

## VỀ BIẾN THIÊN CỦA VẾT MÙI TRONG PHƯƠNG PHÁP ACO VÀ CÁC THUẬT TOÁN MỚI\*

ĐỖ ĐỨC ĐÔNG<sup>1</sup>, HOÀNG XUÂN HUẤN<sup>2</sup>

<sup>1</sup>*Viện Công nghệ thông tin, Đại học Quốc gia Hà Nội*

<sup>2</sup>*Đại học Công nghệ, Đại học Quốc gia Hà Nội*

**Tóm tắt.** Trong các thuật toán tối ưu hóa đàn kiến (Ant Colony Optimization-ACO), vết mùi biểu thị thông tin học tăng cường để tìm kiếm lời giải cho các bài toán tối ưu tổ hợp NP khó. Bài báo trình bày một số phân tích toán học về tính biến thiên của vết mùi trong hai thuật toán ACO thông dụng nhất: ACS (Ant Colony System) và MMAS (Max-Min Ant System), các phân tích này là cơ sở cho đề xuất các quy tắc cập nhật mùi mới.

**Abstract.** In Ant colony optimization (ACO) algorithms, pheromone trails express reinforcement learning information to find solutions for NP-hard problems in combinatorial optimization. This paper presents mathematical analysis on behavior of pheromone trail in two most popular ACO algorithms: ACS (Ant Colony System) and MMAS (Max-Min Ant System), and suggests new ideas for improving pheromone update rules.

### 1. GIỚI THIỆU

Tối ưu hóa đàn kiến (Ant colony optimization-ACO) do Dorigo đề xuất [6] đã thu hút nhiều người quan tâm nghiên cứu và ứng dụng để giải các bài toán tối ưu tổ hợp khó [1, 2, 7, 11, 15, 16].

Trên đường đi, mỗi con kiến thực để lại một vết hoá chất gọi là vết mùi (pheromone trail) và theo vết mùi của các con kiến khác để tìm đường đi. Đường có nồng độ vết mùi càng cao thì càng có nhiều khả năng được các con kiến chọn. Nhờ cách giao tiếp gián tiếp này [7] đàn kiến tìm được đường đi ngắn nhất. Theo ý tưởng này, các thuật toán ACO sử dụng kết hợp thông tin kinh nghiệm (heuristic) và học tăng cường qua các vết mùi của các con kiến nhân tạo để giải các bài toán tối ưu tổ hợp bằng cách đưa về bài toán tìm đường đi tối ưu trên đồ thị cấu trúc tương ứng của bài toán.

Thuật toán ACO đầu tiên [6] là hệ kiến (AS) giải bài toán người chào hàng (TSP), đến nay đã có nhiều biến thể được đề xuất, thông dụng nhất là hệ đàn kiến (ACS) [7, 8] và hệ kiến Max-Min (MMAS) [18]. Chúng được áp dụng rộng rãi [1, 7, 11, 16, 18] để giải nhiều bài toán tối ưu tổ hợp và hiệu quả nổi trội của chúng đã được chứng tỏ bằng thực nghiệm. Ngoài các ứng dụng, có những nghiên cứu về đặc tính của thuật toán [5, 9, 11, 12, 13, 15, 17] như ảnh hưởng của vết mùi, thời gian chạy và tính hội tụ, ...

Trong [17,18] Stützle và Dorigo đã chứng minh tính hội tụ tới lời giải tối ưu của các thuật toán MMAS nhưng chưa tính tới ảnh hưởng của tham số heuristic, khuynh hướng hội tụ của

---

\*Nghiên cứu được hoàn thành dưới sự hỗ trợ từ Quỹ phát triển KHCNQG NAFOSTED

vết mùi được xét khi giả thiết đã tìm được lời giải tối ưu. Còn Gutjahr [10,11] thì đưa về xét biến thiên của một quá trình Markov để rút ra tính hội tụ tới lời giải tối ưu và sự hội tụ của cường độ vết mùi cho các biến thể của thuật toán MMAS mà chưa khảo sát cho ACS.

Tuy nhiên trong các bài toán tối ưu tổ hợp thì số phương án là hữu hạn nên kết quả về việc xác suất tìm thấy lời giải hội tụ về 1 khi số lần lặp dần ra vô hạn là tầm thường. Trong bài báo này sẽ phân tích chi tiết hơn về các đặc tính biến thiên của vết mùi trong các thuật toán ACO, trên cơ sở đó đề xuất các quy tắc cập nhật mùi mới. Kết quả thực nghiệm cho thấy ưu điểm của các đề xuất này.

Bài báo gồm: Mục 2 giới thiệu phương pháp tối ưu đàn kiến và hai thuật toán thông dụng: ACS và MMAS. Mục 3 trình bày các kết quả khảo sát xu thế thay đổi của vết mùi trong các thuật toán ACO. Các quy tắc cập nhật mùi mới được giới thiệu trong Mục 4. Mục 5 giới thiệu kết quả thực nghiệm đối với bài toán TSP để minh họa cho các đề xuất mới. Và sau cùng là phần kết luận.

## 2. PHƯƠNG PHÁP TỐI ƯU HÓA ĐÀN KIẾN (ACO)

Trong các thuật toán ACO, các hệ ACS và MMAS là hai thuật toán thông dụng nhất và sẽ được nghiên cứu trong bài này. Vì vậy, trước khi giới thiệu thuật toán ACO tổng quát, ta tóm tắt bài toán TSP và hai thuật toán ACS, MMAS [7,8,18].

### 2.1. ACS và MMAS giải bài toán TSP

#### 2.1.1. Bài toán TSP

Bài toán TSP được phát biểu như sau. Cho  $n$  thành phố, người chào hàng cần tìm một chu trình có đường đi ngắn nhất qua mỗi thành phố đúng một lần. Để đơn giản, ta xét bài toán trên đồ thị vô hướng  $G = (V, E)$  trong đó tập đỉnh  $V$  ( $n$  đỉnh) ký hiệu tập các thành phố và  $E$  là tập các cạnh,  $(i, j)$  biểu thị đường đi nối các đỉnh  $i, j \in V$  và có độ dài  $d_{i,j}$  ( $d_{i,j}$  là khoảng cách từ thành phố  $i$  tới thành phố  $j$ ).

Để giải bài toán này, phương pháp ACO dùng biến vết mùi  $\tau_{i,j}$  kết hợp với mỗi cạnh  $(i, j)$  và ban đầu được khởi tạo bởi giá trị  $\tau_0$ . Có  $m$  con kiến nhân tạo, ở bước lặp  $t$  chúng thực hiện các thủ tục *xây dựng lời giải ngẫu nhiên* và *cập nhật mùi* theo từng thuật toán. Sau một số bước lặp cho trước, lời giải tốt nhất tìm được là lời giải của bài toán.

#### 2.1.2. Hệ ACS (Ant Colony System)

*Xây dựng lời giải.* Ban đầu mỗi con kiến được đặt ngẫu nhiên ở một đỉnh xuất phát và thăm các đỉnh khác để xây dựng đường đi với thủ tục *bước ngẫu nhiên* theo quy tắc chuyển trạng thái sau.

*Quy tắc chuyển trạng thái.* Giả sử con kiến  $k$  đang ở đỉnh  $i$ , nó chọn đỉnh  $s$  tiếp theo nhờ quy tắc:

$$s = \begin{cases} \operatorname{argmax}_{u \in J_k(i)} \{ \tau_{i,u}(t) \eta_{i,u}^\alpha \} & : q \leq q_0, \\ j & : q > q_0, \end{cases} \quad (1)$$

trong đó  $q$  là giá trị chọn ngẫu nhiên thuộc khoảng  $(0, 1)$ ,  $q_0 \in (0, 1)$  là tham số cho trước,  $\eta_{i,j} = \frac{1}{d_{i,j}}$ , còn  $j$  là đỉnh được chọn theo xác suất cho bởi công thức 2 với  $\alpha = 1$ .

$$P_{i,j}^k(t) = \begin{cases} \frac{\tau_{i,j}^\alpha(t)\eta_{i,j}^\beta}{\sum_{u \in J_k(i)} \tau_{i,u}^\alpha(t)\eta_{i,u}^\beta} & , \forall j \in J_k(i) \\ 0 & , \forall j \notin J_k(i). \end{cases} \quad (2)$$

*Quy tắc cập nhật mùi.* Mùi sẽ được cập nhật theo cả hai quy tắc địa phương và toàn cục.

+ *Cập nhật mùi địa phương.* Trong khi xây dựng lời giải của mình, mỗi con kiến sẽ cập nhật mùi cho cạnh  $(i, j)$  mà nó đi qua theo công thức

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \rho\tau_0. \quad (3)$$

+ *Cập nhật mùi toàn cục.* Có hai cách cập nhật mùi toàn cục được trình bày tương ứng trong [7, 8].

a) Cập nhật mùi [7] được áp dụng cho những cạnh  $(i, j)$  thuộc đường đi ngắn nhất  $w^*(t)$  theo công thức

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j} \quad (4)$$

trong đó  $\Delta\tau_{i,j} = \rho L_{gb}^{-1}$  và  $L_{gb}(t)$  là độ dài đường đi ngắn nhất  $w^*(t)$  cho đến bước lặp  $t$  hoặc của bước lặp  $t$  (tương ứng gọi là lời giải G-best hoặc i-best). Về sau ta sẽ xét quy tắc này khi nói đến ACS.

b) Cập nhật mùi [8] áp dụng cho tất cả các cạnh theo công thức 4 trong đó:

$$\Delta\tau_{i,j} = \begin{cases} \rho L_{gb}^{-1} & , \forall (i, j) \in w(t) \\ 0 & , \forall (i, j) \notin w(t) \end{cases} \quad (5)$$

### 2.1.3. Hệ MMAS (Max-MinAnt System)

*Xây dựng lời giải.* Trong hệ này, thủ tục xây dựng lời giải thực hiện khác với ACS là chỉ dùng quy tắc chuyển trạng thái ngẫu nhiên với xác suất cho bởi 2.

*Quy tắc cập nhật mùi.* Với hai số dương  $\tau_{min}$  và  $\tau_{max}$  thích hợp cho trước, ban đầu khởi tạo  $\tau_0 = \tau_{max}$ . Ở bước lặp  $t$ , trước hết vết mùi trên mỗi cạnh được cập nhật toàn cục theo công thức 4 và 6. Sau đó thực hiện chặn mùi như sau

$$\tau_{i,j} = \begin{cases} \tau_{i,j} & , \tau_{min} \leq \tau_{i,j} \leq \tau_{max} \\ \tau_{min} & : \tau_{i,j} \leq \tau_{min} \\ \tau_{max} & : \tau_{i,j} \geq \tau_{max}. \end{cases} \quad (6)$$

## 2.2. Bài toán tổng quát

Bây giờ ta sẽ phát biểu bài toán tối ưu tổ hợp và thuật toán ACO tổng quát dựa trên cách phát biểu trong [11,17].

Xét bài toán cực tiểu hoá  $(S, f, \Omega)$ , trong đó  $S$  là tập hữu hạn trạng thái,  $f$  là hàm mục tiêu xác định trên  $S$  còn  $\Omega$  là các ràng buộc để xác định  $S$  qua các thành phần của tập hữu hạn  $C$  và các liên kết của tập này. Các tập  $S, C$  và  $\Omega$  có các đặc tính sau.

1)  $C = \{c_1, \dots, c_n\}$  là tập hữu hạn gồm  $n$  thành phần. Ta ký hiệu  $X$  là tập các dãy trong  $C$  độ dài không quá  $h : X = \{ \langle u_0, \dots, u_k \rangle \mid u_i \in C \forall i \leq k \leq h \}$ .

2) Tồn tại tập con  $X^*$  của  $X$  và ánh xạ  $\varphi$  từ  $X^*$  lên  $S$  sao cho  $\varphi^{-1}(s)$  không rỗng với mọi  $s \in S$ . Trong đó tập  $X^*$  có thể xây dựng được từ tập con  $C_0$  nào đó của  $C$  nhờ mở rộng tuần tự dưới đây.

3) Từ  $C_0$  mở rộng được thành  $X^*$  theo thủ tục tuần tự:

i)  $x_0 = \langle u_0 \rangle$  là mở rộng được với mọi  $u_1 \in C_0$ .

ii) Nếu  $x_k = \langle u_0, \dots, u_k \rangle$  là mở rộng được thì từ các ràng buộc  $\Omega$  xác định được tập con  $J(x_k)$  của  $C$  sao cho với mọi  $u_{k+1} \in J(x_k)$  thì  $x_{k+1} = \langle u_0, \dots, u_k, u_{k+1} \rangle$  là mở rộng được hoặc  $x_k \in X^*$  khi  $J(x_k)$  là rỗng.

iii) Với mọi  $u_0 \in C_0$ , thủ tục mở rộng nêu trên xây dựng được mọi phần tử của  $X^*$ .

*Vi dụ 2.1.* Trong bài toán TSP thì  $S$  là các chu trình trên đồ thị đầy,  $f$  là độ dài đường đi,  $\Omega$  là ràng buộc các chu trình qua mọi đỉnh và mỗi đỉnh qua đúng một lần,  $C_0$  là tập các đỉnh của đồ thị, nếu  $x_k = \langle u_0, \dots, u_k \rangle$  là đường đi qua  $u_0, \dots, u_k$  thì  $J(x_k)$  là tập các đỉnh thuộc  $C$  không tham gia vào  $x_k$ .

### 2.3. Thuật toán tổng quát

Giả sử với mỗi cạnh nối các đỉnh  $i, j \in C$  có trọng số heuristic  $h_{i,j}$  để định hướng chọn thành phần mở rộng là  $j$  khi thành phần cuối của  $x_k$  là  $i$  theo thủ tục nêu trên ( $h_{i,j} > 0 \forall (i, j)$ ). Dân kiến  $m$  con sẽ xây dựng lời giải trên đồ thị đầy có trọng số  $G = (V, E, H, \tau)$ , trong đó  $V$  là tập đỉnh tương ứng với tập thành phần  $C$  đã nêu ở trên (về sau ta vẫn dùng ký hiệu  $C$  thay cho  $V$  để chỉ tập đỉnh),  $E$  là tập các cạnh,  $H$  là vectơ các trọng số heuristic của cạnh tương ứng (trong bài toán TSP nó là vectơ mà thành phần là nghịch đảo độ dài của cạnh tương ứng) còn  $\tau$  vectơ vết mùi tích lũy được (ban đầu được khởi tạo bằng  $\tau_0 > 0$ ). Với số lần lặp  $N_c$  định trước, các thuật toán được mô tả hình thức như sau:

```

Procedure of ACO algorithms;
begin
  Initialize; // khởi tạo m con kiến
  Repeat
    Construct solutions; // mỗi con kiến xây dựng lời giải
    Update trail; // cập nhật mùi
  Until End condition; // điều kiện kết thúc
end;

```

#### 2.3.1. Xây dựng lời giải tuần tự

Sau khi khởi tạo các tham số dương  $\alpha, \tau_1$  (sẽ giải thích rõ dưới đây cho từng quy tắc cập nhật mùi) và lượng mùi ban đầu  $\tau_0$ , các con kiến thực hiện lặp thủ tục xây dựng lời giải tuần tự. Trong mỗi lần lặp  $t$ , mỗi con kiến  $s$  chọn ngẫu nhiên một đỉnh xuất phát trong  $C_0$  và kết hợp thông tin heuristic với thông tin mùi để xây dựng lời giải ngẫu nhiên theo thủ tục mở rộng tuần tự nêu trên với xác suất chọn đỉnh tiếp theo như sau. Quy tắc chuyển trạng thái. Giả sử kiến  $r$  đã xây dựng  $x_k = \langle u_0, \dots, i \rangle$  là mở rộng được, nó chọn đỉnh  $y$  thuộc  $J(x_k)$  để mở rộng thành  $x_{k+1} = \langle u_0, \dots, i, y \rangle$  nhờ chọn ngẫu nhiên với xác suất chọn  $y$  cho bởi công thức (7):

$$P(y/\tau, x_k) = \begin{cases} \frac{\tau_{i,y}^\alpha h_{i,y}}{\sum_{j \in J(x_k)} \tau_{i,j}^\alpha h_{i,j}} & , y \in J(x_k) \\ 0 & , y \notin J(x_k). \end{cases} \quad (7)$$

Quá trình mở rộng tiếp tục cho tới khi kiến  $r$  tìm được lời giải chấp nhận được  $x(r)$  trong  $X^*$  và do đó  $s(r) = \varphi(x(r)) \in S$ . Khi đó ta nói cạnh  $(i, y)$  thuộc  $x(r)$ .

Do giả thiết *iii*) trong mục 2.2, về sau ta sẽ xem  $x(r)$  và  $s(r)$  như nhau và không phân biệt  $X^*$  với  $S$  để tiện trình bày. Ký hiệu  $w(t)$  là lời giải tốt nhất các con kiến tìm được cho tới lần lặp này và  $w^i(t)$  là lời giải tốt nhất trong bước lặp, nếu  $w^i(t)$  không tốt hơn  $w(t-1)$  ta sẽ có  $w(t) = w(t-1)$ . Ta sẽ quan tâm tới các lời giải gần đúng  $w(t)$  này.

### 2.3.2. Cập nhật mùi

Có nhiều quy tắc cập nhật mùi, ở đây ta xét hai quy tắc điển hình và được sử dụng phổ biến nhất hiện nay xuất phát từ ACS và MMAS. Giả sử  $g$  là một hàm giá trị thực xác định trên  $S$  sao cho  $0 < g(s) < \infty \forall s \in S$  và  $g(s) > g(s')$  nếu  $f(s) < f(s')$  (trong bài toán TSP  $g(s)$  là nghịch đảo độ dài đường đi tương ứng), khi đó ở mỗi bước lặp cường độ vết mùi sẽ thay đổi theo một trong các quy tắc sau đây.

*Quy tắc ACS:* Quy tắc này phỏng theo ACS, bao gồm cả cập nhật địa phương và toàn cục.  
 + *Cập nhật mùi địa phương.* Nếu con kiến  $h$  thăm cạnh  $(i, j)$ , tức là  $(i, j) \in s(h)$  thì cạnh này sẽ thay đổi mùi theo công thức

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \rho\tau_1. \tag{8}$$

+ *Cập nhật mùi toàn cục.* Cập nhật mùi toàn cục chỉ cho các cạnh thuộc  $w(t)$ :

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \rho g(w(t)), \forall (i, j) \in w(t). \tag{9}$$

*Quy tắc MMAS.* Quy tắc này thực hiện theo MMAS. Sau khi mỗi con kiến đều xây dựng xong lời giải ở mỗi bước lặp, vết mùi được thay đổi theo công thức

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j}, \tag{10}$$

trong đó,

$$\Delta\tau_{i,j} = \begin{cases} \rho g(w(t)) & (i, j) \in w(t) \\ \max\{\tau_1 - (1 - \rho)\tau_{i,j}, 0\} & (i, j) \notin w(t), \end{cases} \tag{11}$$

ở đây  $\tau_1 > 0$  là tham số.

*Chú ý.*

1) So với MMAS trong mục 2.4.1, ở công thức (11) ta xem  $\tau_1 = \tau_{\min}$  và giả thiết với mọi  $t$  đều có  $\tau_{\max} \geq g(w(t))$  (giả thiết này chỉ để đơn giản cho trình bày).

2) Các quy tắc cập nhật mùi trên là quy tắc G-best, nếu trong các công thức (9) và (11) thay  $w(t)$  bởi  $w^i(t)$  thì ta nói là quy tắc i-best, trong mục sau ta chỉ xét cho quy tắc G-best.

## 3. PHÂN TÍCH TOÁN HỌC VỀ XU THẾ VẾT MÙI ACO

Phần này sẽ nghiên cứu tính hội tụ của các thuật toán ACS và MMAS, sau khi ước lượng xác suất tìm thấy một phương án ở bước lặp  $t$ , ta sẽ khảo sự thay đổi của vết mùi.

### 3.1. Ước lượng xác suất tìm thấy một phương án

**Mệnh đề 3.1.** *Các khẳng định sau đúng*

- a) *Bài toán tổng quát luôn có lời giải tối ưu.*
- b) *Với mỗi kết quả thực nghiệm, các giá trị  $f(w(t))$  luôn hội tụ khi  $t$  dần ra vô hạn.*

c) Ký hiệu tập lời giải của bài toán là  $S^*$  và giá trị tối ưu là  $g^*$  ( $g^* = f(s) : s \in S^*$ ) thì với mọi cạnh  $(i, j) \in E$  ta có đánh giá sau

$$0 < \tau_{\min} = \min\{\tau_0, \tau_1, g(w(1))\} \leq \tau_{i,j} \leq \max\{\tau_0, \tau_1, g(w(1))\} = \tau_{\max}. \quad (12)$$

*Chứng minh.* Khẳng định a) là hiển nhiên vì tập  $S$  (và  $X^*$ ) là tập hữu hạn. Khẳng định b) suy từ tính đơn điệu giảm của dãy  $f(w(t))$  và dãy này bị chặn bởi  $g^*$ . Khẳng định c) dễ dàng nhận được nhờ chứng minh quy nạp theo  $t$  với lưu ý rằng ở mỗi lần cập nhật mùi, cường độ vết mùi của các cạnh  $(i, j)$  theo quy tắc ACS có dạng:  $\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \rho\delta_{i,j}$ , trong đó  $\tau_{\min} \leq \tau_{i,j}$ ,  $\delta_{i,j} \leq \tau_{\max}$ , còn cường độ vết mùi các cạnh cập nhật theo MMAS suy từ các công thức (10) và (11). ■

Về sau ta sẽ giả thiết  $\tau_1 < g(w(t)) \forall t$  và như vậy  $\tau_{\max} = g^*$ .  
**Định nghĩa 3.1.** Với mọi  $i \in C$ , đại lượng  $k_*(i) = \min\{\frac{h_{i,j}}{h_{i,k}} : j, k \in C\}$  được gọi là hệ số lệch heuristic của đỉnh  $i$  còn đại lượng  $k_* = \min\{k_*(i) : i \in C\}$  được gọi là hệ số lệch heuristic của bài toán.

Với mọi  $s \in S$ , ta ký hiệu  $p_s(t)$  là xác suất để  $m$  con kiến tìm được  $s$  ở bước lặp  $t$ , mệnh đề sau cho ta một ước lượng cận dưới của nó.

**Định lý 3.1.** Với mọi  $s \in S$  và với mọi  $t$ , ta luôn có:

$$p_s(t) \geq p_{\min} > 0, \quad (13)$$

trong đó,  $p_{\min}$  xác định bởi công thức

$$p_{\min} = 1 - \exp\left(-\frac{mk_*^h \tau_{\min}^{h\alpha}}{n^h \tau_{\max}^{h\alpha}}\right). \quad (14)$$

*Chứng minh.* Giả sử  $s$  được xác định bởi  $x(\in X^*) = \langle u_0, \dots, u_k \rangle$  ( $k \leq h$ ) được mở rộng từ  $u_0$ . Với mỗi bước lặp  $t$  và  $\forall i < h$ , từ (9) và (14) ta có đánh giá sau đối với xác suất  $p(x_{i+1}/x_i)$  để  $x_{i+1} = \langle u_0, \dots, u_i, u_{i+1} \rangle$  được mở rộng từ  $x_i = \langle u_0, \dots, u_i \rangle$  bởi mỗi con kiến  $r$  là

$$p(x_{i+1}/x_i) \geq \frac{\tau_{\min}^\alpha h_{u_i u_{i+1}}}{\sum_{u \in J_r(u_i)} \tau_{\max}^\alpha h_{u_i u_{i+1}}} \geq \frac{k_* u_i \tau_{\min}^\alpha}{n \tau_{\max}^\alpha}. \quad (15)$$

Do đó, nếu gọi  $p(r, s)$  là xác suất để kiến  $r$  tìm được  $s$  ta có

$$p(r, s) = \prod_{i=1}^k p(x_i/x_{i-1}) \geq \frac{k_*^k \tau_{\min}^{k\alpha}}{n^k \tau_{\max}^{k\alpha}} \geq \frac{k_*^h \tau_{\min}^{h\alpha}}{n^h \tau_{\max}^{h\alpha}} > 0.$$

Vậy ta có ước lượng sau đối với xác suất trong lần lặp này cả  $m$  kiến không tìm thấy

$$p(\neg s) \leq \left(1 - \frac{k_*^h \tau_{\min}^{h\alpha}}{n^h \tau_{\max}^{h\alpha}}\right)^m \leq \exp\left(-\frac{mk_*^h \tau_{\min}^{h\alpha}}{n^h \tau_{\max}^{h\alpha}}\right).$$

Suy ra,  $p_s(t) \geq 1 - \exp\left(-\frac{mk_*^h \tau_{\min}^{h\alpha}}{n^h \tau_{\max}^{h\alpha}}\right) = p_{\min}$ , mệnh đề được chứng minh. ■

**Chú ý.** Nếu dùng quy tắc chuyển trạng thái cho bởi (7) thì dễ dàng nhận được

$$p_{\min} = (1 - q_0) \left[1 - \exp\left(-\frac{mk_*^h \tau_{\min}^{h\alpha}}{n^h \tau_{\max}^{h\alpha}}\right)\right].$$

Kí hiệu  $P(t)$  là xác suất tìm được lời giải trong  $t$  bước lặp (hay  $w(t) \in S^*$ ). Định lý sau là mở rộng của Định lý 4.1 trong [7], đảm bảo tính hội tụ cho các thuật toán.

**Định lý 3.2.** Với mọi  $\varepsilon > 0$  bé tùy ý, tồn tại  $T$  sao cho với mọi  $t > T$  ta đều có:  $P(t) > 1 - \varepsilon$ .

*Chứng minh.* Theo Mệnh đề 3.1, luôn tồn tại lời giải tối ưu  $s^* \in S^*$ . Từ Định lý 3.1, ta có ngay ước lượng

$$P(t) \geq 1 - \prod_{i=1}^t [1 - p_{s^*}(i)] = 1 - (1 - p_{\min})^t = 1 - \exp\left(-t \frac{mk_*^h \tau_{\min}^{h\alpha}}{n^h \tau_{\max}^{h\alpha}}\right). \quad (16)$$

Biểu thức (16) cho ta kết luận của định lý khi chọn  $T$  đủ lớn. ■

### 3.2. Đặc tính của vết mùi

Ta thấy rằng trong thực tế, ở các bước lặp  $t$  đủ lớn thì khả năng  $g(w(t)) > g(w(t + 1))$  (và do đó  $w(t + 1) \neq w(t)$ ) rất bé nên có thể từ bước lặp  $t_0$  có các cạnh  $(i, j)$  không bao giờ thuộc vào  $w(t)$ ,  $\forall t > t_0$  hoặc luôn thuộc vào nó. Ta sẽ khảo sát đặc điểm của  $\tau_{i,j}$  trong các trường hợp này.

**Định lý 3.3.** Giả sử cạnh  $(i, j)$  thuộc vào lời giải chấp nhận được  $s$  nào đó và tồn tại  $T$  sao cho  $(i, i) \in w(t)$ ,  $\forall t > T$  thì các khẳng định sau đúng

- a)  $\tau_{i,j}(t)$  hội tụ theo xác suất tới  $\tau_1$  nếu dùng quy tắc cập nhật mùi ACS.
- b)  $\tau_{i,j}(t) = \tau_1$  với mọi  $t > T + \frac{\ln \frac{\tau_1}{g^*}}{\ln(1-\rho)}$  nếu dùng quy tắc cập nhật mùi MMAS.

*Chứng minh.* a)  $\forall \varepsilon > 0$  ta cần chứng minh

$$\lim_{t \rightarrow \infty} P(|\tau_{i,j}(t) - \tau_1| > \varepsilon) = 0.$$

Công thức cập nhật mùi địa phương (8) có thể viết như sau:  $\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \rho\tau_1 = \tau_1 + (1 - \rho)(\tau_{i,j} - \tau_1)$ . Nếu từ lần lặp thứ  $T$  tới bước lặp  $t$  cạnh  $(i, j)$  được cập nhật địa phương  $k$  lần thì  $\tau_{i,j} = \tau_1 + (1 - \rho)^k [\tau_{i,j} - \tau_1]$ . Do đó tồn tại  $k_0$  đủ lớn để cho với mọi  $k > k_0$  ta có:  $|\tau_{i,j} - \tau_1| = |(1 - \rho)^k [\tau_{i,j} - \tau_1]| \leq \varepsilon$ . Ta sẽ chứng minh

$$\lim_{t \rightarrow \infty} P(k \leq k_0) = 0.$$

Thực vậy, nếu đặt  $p_0 = \frac{k_*^h \tau_{\min}^{h\alpha}}{n^h \tau_{\max}^{h\alpha}}$  thì ta có đánh giá sau đối với xác suất  $P(r; (i, j))$  con kiến  $r$  có cập nhật địa phương cạnh  $(i, j)$  ở mỗi bước lặp:  $P(r; (i, j)) \geq p_0 > 0$ . Nên xác suất  $P(r; \neg(i, j))$  con kiến  $r$  không cập nhật địa phương cạnh  $(i, j)$  ở mỗi bước lặp thoả mãn

$$P(r; \neg(i, j)) \leq 1 - p_0 = q_0 < 1. \quad (17)$$

Giả sử  $t = T + q$  thì từ lần lặp thứ  $T$  tới  $t + q$  có thể xem như có  $mq$  con kiến cập nhật địa phương ngẫu nhiên cạnh  $(i, j)$  với xác suất không cập nhật thoả mãn (17) trong bất cứ điều kiện nào. Lập luận tương tự như phép thử Becnouilli ta có đánh giá

$$P(k \leq k_0) \leq \lim_{t \rightarrow \infty} \sum_{i=1}^{k_0} C_{mq}^i q_0^{mq-i} = 0. \quad (18)$$

b) Với  $t = T + q$  ta có

$$\tau_1 \leq \tau_{i,j}(t) = \max\{(1 - \rho)^q \tau_{i,j}(T), \tau_1\} \leq \max\{(1 - \rho)^q g^*, \tau_1\}, \forall q \geq \frac{\ln \frac{\tau_1}{g^*}}{\ln(1 - \rho)} \quad \blacksquare$$

**Định lý 3.4.** Giả sử cạnh  $(i, j) \in w(t), \forall t \geq T$  thì các khẳng định sau đúng

a) Nếu cập nhật mùi theo ACS thì

$$\lim_{t \rightarrow \infty} \tau_{i,j}(t) \geq \tau_1 + \rho \frac{g(w(T)) - \tau_1}{1 - (1 - \rho)^{m+1}}. \quad (19)$$

b) Nếu cập nhật mùi theo MMAS thì

$$\lim_{t \rightarrow \infty} \tau_{i,j}(t) \geq g(w(T)). \quad (20)$$

*Chứng minh.* a) Ký hiệu  $k_u$  là số lần cập nhật mùi địa phương của cạnh  $(i, j)$  trong bước lặp  $u$ , khi đó ta có:  $\tau_{i,j}(u+1) = (1 - \rho)^{k_u+1} \tau_{i,j}(u) + \sum_{v=1}^{k_u+1} (1 - \rho)^v \rho \tau_1 + (1 - \rho) \rho [g(w(T)) - \tau_1]$  (có một lần cập nhật toàn cục trong mỗi lần lặp). Do  $g(w(t))$  là hàm đơn điệu tăng theo  $t$  nên ta suy ra

$$\tau_{i,j}(t) \geq (1 - \rho)^{q_1} \tau_{i,j}(T) + \sum_{v=1}^{q_1} (1 - \rho)^v \rho \tau_1 + \sum_{v=1}^{t-T} (1 - \rho)^{h_v} \rho [g(w(T)) - \tau_1],$$

trong đó  $q_t = \sum_{u=T+1}^t (k_u + 1)$  và  $h_1 = 1; h_{v+1} = h_v + k_{t-v+2} + 1$ , với mọi  $v \geq 2$ .

Bởi vì  $k_u \leq m$  với mọi  $u$  nên  $t$

$$\tau_{i,j}(t) \geq (1 - \rho)^{q_1} \tau_{i,j}(T) + \sum_{v=1}^{q_1} (1 - \rho)^v \rho \tau_1 + \sum_{v=0}^{t-T-1} (1 - \rho)^{v(m+1)} \rho [g(w(T)) - \tau_1]. \quad (21)$$

Lấy giới hạn dưới khi  $t$  dần ra vô hạn biểu thức (20), lưu ý rằng  $\lim_{t \rightarrow \infty} q_t = \infty$  ta có biểu thức (19).

Khẳng định b) chứng minh tương tự nhờ lấy giới hạn biểu thức

$$\tau_{i,j}(t) \geq (1 - \rho)^{t-T} \tau_{i,j}(T) + \sum_{v=0}^{t-T-1} (1 - \rho)^v \rho g(w(T)). \quad (22)$$

**Chú ý.** Các công thức (21) và (22) trong chứng minh định lý này cho ta ước lượng cận dưới của  $g(w(t))$ . ■

## 4. THẢO LUẬN

### 4.1. Nhận xét chung về các thuật toán ACO

Nhờ kết hợp thông tin heuristic và thông tin học tăng cường nhờ mô phỏng hoạt động của đàn kiến, các thuật toán ACO có các ưu điểm nổi trội sau:

1) Việc tìm kiếm ngẫu nhiên dựa trên các thông tin heuristic làm cho phép tìm kiếm linh hoạt và mềm dẻo trên không gian rộng hơn phương pháp heuristic sẵn có, do đó cho ta lời giải tốt hơn và có thể tìm được lời giải tối ưu.

2) Sự kết hợp học tăng cường thông qua thông tin về cường độ vết mùi cho phép ta từng bước thu hẹp không gian tìm kiếm mà vẫn không loại bỏ các lời giải tốt, do đó nâng cao chất lượng thuật toán. Bây giờ ta có các nhận xét chung về đặc tính khai thác và khám phá của các thuật toán.



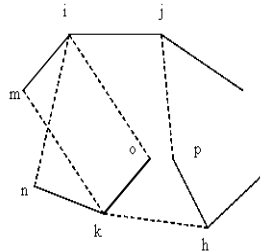
**4.2. Tính khai thác và khám phá**

Tính khai thác là việc tập trung tìm kiếm lời giải quanh phạm vi của các cạnh  $(i, j)$  thuộc các lời giải tốt nhất đã biết tới thời điểm đang xét còn tính khám phá là tìm kiếm ở các phạm vi khác. Trong cách cập nhật mùi G-best, ta đã biết  $w(t)$  nên việc tìm kiếm quanh nó sẽ hạn chế nhiều tính khám phá còn khi cập nhật theo i-best sẽ mở rộng miền này hơn. Vì vậy trong thực hành cập nhật theo i-best tốt hơn G-best.

Trong các bài toán tối ưu tổ hợp, thường thì xác suất để một phương án cho trước được các con kiến tìm được trong mỗi phép lặp rất bé. Vì vậy có thể sau một số bước lặp cường độ vết mùi trên mỗi thành phần của nó xác định theo các công thức (8 - 11) sẽ bé và giảm khả năng khám phá được chúng mặc dù chúng có thể vẫn rất hứa hẹn thuộc lời giải tốt. Chẳng hạn, với bài toán TSP ta có mệnh đề sau.

**Mệnh đề 4.1.** Trong bài toán TSP không định hướng, mỗi chu trình Haminton (đường liền) qua cạnh  $(i, j)$  và không qua cạnh  $(k, h)$  có thể đổi nhiều nhất 5 cạnh để có được chu trình đi qua cạnh  $(k, h)$  mà không qua  $(i, j)$ .

*Chứng minh.* Kết luận của mệnh đề dễ dàng nhận được từ hình dưới đây. Trong đó chu trình gồm các cạnh (đường liền) liền qua  $(i, j)$  và nhờ đổi đi 5 cạnh tương ứng  $\{(m, i), (n, k), (i, j), (k, o), (p, h)\}$  thành  $\{(m, k), (n, i), (i, o), (k, h), (p, j)\}$  (đường đứt đoạn) sẽ được chu trình đi qua cạnh  $(k, h)$  mà không qua  $(i, j)$ . ■



Hình 4.1. Hai chu trình khác nhau 5 cạnh, đường liền qua cạnh  $(i, j)$  và đường đứt đoạn qua cạnh  $(k, h)$ .

*Các điểm hạn chế của ACO*

1) Mệnh đề trên cho thấy khi thuật toán mới bắt đầu, các vết mùi khởi tạo như nhau thì một cạnh  $(k, h)$  “tốt hơn” cạnh  $(i, j)$  do nó thuộc chu trình dài hơn có thể đảo ngược một cách rất ngẫu nhiên. Khi một cạnh do ngẫu nhiên mà không được cập nhật mùi sau một số bước thì cường độ mùi của nó nhanh chóng bị giảm xuống và khó được các con kiến chọn sau đó mặc dù “chất lượng” của nó chưa chắc đã là “xấu”.

2) Nếu khởi tạo mùi như nhau và không dùng thông tin heuristic thì xác suất của mỗi cạnh được mỗi con kiến đã cho sử dụng trong lần lặp đầu là  $2/(n - 1)$ , xác suất này rất bé khi  $n$  lớn. Như vậy tùy theo từng loại bài toán mà tỷ lệ giữa  $\tau_0$  và  $\tau_1$  rất có ý nghĩa để cân bằng giữa tính khám phá và khai thác của thuật toán.

3) Các lượng mùi cập nhật theo các công thức (5.a), (5.b) hay (8)-(11) phụ thuộc vào giá trị hàm mục tiêu của lời giải mà các con kiến xây dựng được trong các bước lặp. Việc xác định các giá trị  $\tau_0$ ,  $\tau_1$  hay  $\tau_{\min}$ ,  $\tau_{\max}$  cũng phụ thuộc vào tương quan với các giá trị chưa được xác định trước này của từng bài toán thì thuật toán mới tốt được. Điều này rất khó thực hiện.

Bây giờ ta sẽ bình luận cụ thể hơn về từng thuật toán đã nêu.

### 4.3. Các thuật toán cập nhật mùi theo quy tắc ACS

Trong cách cập nhật mùi toàn cục của ACS theo [8] bởi công thức [4] và (5.b) không đảm bảo cường độ vết mùi thỏa mãn thuộc khoảng  $[\tau_{\min}, \tau_{\max}]$ , vết mùi của những cạnh không được các con kiến sử dụng và không thuộc đường đi tốt sẽ chóng dần về 0 và các con kiến sau sẽ có bỏ qua các cạnh này. Tuy vậy, như đã chỉ ra ở trên, cạnh này vẫn có thể là cạnh tham gia vào lời giải “tốt” mà bị loại do rủi ro. Hiện tượng này làm giảm tính khám phá của ACS, vì thế hiệu quả của nó kém MMAS [18].

Việc chọn quy tắc cập nhật mùi toàn cục trong [7] đã khắc phục được hạn chế này nên đến nay nó được thay cho cách cập nhật ở [8]. Tuy vậy việc cập nhật địa phương chưa cho thấy rõ ý nghĩa của học tăng cường.

### 4.4. Các thuật toán cập nhật mùi theo quy tắc MMAS

Theo quy tắc này việc tìm kiếm chỉ tập trung quanh lời giải tốt nhất còn các cạnh không thuộc lời giải này sẽ có cường độ vết mùi nhanh chóng tụt về  $\tau_{\min}$  theo đánh giá b) của Định lý 3. Vì vậy khi  $\tau_{\min}$  nhỏ hơn nhiều so với  $\tau_{\max}$  thì tính khám phá kém còn chọn  $\tau_{\min}$  gần với  $\tau_{\max}$  thì thuật toán chủ yếu là tìm kiếm ngẫu nhiên dựa theo thông tin heuristic.

### 4.5. Đề xuất các phương pháp cập nhật mùi mới

Dựa trên các phân tích trên, hai quy tắc cải tiến của ACS và MMAS được đề xuất như sau: a) *Phương pháp cập nhật mùi đa mức: MLAS*

Dựa vào nhận xét ở mục 4.3, thay cho việc bay hơi vết mùi ở các thành phần không thuộc các lời giải của mỗi con kiến trong  $m$  mỗi lần cập nhật mùi ở mỗi bước lặp ta cho  $\tau_1$  và  $\tau_{\max}$  tăng dần. Độ lệch giữa  $\tau_1$  và  $\tau_{\max}$  cho phép ta điều khiển tính hội tụ và khám phá. Nếu thấy lời giải tốt ít thay đổi thì cho  $\tau_1$  gần  $\tau_{\max}$  để tăng tính khám phá và ngược lại cho  $\tau_1$  dịch xa  $\tau_{\max}$  để cho lời giải tập trung tìm kiếm quanh lời giải tốt nhất tìm được.

b) *Phương pháp Max-Min trộn: SMMAS*

Dựa vào nhận xét ở mục 4.4, ta thấy không nên giảm vết mùi ở các cạnh không thuộc lời giải tốt quá nhanh như quy tắc MMAS mà nên dùng quy tắc Max-Min trộn như sau:

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j}, \text{ với } \Delta\tau_{i,j} = \begin{cases} \rho\tau_{\min} & , \forall(i,j) \in w(t) \\ \rho\tau_{\max} & , \forall(i,j) \notin w(t). \end{cases} \quad (23)$$

Trong hai phương pháp này, SMMAS đơn giản và dễ dùng hơn nhưng không mềm dẻo bằng MLAS.

Dễ dàng kiểm tra được các thuật toán này có cùng độ phức tạp nhưng MLAS có cùng độ phức tạp nhưng ít phép toán hơn ACS [7] vì không phải tính hàm mục tiêu ở lượng mùi cập nhật địa phương và toàn cục, còn SMMAS ít phép toán hơn MMAS vì không phải so vết mùi với  $\tau_{\min}$  và  $\tau_{\max}$ .

## 5. KẾT QUẢ THỰC NGHIỆM

Trong [4], kết quả thực nghiệm của một thuật toán SMMAS và biến thể khác của MLAS đối với bài toán lập lịch sản xuất cho thấy thuật toán với các quy tắc cập nhật mùi này tốt hơn ACS và MMAS. Dưới đây giới thiệu kết quả thực nghiệm so sánh ba phương pháp MMAS, SMMAS và MLAS với 5 bộ test eil51, kroA100, d198, lin318 và rat783 với số đỉnh tương ứng là 51, 100, 198, 318 và 783 đỉnh. Đây là các bộ test chuẩn của bài toán TSP được Dorigo và Stützle dùng để thực nghiệm và công bố [8,18] (các bộ test lấy từ địa chỉ [19]). Kết quả của MMAS trong Bảng 1 là kết quả chạy chương trình mới nhất của chính tác giả Stützle,

download từ địa chỉ [20], kết quả chạy theo chương trình này tốt hơn kết quả đã công bố trong các bài báo [8,18]. MMAS [18] khi công bố thì thực nghiệm cho thấy tốt hơn ACS, hơn nữa MMAS đơn giản hơn nên nhiều người ưa dùng.

Trong SMMAS và MLAS, các tham số được thiết đặt như sau:  $m = N/2$ ;  $\alpha = 1$ ;  $\beta = 2$ ;  $\rho = 0,05$ . Ngoài ra, trong SMMAS tỉ lệ  $\frac{\tau_{max}}{\tau_{min}}$  đặt bằng  $N.k$  với  $k = \begin{cases} \frac{N+50}{100} & N \geq 50 \\ 1 & N < 50 \end{cases}$ , việc

điều khiển  $\tau_{min}$ ,  $\tau_{max}$  trong MLAS thực hiện như trong [4], thực nghiệm với số lần chạy và số lời giải như Stützle đã làm [18]. Cụ thể, với các test eil51, kroA100 và d198, cả 3 phương pháp cho chạy 25 lần, mỗi lần cùng chạy với số lời giải  $S = 10000 \times N$  ( $N$  là số đỉnh), hai test lin318 và rat783 chạy 10 lần, mỗi lần cùng chạy với số lời giải  $S = 10^6$ , sau đó so sánh kết quả trung bình và kết quả tốt nhất của các lần chạy.

Bảng 1. Kết quả thực nghiệm so sánh ba phương pháp MMAS, SMMAS và MLAS

Test	MMAS		SMMAS		MLAS	
eil51	426.44(0.10%)		426.08(0.02%)		426.00(0.00%)	
	426	428	426	427	426	426
kroA100	21304.40(0.11%)		21305.76(0.11%)		21289.70(0.04%)	
	21282	21378	21282	21379	21282	21319
d198	15950.96(1.08%)		15990.44(1.33%)		15970.60 (1.21%)	
	15875	16034	15828	16116	15902	16040
lin318	43106.30(2.56%)		42445.80(0.99%)		42438.10(0.97%)	
	42859	43376	42201	42873	42257	42639
rat783	8951.5(1.65%)		8938.80(1.51%)		8933.00(1.44%)	
	8920	8986	8868	8989	8904	8959

Kết quả của các phương pháp với từng bộ test nằm trong ô giao giữa cột và dòng tương ứng. Trong đó số ở trên biểu thị kết quả trung bình và số trong ngoặc biểu thị độ lệch của kết quả trung bình với kết quả tối ưu của test đó, hai số ở dưới lần lượt là kết quả tốt nhất và kết quả tồi nhất trong các lần chạy. Kết quả trung bình phản ánh chất lượng của thuật toán còn các kết quả tốt nhất và tồi nhất để tham khảo về tính khám phá của nó. Các kết quả được tô đậm là kết quả tốt nhất trong 3 phương pháp.

**Nhận xét:** 1) *So sánh kết quả trung bình.* Trong 5 test được thực nghiệm có 4 test MLAS cho kết quả trung bình tốt nhất và SMMAS tốt hơn MMAS. Chỉ duy nhất test d198 là MMAS có kết quả trung bình tốt hơn SMMAS và MLAS nhưng không đáng kể.

2) *Kết quả tốt nhất.* Trong 5 test được thực nghiệm, có 2 test với số đỉnh nhỏ (eil51 - 51 đỉnh, kroA100 - 100 đỉnh) cả ba thuật toán đều tìm được lời giải tối ưu, 3 test sau thì SMMAS cho kết quả tốt hơn MMAS và MLAS.

## 6. KẾT LUẬN

Ta khảo sát tính biến thiên của vết mùi trong các thuật toán ACO điển hình, dựa trên các kết quả đạt được, đề xuất một số hướng cải tiến thuật toán mà hiệu quả của chúng đã được thử nghiệm cho một số bài toán cụ thể. Thực nghiệm cho thấy chúng cho lời giải tốt hơn ACS và MMAS. Tuy vậy các thuật toán ACO vẫn là thuật toán thực nghiệm nên việc lựa chọn và cài đặt thuật toán phải dựa vào từng bài toán cụ thể.

## TÀI LIỆU THAM KHẢO

- [1] C. Blum, ACO applied to group shop scheduling, A case study on intensification and diversification, *Proc. of ANTS 2002, Third International Workshop on ant algorithms*, Brussels, Belgium, September 12-14, 2002. (LNCS, vol. 2463).
- [2] B. Bullnheimer, R.F. Hartl, C. Strauss, A new rank based version of the ant system - a computational study, *Central European J. Oper. Res. Econom.* **7** (1999) 25–38.
- [3] Dinh Quang Huy, Do Duc Dong, Hoang Xuan Huan, Multi-level ant system : a new approach through the new pheromone update for ant colony optimization, *Proc. of The IEEE RIFV06*, HCM city, Vietnam, February 12-16, 2006 (55–58).
- [4] Do Duc Dong, Dinh Quang Huy, Hoang Xuan Huan, On the pheromone update rules of ant colony optimization approaches for the job shop scheduling problem, *Proc. OfPRIMA 2008*, Hanoi, Vietnam, December 15-16, 2008 (153–160).
- [5] B. Doerr, F. Neumann, D. Sudholdt, and C. Witt, On the influence of pheromone updates in ACO algorithms, “Technical Report CI-223/07”, University of Dortmund, SFB 531, 2007.
- [6] M. Dorigo, “Optimization, learning and natural algorithms”, PhD. dissertation, Milan Polytechnique, Italy, 1992.
- [7] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, 2004.
- [8] M. Dorigo and L.M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Trans. on Evolutionary Computation* **1** (1) (1997) 53–66.
- [9] W.J. Gutjahr, An ant based system and its convergence, future generation, *Comput. Systems* **16** (2000) 873–888.
- [10] W.J. Gutjahr, ACO algorithms with guaranteed convergence to the optimal solution, *Info.Proc. Lett.* **83** (3) (2002) 145–153.
- [11] W. J. Gutjahr, Mathematical runtime analysis of ACO algorithms: survey on an emerging issue, *Swarm Intelligence* **1** (1) (2007) 59–79.
- [12] Hoang Xuan Huan and Dinh Trung Hoang, On the ant colony system for postman problem, *Journal of Science*, Vietnam National University, Hanoi, **XVIII** (1) (2002) 29–36.
- [13] Hoàng Xuân Huấn và Đỗ Đức Đông, Về vết mùi trong các thuật toán ACO và khung cảnh mới, *Kỷ yếu hội thảo quốc gia một số chủ đề chọn lọc của công nghệ thông tin*, Đồng Nai, 2009 (534–547).
- [14] F. Neumann, D. Sudholt, and Carsten Witt, Rigorous analyses for the combination of ant colony optimization and local search, *Proceedings of the Sixth International Conference on Ant Colony Optimization and Swarm Intelligence*, Brussels, Belgium, September 22-24, 2008 (ANTS 2008, LNCS 5217, Springer, 2008) 132–143.
- [15] P. Pellegrini and A. Ellero, The small world of pheromone trails, *Proc. of the 6<sup>th</sup> International Conference on Ant Colony Optimization and Swarm Intelligence*, Brussels, Belgium, September 22-24, 2008 (387–394).
- [16] M. Sampels, J. Knowles, and K. Socha, A max-min ant system for the university timetabling problem, *Proc. of the 3<sup>rd</sup> International Workshop on Ant Algorithms (ANTS 2002)*, Brussels, Belgium, September 2002 (1–13).
- [17] T. Stützle and M. Dorigo, A short convergence proof for a class of ACO algorithms, *IEEE-EC* **6** (4) (2002) 358–365.
- [18] T. and H. H. Hoos, Max-min ant system, Future Gene, *Comput. Syst.* **26** (8) (2000) 889–914.
- [19] [www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/](http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/)
- [20] <http://iridia.ulb.ac.be/~mdorigo/ACO/aco-code/public-software.htm>

Ngày nhận bài 29 - 3 - 2011  
 Nhận lại sau sửa 2 - 6 - 2011