

# PHÂN TÍCH TÍNH HỘI TỤ CỦA THUẬT TOÁN DI TRUYỀN LAI MỚI

LỤC TRÍ TUYẾN<sup>1</sup>, NGUYỄN HỮU MÙI<sup>2</sup>, VŨ ĐÌNH HÒA<sup>2</sup>

<sup>1</sup>*Viện Công nghệ Thông tin, Viện Hàn Lâm Khoa học và Công nghệ Việt Nam*

<sup>2</sup>*Khoa Công nghệ Thông tin, Đại học Sư phạm Hà nội*

**Tóm tắt.** Trong một bài báo gần đây, chúng tôi đã trình bày một thuật toán di truyền lai mới (NHGA) cho bài toán lập lịch Job Shop (JSP). Phương pháp mã hóa chúng tôi sử dụng là mã hóa số tự nhiên thay cho mã hóa nhị phân. Cách mã hóa này có nhiều ưu điểm, nhưng vấn đề hội tụ của nó vẫn còn là vấn đề mở trong nhiều năm nay. Bài báo này phân tích các thuộc tính hội tụ của NHGA bằng cách áp dụng các tính chất của xích Markov. Trên cơ sở phân tích xích Markov của thuật toán di truyền, chúng tôi đã chứng minh tính hội tụ tối ưu toàn cục của phương pháp mã hóa số tự nhiên.

**Từ khóa.** Lập lịch, giải thuật di truyền, hội tụ toàn cục, xích Markov.

**Abstract.** In this paper, we propose a new hybrid genetic algorithm (NHGA) for the job shop scheduling problem (JSP). The method of encoding we used was Natural coding instead of traditional binary coding. This manner of coding has a lot of advantages but its convergence has been still an open issue so far. This paper analyzes the convergence properties of the NHGA by applying the properties of Markov chain. Based on Markov chain analysis of the genetic algorithm, we show that the proposed algorithm converges to the global optimum in term of Natural coding.

**Key words.** Job shop scheduling, genetic algorithm, global convergence, Markov chain.

## 1. GIỚI THIỆU

Thuật toán di truyền (GA) phỏng theo các quá trình tiến hoá tự thích nghi của các quần thể sinh học để tối ưu hoá các hàm mục tiêu. Thuật toán này được phát triển bởi John Holland [5], GA được đặc trưng bởi các chiến lược tìm kiếm dựa trên quần thể và các toán tử di truyền: chọn lọc, đột biến và trao đổi chéo. Nakano và Yamada [11] nằm trong số những người đầu tiên đã áp dụng GA cổ điển dùng phép biểu diễn các lời giải theo số nhị phân cho bài toán lập lịch Job Shop (JSP). Sau đó họ còn đề xuất một số phương pháp kết hợp GA với các kỹ thuật tìm kiếm khác và đã thu được nhiều thành tựu đáng kể trong việc chinh phục JSP. Ulder và những người khác [10] là những người đầu tiên đề xuất phương pháp tìm kiếm cục bộ di truyền, phương pháp này lai ghép giữa GA và kỹ thuật tìm kiếm cục bộ cho JSP. Gần đây hơn, nhiều nhà nghiên cứu đã đề xuất các phương pháp lai kết hợp GA với nhiều kỹ thuật tìm kiếm khác để giải quyết bài toán này phức tạp này. Chẳng hạn như: Lee Hui Peng và những người khác [6], F. Guerriero [3], Rui Zhang và những người khác [12], Yamada [14], ... Tuy nhiên, cho tới nay chưa có phương pháp nào giải quyết triệt để bài toán này nhất là trường hợp nhiều máy và nhiều công việc.

Trong công trình nghiên cứu đã được công bố gần đây chúng tôi đã trình bày một thuật

toán di truyền lai mới cho JSP, để hiểu chi tiết về thuật toán này, bạn đọc có thể tham khảo trong [9]. Trong bài báo này chúng tôi trình bày một vấn đề vướng mắc vẫn còn để ngỏ trong mấy năm qua đó là: Chứng minh tính hội tụ của thuật toán di truyền lai mới cho bài toán lập lịch Job Shop.

## 2. THUẬT TOÁN DI TRUYỀN LAI MỚI CHO BÀI TOÁN LẬP LỊCH JSP

Thuật toán di truyền là một thuật toán mà đã trở nên rất nổi tiếng và phổ biến, vì vậy ở đây chúng tôi không đi sâu vào mô tả bài toán mà chỉ đưa ra thuật toán truyền thống và thuật toán di truyền lai mới để tập trung phân tích tính hội tụ của chúng dựa trên lý thuyết xác suất.

- **Thuật toán di truyền truyền thống**

```

Begin
t = 0
Khởi tạo P(t)
Xác định độ thích nghi của mỗi cá thể
repeat
Thực hiện lai ghép
Thực hiện đột biến
Xác định độ thích nghi của mỗi cá thể
Thực hiện chọn lọc
Xác định độ thích nghi mỗi cá thể
until một số tiêu chuẩn dừng được thỏa mãn
End

```

Thuật toán truyền thống trên không hội tụ hoàn toàn [4]. Vì vậy, chúng tôi cải tiến thuật toán trên với sự bổ sung thêm cá thể tinh hoa và thêm vào toán tử sao chép như dưới đây.

- **Thuật toán di truyền lai mới:** Siêu cá thể là cá thể có tính chất đặc biệt, nó không tham gia vào các quá trình lai ghép, đột biến hay chọn lọc, sau khi thao tác 3 toán tử cơ bản trên với quần thể, chúng ta sẽ thực hiện thêm một toán tử mới, đó là toán tử sao chép. Toán tử này có tác dụng sao chép cá thể có độ thích nghi cao hơn so với siêu cá thể ở trạng thái tương ứng vào vị trí của siêu cá thể. Thuật toán tiến hóa với siêu cá thể như sau

```

Begin
t = 0
Khởi tạo P(t) với siêu cá thể // cá thể có độ thích nghi tốt nhất//
Xác định độ thích nghi của mỗi cá thể //được chọn là siêu cá thể//
repeat
Thực hiện lai ghép
Thực hiện đột biến
Xác định độ thích nghi của mỗi cá thể
Thực hiện chọn lọc
Xác định cá thể có độ thích nghi cao nhất
Thực hiện sao chép
until một số tiêu chuẩn dừng được thỏa mãn
End

```

Với thuật toán di truyền lai mới này, chúng tôi sẽ sử dụng lý thuyết xích Markov để chứng minh bài toán hội tụ đến lời giải tối ưu toàn cục với xác suất 1 theo thời gian.

### 3. PHÂN TÍCH TÍNH HỘI TỤ CỦA THUẬT TOÁN DI TRUYỀN CHO BÀI TOÁN LẬP LỊCH JSP

#### 3.1. Thuật toán di truyền truyền thống

##### 3.1.1 Bài toán

Bài toán được giả sử với  $n$  công việc và  $m$  máy với một tuần tự công việc xác định. Mỗi lời giải coi là một cá thể, mỗi phần công việc được xử lý trên mỗi máy là một gen. Gọi  $N$  là số cá thể của quần thể. Khi đó, mỗi một trạng thái của quần thể có thể coi là một dãy mã hóa theo số tự nhiên có dài  $n \times m \times N$ , trong đó phép chiếu  $\pi_k(i)$  cho ta cá thể thứ  $k$  trong quần thể. Không gian tìm kiếm có số các trạng thái là  $(n!)^{m \times N}$ .

##### 3.1.2 Liên hệ giữa bài toán JSP và xích Markov

Như vậy, mỗi thế hệ trong thuật toán di truyền gồm một quần thể được đặc trưng bởi hàm thích nghi, và mỗi giá trị này ta coi là một trạng thái. Vì thế hệ thứ  $t + 1$  (gọi là  $Z_{t+1}$ ) sinh ra một cách ngẫu nhiên từ thế hệ thứ  $t$  ( $Z_t$ ) nên xác suất để quần thể chuyển từ trạng thái  $i$  ở thế hệ  $t$  sang trạng thái  $j$  ở thế hệ  $t + 1$  chỉ phụ thuộc vào thế hệ thứ  $t$  nên dãy  $\{Z_t, t \geq 1\}$  có thể coi là một xích Markov với không gian trạng thái chính là không gian tìm kiếm. Từ đây, ta có thể phân tích sự hội tụ theo xác suất của xích Markov này.

Bây giờ ta đi tính ma trận xác suất chuyển của xích Markov có được từ thuật toán di truyền.

- **Ma trận xác suất chuyển trạng thái của quần thể gây ra bởi toán tử lai ghép**

Xét toán tử lai, gọi  $\mathbf{C}$  là ma trận chuyển trạng thái của quần thể gây ra bởi toán tử lai ghép, ta có:

$$\mathbf{C} = \begin{pmatrix} c_{11} & \cdots & c_{1(n!)^{m \cdot N}} \\ \vdots & \ddots & \vdots \\ c_{(n!)^{m \cdot N} 1} & \cdots & c_{(n!)^{m \cdot N} (n!)^{m \cdot N}} \end{pmatrix}$$

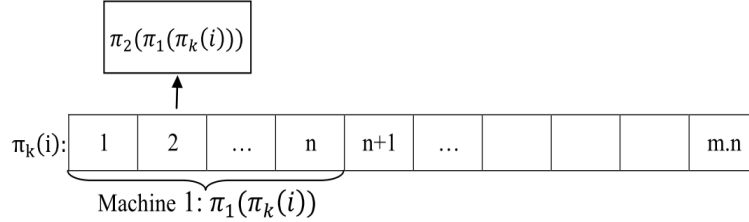
Với  $c_{ij}$  là xác suất của quần thể chuyển từ trạng thái  $i$  sang trạng thái  $j$  với xác suất lai  $p_c$ . Ta thấy với đầu vào của thuật toán tiến hóa truyền thống, xác suất lai là  $p_c \in [0, 1]$ , quá trình lai diễn ra bằng việc chọn bất kỳ một cá thể cha mẹ ở trạng thái hiện thời  $i$  với xác suất lai  $p_c$ , tiến hành cho lai ghép 3 cá thể theo luật GT hoàn toàn ngẫu nhiên [14], sau quá trình lai ghép, quần thể có thể ở trạng thái  $j$  bất kỳ với xác suất  $c_{ij}$  và

$$\sum_{j=1}^{(n!)^{m \cdot N}} c_{ij} = 1$$

xác suất  $c_{ij}$  này không phụ thuộc vào việc quần thể đang ở thế hệ thứ bao nhiêu, đang ở thời điểm nào, mà chỉ phụ thuộc vào xác suất lai  $p_c$  cố định theo đầu vào của thuật giải. Như vậy, ma trận chuyển trạng thái sinh ra bởi phép lai  $\mathbf{C}$  của quần thể là một ma trận ngẫu nhiên và cố định không phụ thuộc vào trạng thái hiện thời cũng như thời điểm của quần thể.

- **Ma trận xác suất chuyển trạng thái của quần thể gây ra bởi toán tử đột biến**

Gọi  $i$  là trạng thái tại thời điểm  $t$ , và  $\pi_k(i)$  là cá thể thứ  $k$  trong quần thể gồm  $N$  cá thể,  $\pi_h(\pi_k(i))$  là máy thứ  $h$  trong  $\pi_k(i)$  và  $\pi_l(\pi_h(\pi_k(i)))$  là gene vị trí thứ  $l$ , được mô tả trong Hình 3.1. Thuật toán đột biến xảy ra với  $\pi_k(i)$  có thể mô tả vắn tắt như sau:



Hình 3.1. Gene tại vị trí thứ 2 của trạng thái  $i$  của quần thể

1. Chọn cá thể  $\pi_k(i)$  để thực hiện đột biến với xác suất  $p_m > 0$  (rất nhỏ). Vậy xác suất để đột biến không xảy ra là  $1 - p_m$ .
2. Khi đột biến xảy ra đối với  $\pi_k(i)$ , quá trình đột biến như sau:  
 Với các máy  $\pi_h(\pi_k(i))$ ,  $h = 1, \dots, m$ , tại mỗi máy trong số này:  
 + Không gây đột biến với xác suất  $p > 0$  (xấp xỉ 1).  
 + Ngược lại, thay thế máy thứ  $h$  bởi một hoán vị bất kỳ, khác đồng nhất, với xác suất ngang nhau. Vậy xác suất cho mỗi sự thay đổi này là:

$$\frac{1-p}{n!-1}. \quad (3.1)$$

Với thuật toán trên, một cá thể có thể đột biến thành cá thể bất kỳ trong không gian tìm kiếm với xác suất dương, xác suất này chỉ phụ thuộc vào  $p$ . Gọi  $m_{ij}$  là xác suất chuyển của quần thể từ trạng thái  $i$  sang trạng thái  $j$  thông qua đột biến, thì  $m_{ij} > 0, \forall i, j$ . Xác suất  $m_{ij}$  này chỉ phụ thuộc vào  $p$  (không tính các hằng số  $n$  đã cho), và có thể tính cụ thể được như sau:

Xét hai trạng thái bất kỳ  $i, j$ . Giả sử  $i$  và  $j$  có  $K$  cá thể giống nhau và  $N - K$  cá thể khác nhau ở cùng một thứ tự trong quần thể của chúng, thì:

- + Xác suất cho  $K$  cá thể đó giữ nguyên (không đột biến) là  $(1 - p_m)^K$ .
- + Với  $N - K$  cặp cá thể khác nhau,  $\pi_{k_t}(i)$  và  $\pi_{k_t}(j)$ ,  $t = 1, 2, \dots, N - K$ , gọi  $L_{k_t}$  số các máy mà có các gene giống nhau giữa  $\pi_{k_t}(i)$  và  $\pi_{k_t}(j)$ , vậy  $m - L_{k_t}$  là số máy mà có các gene khác nhau.

- Xác suất xảy ra để  $L_{k_t}$  máy giống nhau này là  $p^{L_{k_t}}$ .

- Xác suất để  $m - L_{k_t}$  máy khác nhau còn lại theo 3.1 là  $\left(\frac{1-p}{n!-1}\right)^{m-L_{k_t}}$ .

Do đó,

$$m_{ij} = (1 - p_m)^K \cdot p_m^{N-K} \cdot \prod_{t=1}^{N-K} \left( p^{L_{k_t}} \cdot \left( \frac{1-p}{n!-1} \right)^{m-L_{k_t}} \right) > 0. \quad (3.2)$$

Vậy,  $\mathbf{M} = [m_{ij}]$  là một ma trận xác suất dương.

- Ma trận xác suất chuyển trạng thái của quần thể gây ra bởi toán tử chọn lọc

Toán tử thứ 3 sẽ được thực hiện để sinh ra một quần thể mới từ quần thể cũ là toán tử chọn lọc. Gọi  $\mathbf{S}$  là ma trận chuyển trạng thái của quần thể gây ra bởi toán tử chọn lọc, ta có:

$$\mathbf{S} = \begin{pmatrix} s_{11} & \cdots & s_{1(n!)^{m.N}} \\ \vdots & \ddots & \vdots \\ s_{(n!)^{m.N}} & \cdots & s_{(n!)^{m.N}(n!)^{m.N}} \end{pmatrix}$$

Với  $s_{ij}$  là xác suất quần thể chuyển từ trạng thái  $i$  sang trạng thái  $j$  gây ra bởi toán tử chọn lọc.

Xác suất để một cá thể  $\pi_k(i)$  trong quần thể hiện thời ở trạng thái  $i$  được giữ lại trong quần thể mới là

$$p_k = f(\pi_k(i)) / \sum_{h=1}^N f(\pi_h(i)) \quad (3.3)$$

Như vậy, xác suất biến đổi từ trạng thái  $i$  sang trạng thái  $j$  của quần thể gây ra bởi toán tử chọn lọc chỉ phụ thuộc vào hàm  $f$  mà không phụ thuộc vào trạng thái hiện thời cũng như thời điểm hiện tại của quần thể.

Từ 3.3, xác suất để quần thể hiện thời giữ nguyên trạng thái  $i$  sau phép chọn lọc sẽ là:

$$s_{ii} = \frac{\prod_{k=1}^N f(\pi_k(i))}{\left(\sum_{h=1}^N f(\pi_h(i))\right)^N}. \quad (3.4)$$

Vì hàm  $f$  là một hàm luôn dương nên từ 3.4 ta suy ra:  $s_{ij} > 0$ .

Ma trận chuyển trạng thái  $\mathbf{S}$  luôn có ít nhất một phần tử trên một cột là dương nên ma trận  $\mathbf{S}$  là ma trận thỏa mãn cột.

Tóm lại, quá trình sinh ra một thế hệ mới từ thế hệ cũ theo giải thuật tiến hóa truyền thống sẽ được thực hiện thông qua 3 toán tử lai ghép, đột biến và chọn lọc với các ma trận chuyển trạng thái  $\mathbf{C}$ ,  $\mathbf{M}$  và  $\mathbf{S}$  tương ứng. Gọi  $\mathbf{P}$  là ma trận chuyển trạng thái tổng hợp từ thế hệ  $t$  sang thế hệ  $t + 1$ , ta có:

$$\mathbf{P} = \begin{pmatrix} p_{11} & \cdots & p_{1(n!)^{m.N}} \\ \vdots & \ddots & \vdots \\ p_{(n!)^{m.N}} & \cdots & p_{(n!)^{m.N}(n!)^{m.N}} \end{pmatrix} = \mathbf{C} \times \mathbf{M} \times \mathbf{S}$$

Vì các ma trận  $\mathbf{C}$ ,  $\mathbf{M}$  và  $\mathbf{S}$  là các ma trận ngẫu nhiên không phụ thuộc vào thời điểm cũng như trạng thái hiện thời của quần thể nên ma trận  $\mathbf{P}$  với các phần tử  $p_{ij}$  cũng không phụ thuộc vào trạng thái hiện thời cũng như thời điểm của quần thể.

Lại có ma trận  $\mathbf{C}$ ,  $\mathbf{M}$  và  $\mathbf{S}$  là các ma trận ngẫu nhiên,  $\mathbf{M}$  là ma trận dương và  $\mathbf{S}$  là ma trận thỏa mãn cột, từ **Bổ đề 3.1** sau đây ta suy ra ma trận xác suất chuyển trạng thái tổng hợp  $\mathbf{P}$  là một ma trận ngẫu nhiên dương.

**Bổ đề 3.1.** (xem [4])

Gọi  $\mathbf{C}$ ,  $\mathbf{M}$ ,  $\mathbf{S}$  là các ma trận ngẫu nhiên, trong đó  $\mathbf{M}$  là một ma trận dương và  $\mathbf{S}$  là ma trận thỏa mãn cột. Khi đó tích  $\mathbf{C} \times \mathbf{M} \times \mathbf{S}$  là một ma trận ngẫu nhiên dương.

Như vậy thuật toán tiến hóa truyền thống có thể được mô tả thông qua một xích Markov với trạng thái của quần thể nằm trong không gian trạng thái và ma trận xác suất chuyển trạng thái  $\mathbf{P}$  dương. Ta suy ra thuật toán này chính là một xích Markov Ergodic (xích Markov Ergodic xem [7]).

### 3.2. Thuật toán di truyền lai mới với siêu cá thể

Với sự xuất hiện của siêu cá thể, lực lượng của quần thể sẽ tăng lên  $N + 1$ , như vậy không gian trạng thái của quần thể sẽ không còn là  $(n!)^{m \times N}$  nữa, mà sẽ là  $(n!)^{m \times (N+1)}$ .

Ta đặt vị trí của siêu cá thể ở đầu của chuỗi mã hóa chiều dài  $(N + 1) \times m \times n$  và ta có thể truy xuất đến siêu cá thể đó thông qua phép chiếu  $\pi_0(i)$  khi quần thể ở trạng thái  $i$ . Với mỗi trạng thái của quần thể, chúng ta có 1 siêu cá thể tương ứng đứng ở đầu, ta sắp xếp không gian các trạng thái theo thứ tự giảm dần độ thích nghi của siêu cá thể (tức là  $i < j \rightarrow f(\pi_0(i)) > f(\pi_0(j))$ ).

Khi đó ma trận chuyển trạng thái gây ra bởi các toán tử lai, đột biến và chọn lọc sẽ có kích thước là  $(n!)^{m \times (N+1)} \times (n!)^{m \times (N+1)}$ , gọi  $\mathbf{C}^+$ ,  $\mathbf{M}^+$ ,  $\mathbf{S}^+$  lần lượt là ma trận chuyển trạng thái gây ra bởi các toán tử lai ghép, đột biến và chọn lọc lên quần thể có chứa siêu cá thể, vì các siêu cá thể không tham gia vào quá trình tiến hóa nên các ma trận  $\mathbf{C}^+$ ,  $\mathbf{M}^+$ ,  $\mathbf{S}^+$  sẽ có dạng:

$$\mathbf{C}^+ = \begin{pmatrix} \mathbf{C} & & \\ & \ddots & \\ & & \mathbf{C} \end{pmatrix}; \mathbf{M}^+ = \begin{pmatrix} \mathbf{M} & & \\ & \ddots & \\ & & \mathbf{M} \end{pmatrix}; \mathbf{S}^+ = \begin{pmatrix} \mathbf{S} & & \\ & \ddots & \\ & & \mathbf{S} \end{pmatrix}$$

Với mỗi đường chéo bao gồm  $(n!)^m$  ma trận  $\mathbf{C}$ ,  $\mathbf{M}$ ,  $\mathbf{S}$  đã được xét trong Mục 3.1 với giải thuật tiến hóa chưa có siêu cá thể, từ đó ta có:

$$\mathbf{C}^+ \mathbf{M}^+ \mathbf{S}^+ = \begin{pmatrix} \mathbf{CMS} & & \\ & \ddots & \\ & & \mathbf{CMS} \end{pmatrix} = \begin{pmatrix} \mathbf{P} & & \\ & \ddots & \\ & & \mathbf{P} \end{pmatrix}$$

Toán tử copy được thực hiện bằng một ma trận nâng cấp  $\mathbf{U}$  với kích thước  $(n!)^{m \cdot (N+1)} \times (n!)^{m \cdot (N+1)}$ , trong đó các phần tử được định nghĩa như sau:

Xét một quần thể có siêu cá thể và quần thể đang ở trạng thái  $i$ , gọi  $\pi_l(i)$  là cá thể có độ thích nghi cao nhất trong số các cá thể thường, nghĩa là:

$$f(\pi_l(i)) = \max\{f(\pi_k(i)) | k = 1, \dots, N\} \quad (3.5)$$

thì,  $u_{ij} = 1$  nếu  $f(\pi_0(i)) < f(\pi_l(i))$  với  $j = (\pi_l(i), \pi_1(i), \dots, \pi_N(i))$  thuộc không gian trạng thái. Các trường hợp còn lại,  $u_{ii} = 1$ .

Toán tử sao chép sẽ thay thế cá thể thường có độ thích nghi cao hơn so với siêu cá thể ở cùng thế hệ vào vị trí của siêu cá thể, nếu không có cá thể nào thỏa mãn, quần thể sẽ giữ nguyên. Ta thấy ma trận  $\mathbf{U}$  cũng chỉ phụ thuộc vào giá trị của hàm  $f$ , như vậy ma trận  $\mathbf{U}$  đối với một bài toán có kích thước cố định là cố định, vì các siêu cá thể được sắp xếp theo thứ tự giảm dần của độ thích nghi của siêu cá thể nên ma trận  $\mathbf{U}$  sẽ có dạng:

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_{11} & & \\ & \ddots & \\ \mathbf{U}_{(n!)^m 1} & & \mathbf{U}_{(n!)^m (n!)^m} \end{pmatrix}$$

trong đó các ma trận  $\mathbf{U}_{ij}$  có kích thước  $(n!)^{m \cdot N} \times (n!)^{m \cdot N}$ .

Giả sử **Bài toán 3.1** chỉ có một nghiệm tối ưu, như vậy các trạng thái từ 1 đến  $(n!)^{m \times N}$  là các trạng thái mà tất cả có siêu cá thể (chung một siêu cá thể) mang nghiệm tối ưu (do cách đánh số không gian trạng thái), như vậy trong ma trận nâng cấp  $\mathbf{U}$  ở trên, chỉ có ma

trận con  $\mathbf{U}_{11}$  là ma trận đơn vị trong khi tất cả các ma trận  $\mathbf{U}_{aa}$  với  $a \geq 2$  là các ma trận đơn vị với một vài phần tử trên đường chéo bằng 0. Với  $\mathbf{P} = \mathbf{CMS}$  thì ma trận chuyển cho bài toán NHGA,  $\mathbf{P}^+$ , trở thành

$$\begin{aligned} \mathbf{P}^+ &= \begin{pmatrix} \mathbf{P} & & \\ & \ddots & \\ & & \mathbf{P} \end{pmatrix} \begin{pmatrix} \mathbf{U}_{11} & & \\ & \ddots & \\ \mathbf{U}_{(n!)^{m_1}} & & \mathbf{U}_{(n!)^m(n!)^m} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{P}\mathbf{U}_{11} & & \\ \vdots & \ddots & \\ \mathbf{P}\mathbf{U}_{(n!)^{m_1}} & \cdots & \mathbf{P}\mathbf{U}_{(n!)^m(n!)^m} \end{pmatrix} \end{aligned}$$

Từ đây, ta có thể chứng minh được thuật toán tiến hóa với siêu cá thể có tính chất hội tụ hoàn toàn.

Thật vậy, vì  $\mathbf{U}_{11}$  là ma trận đơn vị nên  $\mathbf{P}\mathbf{U}_{11} = \mathbf{P}$  là một ma trận ngẫu nhiên và dương. Từ các ma trận  $\mathbf{U}_{aa}$  với  $a \geq 2$  là các ma trận đơn vị với vài phần tử trên đường chéo chính bằng 0 nên  $\mathbf{P}\mathbf{U}_{k1} \neq 0$  với mọi  $k \in \{2, \dots, (n!)^m\}$ . Do đó, các ma trận con  $\mathbf{P}\mathbf{U}_{a1}$  với  $a \geq 2$  có thể hợp lại thành ma trận chữ nhật  $\mathbf{R} \neq 0$  và ma trận phụ hợp của ma trận  $\mathbf{P}$  trong  $\mathbf{P}^+$ ,  $\mathbf{T}$ , cũng khác 0:

$$\mathbf{T} \begin{pmatrix} \mathbf{P}\mathbf{U}_{22} & & \\ \vdots & \ddots & \\ \mathbf{P}\mathbf{U}_{(n!)^{m_2}} & \cdots & \mathbf{P}\mathbf{U}_{(n!)^m(n!)^m} \end{pmatrix} \neq 0$$

Như vậy, ma trận  $\mathbf{P}^+$  có thể được viết thành:

$$\mathbf{P}^+ = \begin{pmatrix} \mathbf{P} & 0 \\ \mathbf{R} & \mathbf{T} \end{pmatrix} \tag{3.6}$$

Mặt khác, một định lý trong lý thuyết xích Markov phát biểu như sau:

**Định lý 3.1.** ([7], page 126) *Gọi  $\mathbf{P}$  là ma trận chuyển cỡ  $n \times n$  của một xích Markov thỏa mãn  $\mathbf{P} = \begin{pmatrix} C & 0 \\ R & T \end{pmatrix}$ , với  $C$  cỡ  $m \times m$  là ma trận ngẫu nhiên Ergodic và  $\mathbf{R}, \mathbf{T} \neq 0$ , thì ta có:*

$$\mathbf{P}^\infty = \lim_{k \rightarrow \infty} \mathbf{P}^k = \lim_{k \rightarrow \infty} \begin{pmatrix} \mathbf{C}^k & 0 \\ \sum_{i=0}^k \mathbf{T}^i \mathbf{R} \mathbf{C}^{k-i} & \mathbf{T}^k \end{pmatrix} = \begin{pmatrix} \mathbf{C}^\infty & 0 \\ \mathbf{R}^\infty & 0 \end{pmatrix}$$

là một ma trận ổn định với  $\mathbf{P}^\infty = \mathbf{1}'\Pi^\infty$  ( $\mathbf{1}'$  là ma trận cột toàn phần tử 1) và  $\Pi^\infty = \Pi^{(0)} \times \mathbf{P}^\infty$  là các vector hàng ổn định của ma trận  $\mathbf{P}^\infty$  mà không phụ thuộc vào phân phối xác suất ban đầu  $\Pi^{(0)}$  của xích Markov và thỏa mãn điều kiện:  $\pi_i^\infty > 0$  với mọi  $1 \leq i \leq m$  và  $\pi_i^\infty = 0$  với mọi  $m \leq i \leq n$ .

Theo 3.6, ma trận  $\mathbf{P}^+$  thỏa mãn các điều kiện của **Định lý 3.1**, nên ta có ngay

$$(\mathbf{P}^+)^\infty = \begin{pmatrix} \mathbf{P}^\infty & 0 \\ \mathbf{R}^\infty & 0 \end{pmatrix} \tag{3.7}$$

Như vậy, khi số thế hệ tiến tới vô cùng, phân phối xác suất trạng thái của quần thể tập trung hoàn toàn vào  $(n!)^{m \times N}$  trạng thái đầu, mà tất cả các trạng thái này đều chứa nghiệm tối ưu (gọi là  $f^*$ ). Như vậy:

$$\lim_{t \rightarrow \infty} P(Z_t \rightarrow f^*) = 1. \tag{3.8}$$

Tức là thuật toán tiến hóa với siêu cá thể có tính chất hội tụ hoàn toàn.

#### 4. KẾT LUẬN

Bài báo này đã chứng minh tính hội tụ tối ưu toàn cục của một thuật toán di truyền lai mới mà chúng tôi đề xuất cho bài toán lập lịch Job Shop [9]. Trường hợp chứng minh cho tính hội tụ của thuật toán di truyền dùng mã hóa nhị phân đã được Günter Rudolph giải quyết vào năm 1994. Tuy nhiên, mã hóa nhị phân có nhiều hạn chế trong áp dụng thuật toán di truyền cho các bài toán phức tạp trong thực tế. Trong bài báo này, trên cơ sở phân tích xích Markov của thuật toán di truyền, chúng tôi đã chứng minh tính hội tụ tối ưu toàn cục của phương pháp mã hóa số tự nhiên. Đây là vấn đề vẫn để ngỏ trong nhiều năm qua.

#### TÀI LIỆU THAM KHẢO

- [1] B. Giffler and G. L. Thompson, Algorithms for solving production scheduling problems, *Operations Research* **8** (4) (1960) 487–503.
- [2] E. Seneta, *None-negative Matrices and Markov Chains*, 2nd edition, New York, Springer, 1981.
- [3] F. Guerriero, Hybrid rollout approaches for the job shop scheduling problem, *Journal Optimization Theory and Applications* **139** (2008) 419–438.
- [4] Günter Rudolph, Convergence analysis of canonical genetic algorithms, *IEEE Transactions on Neural Networks, special issue on evolutionary computation* **5** (1) (1994).
- [5] J. H. Holland, “Adaptation in natural and artificial systems”, Univ. of Michigan Press, 1975.
- [6] Lee Hui Peng, Sutinah Salim. A Modified Giffler and Thompson Genetic Algorithm on the job shop scheduling problem, *MATEMATIKA (Universiti Teknologi Malaysia)*, Vol. 22, No. 2, pp.91-107, 2006.
- [7] M. Iosifescu, *Finite Markov Processes and Their Applications*, Chichester: Wiley, 1980.
- [8] Nguyen Duy Tien, *Probability Models and Applications, Part 1-Markov Chains and Its Application*, Publisher of Hanoi National University, 2001.
- [9] Nguyen Huu Mui & Vu Dinh Hoa, A new hybrid genetic algorithm for Job Shop Scheduling Problem, *Proceedings of the 5-th National Conference Of Science And Technology (FAIR 2011)*, Đồng Nai, 2011 (238–248).
- [10] N. L. J. Ulder, E. Pesch, P. J. M. Van Laarhoven, J. H. Bandelt, and E. H. L. Aarts, Genetic local search algorithm for the traveling salesman problem, *Parallel Problem Solving from Nature* **1** (1994) 109–116.
- [11] R. Nakano and T. Yamada, Conventional genetic algorithm for job shop problems, *Proceedings of the fourth International Conference on Genetic Algorithms, (ICGA’ 91)*, University of California, San Diego, July 13-16, 1991 (474–479).
- [12] Rui Zhang and Cheng Wu, A hybrid approach to large-scale job shop scheduling, *Application Intelligence* **32** (2010) 47–59.
- [13] T.E. Davis and J.C. Principe, A simulated annealing like convergent theory for the simple genetic algorithm, *Proceedings of the fourth International Conference on Genetic Algorithms, (ICGA’ 91)*, University of California, San Diego, July 13-16, 1991 (174–181).
- [14] T. Yamada, “Studies on metaheuristics for jobshop and flowshop scheduling problems”, Kyoto University, Kyoto - Japan, 2003.

Ngày nhận bài 14 - 7 - 2012  
 Nhận lại sau sửa ngày 16 - 6 - 2013