nhiên một trạng thái để khai triển tiếp mà bỏ qua các trạng thái khác giúp cho việc tiếp cận kết quả nhanh hơn, ít tốn tài nguyên máy hơn. Tuy vậy nhược điểm như đã nói trên là chưa chắc đã cho ra kết quả tốt nhất. Để tăng khả năng tiếp cận lời giải tối ưu, ta cải tiến phương pháp trên bằng cách mở rộng thêm các trạng thái được xem xét (hệ số mở rộng ký hiệu là bw). Tùy theo mức độ mở rộng mà ta được các kết quả khác nhau. Nếu bw bằng số danh sách trong OPEN thì phương pháp này trở thành phương pháp đã trình bày ở 2.2.3. Nếu bw = 1 thì phương pháp này trở thành phương pháp chọn nhánh ngẫu nhiên. Tùy theo điều kiện, ta có thể chọn bw thích hợp.



(h là hàm đánh giá heuristic)

**H. 8** Kết hợp với phương pháp giới hạn nhánh khai triển

**2.3 Các kết quả**

Để tăng cường khả năng tiếp cận đến kết quả tốt nhất, ta sử dụng thuật toán giới hạn mức độ khai triển với giá trị bw=5 và cho lặp lại 20 lần để nhận kết quả tốt nhất.

Với mô hình sản xuất tự động được trình bày ở trên, ta có một số kết quả sau:

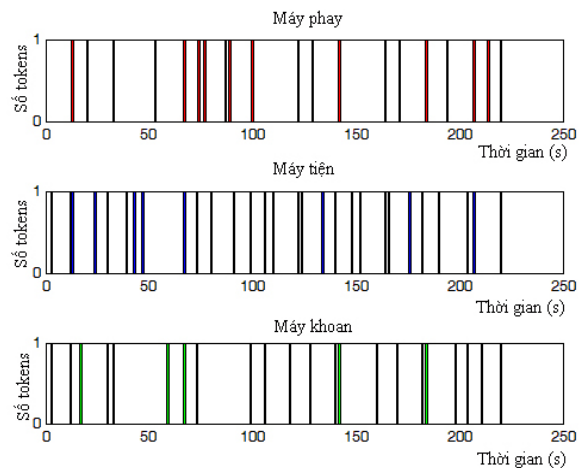
- Tổng thời gian thực hiện toàn bộ quá trình sản xuất là 221 đơn vị thời gian.
- Tiến trình thực hiện theo thứ tự chuỗi thông của mạng Petri như sau:

$\varphi = (t_0, t_{25}, t_{33}, t_{20}, t_2, t_{27}, t_{25}, t_{21}, t_{34}, t_{30}, t_1, t_{33}, t_5, t_{31}, t_{18}, t_{22}, t_3, t_{27}, t_{23}, t_4, t_{34}, t_1, t_{25}, t_{19}, t_{30}, t_{22}, t_{33}, t_3, t_{27}, t_{31}, t_{23}, t_4, t_{34}, t_{25}, t_0, t_{19}, t_{30}, t_{33}, t_{31}, t_{25}, t_{20}, t_{27}, t_2, t_{21}, t_{33}, t_5, t_{18}, t_{34}, t_{27}, t_{30}, t_{25}, t_{31}, t_{34}, t_{30}, t_1, t_{33}, t_{31}, t_{25}, t_{22}, t_{27}, t_{33}, t_3, t_{34}, t_{30}, t_{27}, t_{23}, t_{25}, t_{34}, t_{26}, t_{31}, t_4, t_{33}, t_{19}, t_{30}, t_{35}, t_{27}, t_{28}, t_{31}, t_{34}, t_{30}, t_{36}, t_1, t_{25}, t_{31}, t_{29}, t_{32}, t_{22}, t_0, t_{33}, t_3, t_{20}, t_2, t_{27}, t_{23}, t_{21}, t_4, t_5, t_{34}, t_{19}, t_{30}, t_0, t_{31}, t_{18}, t_{20}, t_2, t_0, t_{21}, t_5, t_{18}, t_{20}, t_2, t_1, t_{21}, t_5, t_{18}, t_{22}, t_3, t_{23}, t_4, t_{19})$ .

- Hiệu năng sử dụng các máy phay, tiện, khoan theo biểu đồ **Error! Reference source not found.**

Hiệu năng của các máy được đánh giá dựa trên số token của các vị trí P22 (biểu diễn máy phay chờ), P23 (biểu diễn máy tiện chờ), P24 (biểu diễn máy khoan chờ). Giá trị bằng 1 là máy đang chờ phục vụ, giá trị 0 là máy đang hoạt động.

Dựa trên biểu đồ, ta có thể thấy được rằng chu trình hoạt động dựa trên chuỗi thông vừa tính toán là tương đối tốt, các máy được sử dụng với hiệu năng cao, ít có thời gian chờ.



**H. 9** Hiệu năng sử dụng các máy công cụ

**3. Kết luận**

Trong bài báo này, chúng tôi đã xây dựng mô hình mạng Petri của một hệ thống sản xuất tự động. Qua đó, chúng tôi đề xuất một phương pháp xây dựng cây trạng thái và kết hợp tìm kiếm trên cây trạng thái bằng cách ứng dụng thuật toán tìm kiếm BFS có đánh giá bằng hàm heuristic. Ưu điểm chính của thuật toán là có thể xây dựng và xác định được chuỗi thông một mạng Petri một cách nhanh chóng. Chuỗi thông này đảm bảo có thể đi đến được trạng thái cuối của mạng Petri (là kết quả mong muốn của một quá trình sản xuất). Phương pháp này có thể sử dụng để lập kế hoạch sản xuất cho một hệ thống sản xuất.

Đối chiếu với các phương pháp của Li Xiaoling và Xing Keyi trong [7]; Kim Young Woo và Suzuki Tatsuya trong [5]; Reyes A. và Yu H. trong [13], chúng tôi nhận thấy phương pháp này đơn giản hơn, thời gian tìm kiếm chuỗi thông (cũng là xác định kế hoạch hoạt động cho hệ thống sản xuất tự động) ngắn hơn.

Về mặt kết quả, chuỗi thông tìm được mặc dù chưa thể khẳng định là tối ưu nhưng đó là kết quả ở

mức chấp nhận được, tương đương với chương trình mô phỏng mạng Petri HPSim. Kết quả này có thể kết hợp với một chương trình điều khiển kiểu mạng Petri để điều khiển hệ thống sản xuất (Phạm Trường Tùng trong [11])

Chúng tôi đề xuất các phương pháp cải tiến thuật toán để đảm bảo khả năng tìm kiếm kết quả tối ưu với thời gian ngắn hơn theo hai hướng sau:

- Xây dựng hàm heuristic: kết quả tìm kiếm của thuật toán phụ thuộc rất nhiều vào hàm đánh giá heuristic, do đó việc cải tiến hàm heuristic sao cho đánh giá đúng khả năng tìm lời giải tốt nhất ở một trạng thái nào đó là một việc rất quan trọng.
- Xây dựng một giải thuật thông minh sử dụng kết quả đánh giá của hàm heuristic để giới hạn mức tìm kiếm bw. Ý tưởng về phương pháp này đã được Mejía Gonzalo và Odrey Nicholas G. trong [10] đề xuất dựa trên thuật toán BAS (Beam A\* Search), Mejía Gonzalo và Niño Karen trong [9] sử dụng cho thuật toán HFBS (Hybrid Filtered Beam Search).

### Tài liệu tham khảo

- [1] Chen Y et al. (2017) *Deadlock recovery for flexible manufacturing systems modeled with Petri nets*. Information Sciences, vol. 381: pp. 290-303
- [2] Dotoli M et al. (2014) *A Timed Petri Nets Model for Intermodal Freight Transport Terminals*. IFAC Proceedings, vol. 47(2): pp. 176-181
- [3] Fortier PJ, Michel HE (2003) *Petri Nets*, in *Computer Systems Performance Evaluation and Prediction*. Digital Press: Burlington, pp. 279-303
- [4] Huang B, Jiang R, Zhang G (2014) *Search strategy for scheduling flexible manufacturing systems simultaneously using admissible heuristic functions and nonadmissible heuristic functions*. Computers & Industrial Engineering, vol. 71, pp. 21-26
- [5] Kim YW, Suzuki T, Narikiyo T, (2007) *FMS scheduling based on timed Petri Net model and reactive graph search*. Applied Mathematical Modelling, vol. 31(6), pp. 955-970
- [6] Lee J, Korbaa O (2004) *Modeling and scheduling of ratio-driven FMS using unfolding time Petri nets*. Computers & Industrial Engineering, vol. 46(4), pp. 639-653
- [7] Li X et al. (2017) *Total energy consumption optimization via genetic algorithm in flexible manufacturing systems*. Computers & Industrial Engineering, vol. 104, pp. 188-200
- [8] Manzhi L et al. (2009) *The research on modeling of coal supply chain based on objectoriented Petri net and optimization*. Procedia Earth and Planetary Science, vol. 1(1), pp. 1608-1616
- [9] Mejía G, Niño K (2017) *A new Hybrid Filtered Beam Search algorithm for deadlock-free scheduling of flexible manufacturing systems using Petri Nets*. Computers & Industrial Engineering, vol. 108, pp. 165-176
- [10] Mejía G, Odrey NG (2005) *An approach using petri nets and improved heuristic search for manufacturing system scheduling*. Journal of Manufacturing Systems, vol. 24(2), pp. 79-92
- [11] Phạm Trường T (2016) *Thiết kế phần mềm mạng Petri điều khiển hệ thống*. Đề tài nghiên cứu khoa học cấp cơ sở - Mã số: 01/CN1-1216
- [12] Phạm Trường T, Phạm Đăng P (2015) *Sử dụng mạng Petri để phân tích hoạt động hệ thống FMS*. Hội nghị Khoa học và công nghệ về cơ khí lần thứ IV, vol. 2, pp. 96-103
- [13] Reyes A et al. (2002) *Integrating Petri Nets and hybrid heuristic search for the scheduling of FMS*. Computers in Industry, vol. 47(1), pp. 123-138
- [14] Savsar M, Aldaihani M (2008) *Modeling of machine failures in a flexible manufacturing cell with two machines served by a robot*. Reliability Engineering & System Safety, vol. 93(10), pp. 1551-1562
- [15] Tüysüz F, Kahraman C (2010) *Modeling a flexible manufacturing cell using stochastic Petri nets with fuzzy parameters*. Expert Systems with Applications, vol. 37(5), pp. 3910-3920