

# CẢI TIẾN VIỆC LẬP LỊCH LUỒNG CÔNG VIỆC TRONG MÔI TRƯỜNG ĐIỆN TOÁN Đám Mây DỰA TRÊN PHƯƠNG PHÁP PSO LÂN CẬN

Phan Thanh Toàn \* Nguyễn Thế Lộc\*

\* Khoa Sư phạm kỹ thuật, trường đại học Sư phạm Hà Nội

+ Khoa Công nghệ thông tin, trường đại học Sư phạm Hà Nội

**Tóm tắt:** Luồng công việc là một dãy có thứ tự các tác vụ cần phải thực thi để đạt được một mục đích, Bài toán lập lịch luồng công việc là bài toán sắp xếp các tác vụ cho thực thi trên một số máy xác định sao cho đạt hiệu quả tốt nhất, đây chính là bài toán quan trọng nhất tại các trung tâm điện toán đám mây. Trong bài báo này chúng tôi sẽ xây dựng một mô hình bài toán luồng công việc trong môi trường điện toán đám mây và đề xuất một thuật toán dựa trên phương pháp tối ưu bầy đàn cục bộ để sắp xếp luồng công việc thực thi trên môi trường điện toán đám mây đảm bảo thời gian hoàn thành luồng công việc nhỏ nhất.

**Keyword:** *Workflow scheduling, Particle Swarm Optimization, cloud computing, local search.*

## I. GIỚI THIỆU

Trong những năm gần đây điện toán đám mây đã được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau của cuộc sống và nghiên cứu khoa học. Trong môi trường điện toán đám mây mọi tài nguyên phần cứng, phần mềm đều được cung cấp cho khách hàng dưới dạng dịch vụ, khách hàng chỉ phải chi trả phí sử dụng theo tài nguyên thực dùng.

Luồng công việc (workflow) là một chuỗi có thứ tự các tác vụ (task) có thể được thực hiện đồng thời hay tuần tự nếu dữ liệu đầu ra của tác vụ này là đầu vào của tác vụ kế tiếp. Rất nhiều ứng dụng trong các lĩnh vực khoa học khác nhau đều yêu cầu phải xử lý một lượng lớn dữ liệu được tổ chức theo dạng luồng công việc. Vấn đề lập lịch luồng công việc trong môi trường điện toán đám mây về bản chất là tìm phương án ánh xạ những tác vụ của luồng công việc tới các máy chủ của đám mây sao cho thời gian xử lý toàn bộ luồng công việc là nhỏ nhất, biết rằng khối lượng tính toán và yêu cầu dữ liệu của các tác vụ, tốc độ tính toán và truyền thông của các máy chủ là khác nhau.

Bài toán lập lịch luồng công việc là một bài toán đã được nghiên cứu từ những năm 1950, và bài toán này đã được chứng minh thuộc lớp NP-Khó.

Trong những năm gần đây đã có rất nhiều ứng dụng khoa học được mô hình hóa bởi dạng đồ thị

luồng công việc như ứng dụng Montage [1], CyberShake [2], Epigenomics [3], LIGO [4], v.v.

Phần tiếp theo của bài báo có cấu trúc như sau. Phần II giới thiệu một số công trình nghiên cứu có liên quan về bài toán lập lịch luồng công việc. Trong phần III chúng tôi trình bày mô hình lý thuyết để biểu diễn năng lực tính toán và truyền thông của đám mây, dựa trên mô hình lý thuyết này, phần IV đề xuất:

- (i) phương thức mới để cập nhật vị trí của cá thể
- (ii) giải pháp để chương trình thoát ra khỏi vùng cực trị địa phương và di chuyển tới một vùng mới trong không gian tìm kiếm
- (iii) thuật toán lập lịch mới tên là LOSPSO

Phần V mô tả các thực nghiệm được tiến hành dựa trên công cụ mô phỏng Cloudsim [5] và phân tích những số liệu thực nghiệm thu được. Phần VI tóm tắt những kết quả chính của bài báo và hướng nghiên cứu sẽ tiến hành trong tương lai.

## II. NHỮNG CÔNG TRÌNH LIÊN QUAN

### 2.1. Những nghiên cứu về bài toán lập lịch

Bài toán lập lịch luồng công việc tổng quát đã được chứng minh là thuộc lớp NP-Khó [6] nghĩa là thời gian để tìm ra lời giải tối ưu tăng rất nhanh theo kích cỡ dữ liệu đầu vào, vì vậy đã có nhiều công trình nghiên cứu nhằm tìm ra lời giải đúng hoặc gần đúng của bài toán này.

N.S.Grigoreva [7] đã đề xuất thuật toán lập lịch điều phối các tác vụ của luồng công việc vào thực hiện trên một hệ thống đa bộ vi xử lý nhằm cực tiểu hóa thời gian hoàn thành luồng công việc. Tác giả đã sử dụng kết hợp phương pháp nhánh cận và kỹ thuật tìm kiếm nhị phân để tìm ra phương án xếp lịch có thời gian hoàn thành luồng công việc là nhỏ nhất.

R. Rajkumar [8] đã đề xuất thuật toán lập lịch luồng công việc dựa trên nhu cầu của khách hàng như

thời gian hoàn thành, chi phí thực thi,... qua đó sẽ điều phối các tác vụ vào thực hiện trên các máy chủ nhằm thỏa mãn tốt nhất nhu cầu của khách hàng.

Các tác giả trong bài báo [9] đã đề xuất thuật toán EGA (Enhanced Genetic Algorithm) lập lịch bằng phương pháp di truyền. Trong công trình các tác giả sử dụng thuật toán Enhanced Max Min trong bước khởi tạo quần thể nhằm tìm ra các cá thể tốt cho quá trình tiến hóa.

Pandey [10] đã đề xuất thuật toán lập lịch luồng công việc PSO Heuristic (Particle Swarm Optimization Heuristic – PSO\_H) trong môi trường điện toán đám mây dựa trên chiến lược tối ưu bầy đàn. Rajkumar đã đề xuất một thuật toán lập lịch phân cấp [10] và đưa vào các tham số dịch vụ khác nhau, chẳng hạn như thời gian đáp ứng. Thuật toán sử dụng tham số này như một quyền ưu tiên để lựa chọn các tác vụ lập lịch. Q.Cao và các đồng nghiệp đã trình bày thuật toán lập lịch dựa trên giải thuật ABC (Activity Based Costing) [11]. Thuật toán này gán mức ưu tiên cho mỗi tác vụ trong luồng công việc theo các tham số về thời gian, không gian, các tài nguyên và chi phí, quá trình lập lịch sẽ sử dụng mức ưu tiên này để quyết định các tác vụ được chọn trong quá trình lập lịch.

Selvi và các cộng sự đã đề xuất thuật toán lập lịch luồng công việc trong môi trường điện toán lưới (Grid) [12], trong công trình tác giả đã vận dụng thuật toán tiến hóa vi phân (DE) vào giải bài toán lập lịch luồng công việc trên môi trường điện toán lưới nhằm cực tiểu thời gian hoàn thành luồng công việc (makespan), trong công trình tác giả đã chỉ ra giá trị Makespan tìm được bởi thuật toán đề xuất là nhỏ hơn so với thuật toán PSO. Xu và các cộng sự đã đề xuất thuật toán COODE [13] (Current Optimum Opposition-based Differential Evolution) nhằm tìm giá trị tối ưu cho các hàm số dựa theo phương pháp tiến hóa vi phân đối xứng, trong công trình tác giả đã đề xuất công thức tìm điểm đối xứng của một điểm dựa theo giá trị tối ưu hiện tại nhằm thay đổi toán tử đột biến trong phương pháp tiến hóa vi phân và tác giả đã so sánh thuật toán COODE với các thuật toán DE và ODE, kết quả đã chỉ ra thuật toán đề xuất COODE tốt hơn các thuật toán đối sánh.

## 2.2. Phương pháp PSO lân cận

Trong phương pháp PSO chuẩn (PSO toàn cục) mỗi cá thể sẽ trao đổi thông tin với toàn bộ các cá thể khác trong quần thể, vector dịch chuyển của mỗi cá thể được cập nhật dựa trên thông tin tốt nhất của cá thể đó và thông tin tốt nhất của toàn bộ quần thể. Vector dịch chuyển và vector vị trí được cập nhật theo công thức sau:

$$v_i^{k+1} = \omega v_i^k + c_1 rand_1 \times (pbest_i - x_i^k) + c_2 rand_2 \times (gbest - x_i^k) \quad (3)$$

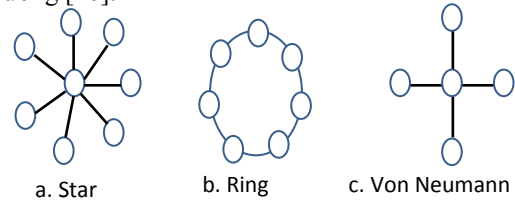
$$x_i^{k+1} = x_i^k + v_i^k \quad (4)$$

Tuy nhiên trong thực tế có nhiều kiến trúc khác nhau để biểu diễn mối quan hệ giữa các cá thể trong quần thể, một số công trình nghiên cứu như

[5],[14],[15] đã đề xuất các kiến trúc lân cận Star, Ring, Von Neuman. Thuật toán PSO cục bộ mỗi cá thể sẽ trao đổi thông tin với một số cá thể lân cận, số cá thể được trao đổi thông tin phụ thuộc vào mô hình kiến trúc lân cận sử dụng trong thuật toán. Công thức cập nhật vector vị trí của cá thể như sau :

$$v_i^{k+1} = \omega \times v_i^k + c_1 rand_1 \times (pbest_i - x_i^k) + c_2 rand_2 \times (lbest_i - x_i^k) \quad (5)$$

Phương pháp PSO toàn cục thường cho tốc độ hội tụ nhanh hơn phương pháp PSO cục bộ tuy nhiên chất lượng lời giải không tốt bằng phương pháp PSO cục bộ vì quần thể hay bị mắc kẹt tại các điểm cực trị địa phương [16].



Hình 1. Các kiến trúc lân cận

### Function Ring( $x_i$ )

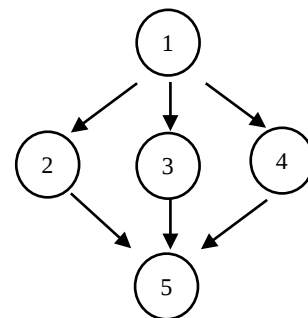
Input: current position  $x_i$   
Output:  $x$  where  $Fitness(x) = \min\{Fitness(x_i), Fitness(Left(x_i)), Fitness(Right(x_i))\}$

## III. MÔ HÌNH LÝ THUYẾT CỦA BÀI TOÁN

Đồ thị luồng công việc được biểu diễn bởi đồ thị có hướng, không có chu trình  $G=(V,E)$ .

**Trong đó:**

- $V$  là tập đỉnh, mỗi đỉnh tương ứng với một tác vụ trong đồ thị luồng công việc.
- $T=\{T_1, T_2, \dots, T_M\}$  là tập các tác vụ
- $E$  là tập cạnh, biểu diễn mối quan hệ giữa các tác vụ. Nếu  $e = (T_i, T_k)$  là một cạnh của đồ thị  $G$ , có nghĩa tác vụ  $T_i$  là tác vụ cha của tác vụ  $T_k$ , và tác vụ  $T_i$  sẽ gửi tới tác vụ  $T_k$  một khối lượng dữ liệu là  $tdata^k$ .
- $S = \{S_1, S_2, \dots, S_N\}$  là tập  $N$  máy chủ trong môi trường điện toán đám mây. Mỗi máy chủ được xác định bởi một năng lực tính toán xác định là  $P(S)$ , đơn vị đo là MI/s (million instructions/second).



Hình 2. Đồ thị biểu diễn một luồng công việc với 5 tác vụ.

- $M$

ỗi cặp máy chủ đều được kết nối với nhau bởi một đường truyền riêng, và có băng thông là  $B(S_i, S_j)$ . Băng thông được xác định bởi một hàm

$$B(): S \times S \rightarrow R^+ \\ B(S_i, S_j) = \infty \text{ và } B(S_i, S_i) = B(S_j, S_j)$$

#### Khái niệm lịch biểu

Mỗi lịch biểu được biểu diễn bởi một hàm  $f()$ :  $T \rightarrow S$ . Trong đó  $f(T_i)$  là máy chủ được giao để thực hiện tác vụ  $T_i$ .

- Thời gian tính toán của tác vụ  $T_i$  là:  $\frac{W_i}{P(f(T_i))}$  ( $i=1,2, \dots, M$ )
- Thời gian truyền dữ liệu giữa tác vụ  $T_i$  và tác vụ con  $T_j$  là  $\frac{D_{ij}}{B(f(T_i), f(T_j))}$

**Hàm mục tiêu:** Bài báo này định nghĩa hàm mục tiêu là:  $Makespan \rightarrow \min$  trong đó Makespan là thời gian hoàn thành luồng công việc, được tính từ khi tác vụ gốc được khởi động cho tới thời điểm tác vụ cuối cùng được thực hiện xong.

### IV. GIẢI PHÁP ĐỀ XUẤT

#### 4.1. Mã hóa cá thể

Theo phương pháp PSO, tại bước lặp thứ  $k$ , cá thể thứ  $i$  trong đàn được xác định bởi vector vị trí  $x_i^k$  (cho biết vị trí hiện tại) và vector dịch chuyển  $v_i^k$  (cho biết hướng dịch chuyển hiện tại). Trong bài toán xếp lịch đang xét, hai vector đó đều có số chiều bằng số tác vụ trong luồng công việc, ký hiệu là  $M$ . Cả vector vị trí và vector dịch chuyển đều được biểu diễn bằng cấu trúc dữ liệu mảng trong ngôn ngữ lập trình java.

**Ví dụ 1:** giả sử luồng công việc gồm tập tác vụ  $T=\{T_1, T_2, T_3, T_4, T_5\}$ , đảm bảo có tập máy chủ  $S=\{S_1, S_2, S_3\}$ . Khi đó cá thể  $x_i$  được biểu diễn bằng vector vị trí  $[1; 2; 1; 3; 2]$  chính là phương án xếp lịch mà theo đó tác vụ  $T_1, T_3$  được bố trí thực hiện bởi máy chủ  $S_1$ , tác vụ  $T_2, T_5$  được thực hiện trên  $S_2$  còn tác vụ  $T_4$  được thực hiện bởi  $S_3$  như dưới đây

|       |       |       |       |       |
|-------|-------|-------|-------|-------|
| $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
| $S_1$ | $S_2$ | $S_1$ | $S_3$ | $S_2$ |

#### 4.2. Phương pháp cập nhật vị trí cá thể

Khi áp dụng công thức cập nhật vị trí của cá thể (4) vào bài toán lập lịch đang xét, chúng ta gặp một vấn đề. Các thành phần của vector dịch chuyển  $v_i^k$  là số thực do công thức (5) tính vector dịch chuyển có những tham số là số thực như  $rand_1, rand_2, c_1, c_2$ . Nhưng vì tập máy chủ  $S$  là hữu hạn và đếm được nên các thành phần của vector vị trí  $x_i$  phải là số nguyên để có thể ánh xạ tới một máy chủ nào đó nơi mà tác vụ tương ứng sẽ được thực hiện, chẳng hạn vector vị trí  $x_i$  trong ví dụ 1 có các thành phần là  $x_i[1]=1, x_i[2]=2, x_i[3]=1, x_i[4]=3, x_i[5]=2$ . Hậu quả là hai vế của phép gán (2) khác kiểu nhau, vế trái  $x_i^{k+1}[t]$  thuộc kiểu số nguyên còn vế phải  $x_i^k[t] + v_i^k[t]$  thuộc kiểu số thực.

Để giải quyết mâu thuẫn này, một số nghiên cứu trước đây như [10] đã làm tròn giá trị số thực ở vế phải rồi gán cho biến vị trí  $x_i^{k+1}[t]$  ở vế trái. Kết quả là nếu giá trị của vế phải là 3.2 thì phân phối tác vụ tới thực thi tại máy chủ có số thứ tự là 3, còn nếu vế phải là 3.8 thì tác vụ sẽ được phân cho máy chủ có số thứ tự là 4. Cách làm có vẻ tự nhiên này thực chất là gán một vị trí được tính toán cẩn thận theo chiến lược PSO cho máy chủ mà số thứ tự của nó tình cờ đúng bằng giá trị nguyên sau khi làm tròn. Cách làm như vậy đã phá hỏng quá trình tiến hóa từng bước của phương pháp PSO.

Để giải quyết vấn đề trên, bài báo này đề xuất cách giải quyết như sau: giá trị thực của vế phải ( $x_i^k[t] + v_i^k[t]$ ) sẽ được để nguyên không làm tròn, còn vế trái  $x_i^{k+1}[t]$  sẽ được gán bởi định danh của máy chủ có tốc độ tính toán gần với giá trị của vế phải nhất so với các máy chủ còn lại. Làm như vậy tác vụ sẽ được gán cho máy chủ có năng lực phù hợp với giá trị được tính toán theo PSO.

$$x_i^{k+1}[t] \leftarrow j \text{ nếu } |P(S_j) - (x_i^k[t] + v_i^k[t])| \leq |P(S_r) - (x_i^k[t] + v_i^k[t])| \forall S_r \in S; t = 1, 2, \dots, M \quad (6)$$

**Ví dụ 2:** giả thiết tập máy chủ  $S$  trong ví dụ 1 có tốc độ tính toán được liệt kê trong Bảng 1 sau đây

Bảng 1. Tốc độ tính toán của các máy chủ

| Máy chủ $S_i$ | Tốc độ xử lý $P(S_i)$ (MI/s) |
|---------------|------------------------------|
| $S_1$         | 3.1                          |
| $S_2$         | 5.2                          |
| $S_3$         | 4.1                          |

Giả sử ở bước thứ  $k+1$  tổng  $x_i^k + v_i^k = [4.4; 2.1; 6.7; 5.6; 10.2]$  thì vector vị trí  $x_i^{k+1}$  sẽ được gán bằng  $[3; 1; 2; 2; 2]$ ; nghĩa là cá thể đó tương ứng với phương án xếp lịch sau đây:

|       |       |       |       |       |
|-------|-------|-------|-------|-------|
| $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
| $S_3$ | $S_1$ | $S_2$ | $S_2$ | $S_2$ |

Thật vậy, thành phần thứ nhất của vector vị trí,  $x_i^{k+1}[1]$ , sẽ nhận giá trị 3, nghĩa là tác vụ  $T_1$  sẽ được gán cho máy chủ  $S_3$  bởi vì:

$$x_i^{k+1}[1] \leftarrow 3 \text{ vì } |P(S_3) - 4.4| \leq |P(S_r) - (4.4)| \forall S_r \in S$$

Nghĩa là trong 3 máy chủ thì máy  $S_3$  có tốc độ tính toán gần với giá trị 4.4 nhất so với 2 máy chủ còn lại, theo bảng 1, do đó tác vụ  $T_1$  được gán cho máy chủ  $S_3$  để thực hiện, tức là  $f(T_1) = S_3$ . Phép gán tương tự cũng được thực hiện với bốn tác vụ còn lại:  $T_2, T_3, T_4, T_5$ .

Vấn đề tương tự cũng xảy ra với phép trừ hai vector vị trí trong công thức (1):  $(pbest_i - x_i^k)$  và  $(gbest - x_i^k)$ . Một số công trình hiện có như [10] chỉ đơn giản thực hiện phép trừ các thành phần số nguyên rồi gán cho máy chủ có số thứ tự tương ứng. Ví dụ nếu  $pbest_i = [2, 4, 3, 3, 5]$  và  $x_i^k = [1, 3, 2, 1, 2]$  thì  $pbest_i - x_i^k = [2-1, 4-3, 3-2, 3-1, 5-2] = [1, 1, 1, 2, 3]$ . Như đã giải thích ở trên, cách làm này thực chất là gán các tác vụ cho những máy chủ mà số thứ tự của nó tình cờ đúng bằng kết quả phép trừ. Cách làm mang tính ngẫu nhiên như vậy đã phá hỏng quá trình từng bước tiếp cận tới vị trí cực trị của phương pháp PSO. Bài báo

này đề xuất một "phép trừ vector" áp dụng riêng cho công thức (5) như sau. Giả sử:

$pbest_i = [x_{i1}, x_{i2}, \dots, x_{iM}]$  với  $x_{ik} \in S (\forall k)$  và  $x_j = [x_{j1}, x_{j2}, \dots, x_{jM}]$  với  $x_{jk} \in S (\forall k)$

Khi đó kết quả phép trừ  $pbest_i - x_j$  được tính như sau:  $pbest_i - x_j = [y_1, y_2, \dots, y_M]$  với các thành phần  $y_k$  là các số thực được tính như sau

$$y_k = \left\{ P(x_{ik}) + \frac{\sum_{q \in S} B(x_{ik}, x_{qk})}{N-1} \right\} - \left\{ P(x_{jk}) + \frac{\sum_{q \in S} B(x_{jk}, x_{qk})}{N-1} \right\} \text{ với } k = 1, 2, \dots, M$$

7)

Theo cách tính này, các máy chủ được xếp thứ tự theo tốc độ tính toán và băng thông của những đường truyền kết nối tới nó. Ví dụ 3 sau đây sẽ minh họa cụ thể hơn.

**Ví dụ 3:**

Ta tiếp tục sử dụng tập máy chủ trong ví dụ 2.

Giả sử  $lbest_j = [2, 1, 2, 1, 1]$ ;  $x_j = [3, 2, 1, 2, 1]$ ;

Vậy  $lbest_j - x_j = [y_1, y_2, y_3, y_4, y_5]$  với  $y_1$  được tính như sau

$$y_1 = \left\{ P(S_2) + \frac{B(S_2, S_1) + B(S_2, S_3)}{3-1} \right\} - \left\{ P(S_3) + \frac{B(S_3, S_1) + B(S_3, S_2)}{3-1} \right\}$$

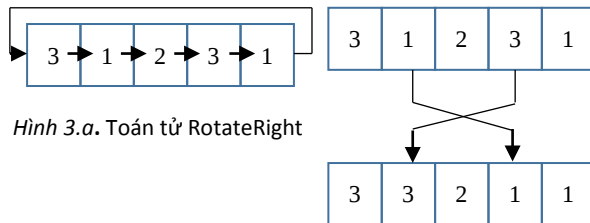
Cách tính tương tự được áp dụng cho các thành phần  $y_2, y_3 \dots y_5$  còn lại.

#### 4.3. Biện pháp thoát khỏi cực trị địa phương

Phương pháp PSO nói riêng và các phương pháp tìm kiếm tiến hóa nói chung đôi khi bị mắc kẹt tại các lời giải cực trị địa phương mà không thể thoát ra để đi tới lời giải tốt hơn. Bài báo này đề xuất sử dụng phương pháp PSO kết hợp với thủ tục tìm kiếm lân cận để định hướng cá thể tốt nhất chuyển sang vùng tìm kiếm mới mỗi khi chương trình bị sa vào vùng cực trị địa phương.

##### 4.3.1. Thủ tục tìm kiếm lân cận

Tìm kiếm lân cận [16] là phương pháp tìm kiếm bắt đầu từ một giải pháp ban đầu  $s_0$  của bài toán và sử dụng các toán tử để di chuyển sang một giải pháp khác của bài toán theo một cấu trúc lân cận xác định nhằm tìm ra một lời giải tốt hơn. Bài báo này đề xuất 2 toán tử Exchange và RotateRight sử dụng cho quá trình tìm kiếm lân cận (xem Hình 3.a và 3.b)



Hình 3.a. Toán tử RotateRight

Hình 3.b. Toán tử Exchange

##### 4.3.2. Giải thuật tìm kiếm lân cận

Function LocalSearch (vector vị trí  $x_i$ )

**Input:** vector vị trí  $x_i$

**Output:** vector vị trí  $x_k$  có  $f(x_k) < f(x_i)$

1. Khởi tạo bước lặp  $t \leftarrow 0$
2. while (điều kiện lặp)

3. Khởi tạo giá trị  $r$  ngẫu nhiên trong đoạn  $[1, M]$
  4.  $x_i \leftarrow \text{RotateRight}(x_i, r)$
  5. Khởi tạo 2 giá trị ngẫu nhiên  $\text{rand}_1, \text{rand}_2$  trong đoạn  $[1, M]$
  6.  $x_k \leftarrow \text{Exchange}(x_i, \text{rand}_1, \text{rand}_2)$
  7. if  $f(x_k) < f(x_i)$  then return  $x_k$
  5. else return  $x_i$
  6.  $t \leftarrow t+1$
  7. End while
- End Function

#### 4.4. Thuật toán đề xuất LOSPSO

Tổng hợp những cải tiến nói trên, thuật toán đề xuất với tên gọi LOSPSO được mô tả như sau.

##### Algorithm LOSPSO

**Input:** tập  $T$ , tập  $S$ , mảng  $W[1 \times M]$ , mảng  $P[1 \times N]$ , mảng  $B[N \times N]$ , mảng  $D[M \times M]$ , hằng số  $K$ , độ lệch  $\varepsilon$ , số cá thể  $SCT$

**Output:** lời giải tốt nhất  $gbest$

1. Khởi tạo vector vị trí và vector dịch chuyển của cá thể  $i$  một cách ngẫu nhiên
  2. Khởi tạo bước lặp  $t \leftarrow 0$ ;
  3. while (điều kiện lặp) do
  4. for  $i=1$  to  $SCT$  do
  5. Tính vector vị trí  $x_i$  theo công thức (6)
  6. end for
  7. for  $i=1$  to  $SCT$  do
  8. Cập nhật  $pbest_i$
  9. end for
  10. Cập nhật  $gbest$
  11. for  $(i=1$  to  $SCT$  do
  12.  $lbest_i := \text{Ring}(x_i)$ ;
  13. end for
  14. for  $i=1$  to  $SCT$  do
  15. Cập nhật vector dịch chuyển  $v_i^k$  theo công thức (5) và (6)
  16. Tính  $x_i$  theo (4)
  17. end for
  18.  $t++$ ;
  19. if (sau  $K$  thế hệ mà độ lệch giữa các  $gbest$  không vượt quá  $\varepsilon$ ) then
  20.  $gbest = \text{LocalSearch}(gbest)$ ;
  21. end if
  22. end while
- return  $gbest$

Thuật toán hoạt động theo phương pháp PSO theo đó tại mỗi bước lặp các cá thể cập nhật vị trí của mình hướng tới vị trí tốt nhất của các cá thể lân cận ( $lbest_i$ ) đồng thời có dựa trên kinh nghiệm cá nhân ( $pbest_i$ ). Nếu sau  $K$  thế hệ liên tiếp mà cá quần thể không cải thiện được một cách đáng kể giá trị  $gbest$  (mức chênh không vượt quá  $\varepsilon$ ) thì chúng tôi quần thể đang hội tụ tại một cực trị địa phương. Khi đó thủ tục LocalSearch được gọi tìm ra cá thể  $gbest$  mới và cá thể này sẽ di cư cá quần thể tới một vùng không gian mới, tại đó quá trình tìm kiếm được tái khởi động.

Độ phức tạp của thuật toán LOSPSO. Trước khi thực hiện thuật toán chính LOSPSO ta cần phải sắp xếp các máy chủ thực thi theo thứ tự tăng dần của tốc độ thực hiện, giải thuật sắp xếp có độ phức tạp về thời gian là  $O(n \cdot \log(n))$ , thủ tục tính ma trận thời gian thực thi của mỗi tác vụ trên các máy chủ có độ phức tạp thời gian tính toán là  $O(M \times N)$ ; với  $M$  là số tác vụ,  $N$  là số máy chủ. Thủ tục tính ma trận thời gian truyền dữ liệu giữa các máy chủ có độ phức tạp tính toán là  $O(N^2)$ . Trong thuật toán LOSPSO thì thủ tục khởi tạo sẽ khởi tạo các cá thể của quần thể một cách ngẫu nhiên, mỗi cá thể được mã hóa bởi một véc tơ độ dài  $M$  do vậy độ phức tạp của thủ tục khởi tạo là  $O(M \times SCT)$ ; với  $SCT$  là số cá thể trong quần thể, trong thực nghiệm chúng tôi sử dụng  $SCT$  là 100. Hàm tính thời gian thực hiện (makespan) của mỗi phương án xếp lịch là  $O(M^2)$ .

Thủ tục RotateRight có độ phức tạp là  $O(M)$ , thủ tục Exchange có độ phức tạp là hằng số, do vậy thủ tục tìm kiếm lân cận LocalSearch có độ phức tạp là  $O(M)$ .

Trong bài toán lập lịch luồng công việc thường số tác vụ lớn hơn số máy chủ ( $M > N$ ) do vậy độ phức tạp của thuật toán LOSPSO là  $(\text{Số thể hệ}) \times O(M^2)$ .

## V. KẾT QUẢ THỰC NGHIỆM

### 5.1. Phân nhóm dữ liệu thực nghiệm

Dữ liệu sử dụng trong các thực nghiệm bao gồm:

- Dữ liệu về tốc độ tính toán của các máy chủ và bảng thông giữa các máy chủ được lấy từ các công ty cung cấp dịch vụ cloud [17] và địa chỉ website (<http://aws.amazon.com/ec2/pricing>).
- Dữ liệu luồng công việc được lấy từ các bộ dữ liệu thử nghiệm được xây dựng theo độ trừ mật khác nhau và các luồng công việc từ các ứng dụng thực tế như ứng dụng Montage [18]
- Những dữ liệu đó được tổng hợp lại và chia thành hai nhóm, nhóm 1 là các luồng công việc ngẫu nhiên với sự khác nhau về hệ số  $\alpha$ , nhóm 2 là các luồng công việc từ ứng dụng Montage.

$$\alpha = \frac{|E|}{M \times (M - 1)/2}$$

- Tham số  $\alpha$  cho biết đồ thị  $G$  phân thành bao nhiêu cấp, mỗi cấp có nhiều hay ít tác vụ, nói cách khác  $\alpha$  phản ánh độ trừ mật của đồ thị  $G$ . Khi làm thực nghiệm với mỗi nhóm, số máy chủ và số tác vụ được giữ cố định còn tỷ lệ  $\alpha$  lần lượt thay đổi như trong các Hình 4,5,6.
- Tham số về máy chủ và số tác vụ của luồng công việc được xác định qua ký hiệu của bộ dữ liệu thực nghiệm, ví dụ T2086 sẽ có 20 tác vụ trong luồng công việc và thực hiện xếp lịch trên 8 máy chủ trong hệ thống điện toán đám mây. Trong quá trình thực nghiệm chúng tôi đã sử dụng các đồ thị luồng công việc với số tác vụ từ 5 đến 50 và hệ thống điện toán đám mây với số máy chủ từ 3 đến 8. Chi tiết về tham số cấu hình hệ thống được trình bày trong phần 5.2.

### 5.2. Tham số cấu hình hệ thống

Các tham số cấu hình của đám mây được thiết lập trong miền giá trị như sau:

- Tốc độ tính toán  $P$  của các máy chủ: từ 1 đến 250 (million instructions/s)
- Khối lượng dữ liệu  $D$  giữa các tác vụ: từ 1 đến 10000 (Mega bit)
- Bảng thông giữa các máy chủ  $B$ : từ 10 đến 100 (Mega bit/s)
- Hệ số quán tính:  $\omega = 0.729$
- Hệ số gia tốc:  $c_1 = c_2 = 1.49445$
- Hằng số:  $K = 30$
- Số cá thể  $SCT$ :  $SCT = 25$
- Độ lệch  $\varepsilon$ : 0.21
- $\alpha$ : từ 0.2 tới 0.7

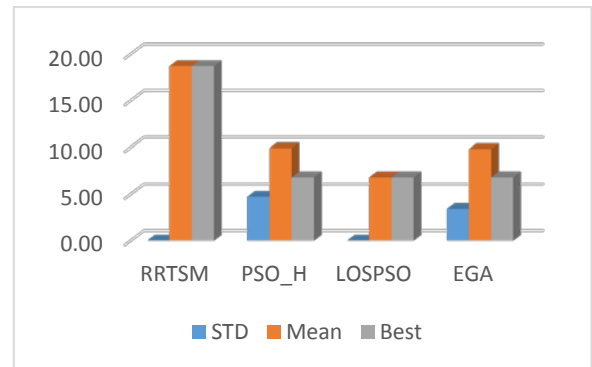
### 5.3. Quá trình tiến hành thực nghiệm

Để kiểm chứng thuật toán đề xuất LOSPSO chúng tôi đã sử dụng công cụ mô phỏng Cloudsim [5] để tạo lập môi trường đám mây kết hợp với dữ liệu luồng công việc của ứng dụng Montage [18]. Các hàm của gói thư viện Jswarm [5] được sử dụng để thực hiện các phương thức Tối ưu bầy đàn. Đối tượng so sánh là thuật toán PSO\_H [10], thuật toán EGA [9] và thuật toán Round Robin [19].

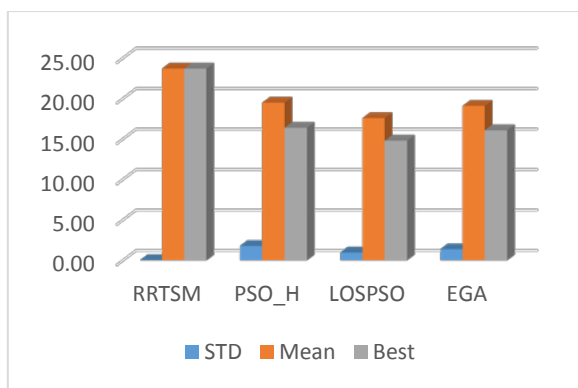
Các chương trình mô phỏng được viết bằng ngôn ngữ Java và chạy trên máy tính cá nhân với bộ vi xử lý Intel Core i5 2.2 GHz, RAM 4GB, hệ điều hành Windows 7 Ultimate. Thực nghiệm được tiến hành một cách độc lập 30 lần trên mỗi bộ dữ liệu thực nghiệm.

### 5.4. Kết quả thực nghiệm

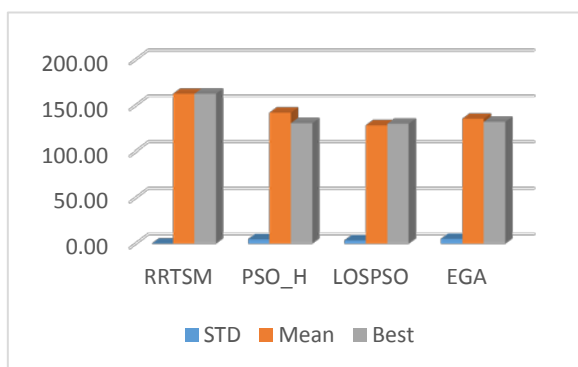
Hình 4,5,6 cho thấy sự chênh lệch về thời gian xử lý (makespan) của lời giải tốt nhất mà thuật toán đề xuất LOSPSO và các thuật toán đối chứng (PSO\_H, EGA và Round Robin) tìm được khi chạy trên các bộ dữ liệu khác nhau thuộc cả 2 nhóm luồng công việc ngẫu nhiên và luồng công việc từ ứng dụng Montage.



Hình 4: So sánh các thuật toán với bộ dữ liệu T532



Hình 5: So sánh các thuật toán với bộ dữ liệu T1051



Hình 6: So sánh các thuật toán với bộ dữ liệu M2032

Kết quả thực nghiệm được trình bày chi tiết trong bảng 2, 3 và các hình 4,5,6, kết quả so sánh giữa giá trị trung bình tính được bởi thuật toán LOSPSO với các thuật toán đối sánh, trong hầu hết các trường hợp thuật toán LOSPSO đều cho kết quả tốt hơn các thuật toán đối sánh, giá trị trung bình tìm được bởi LOSPSO nhỏ hơn giá trị trung bình tìm được bởi PSO\_H từ 4% - 11% và nhỏ hơn giá trị trung bình tìm được bởi thuật toán EGA từ 2% - 7%.

Hình 1,2,3,4 cũng so sánh giữa giá trị tốt nhất tìm được bởi thuật toán LOSPSO với các thuật toán đối sánh, qua đó ta thấy giá trị tốt nhất tìm được bởi LOSPSO nhỏ hơn giá trị tốt nhất tìm được bởi PSO\_H từ 2% - 9%, và nhỏ hơn giá trị tốt nhất tìm được bởi Random từ 20% - 40%, giá trị tốt nhất tìm được bởi thuật toán LOSPSO nhỏ hơn giá trị tốt nhất tìm được bởi thuật toán EGA từ 1% - 8%. Cuối cùng các hình 1,2,3,4 chỉ ra kết quả so sánh giữa độ lệch chuẩn tìm được bởi thuật toán LOSPSO với các thuật toán đối sánh, giá trị độ lệch chuẩn của LOSPSO đều nhỏ hơn độ lệch chuẩn của các thuật toán RRTSM, PSO\_H và EGA, điều đó chứng tỏ thuật toán LOSPSO có chất lượng lời giải tốt hơn các thuật toán đối sánh và độ ổn định trong các lần chạy cũng tốt hơn.

## VI. KẾT LUẬN

Bài báo này đã trình bày một giải thuật tìm kiếm theo phương pháp Tối ưu bầy đàn cục bộ (Local PSO) để tìm lời giải gần đúng cho bài toán lập lịch thực thi luồng công việc trong môi trường điện toán đám mây. Những kết quả chính gồm có:

- Đề xuất một phương thức mới để cập nhật vị trí của cá thể bằng cách ánh xạ một giá trị thực tới máy chủ có tốc độ tính toán và băng thông gần với giá trị đó nhất.
- Đề xuất công thức tính vector dịch chuyển của cá thể thứ  $i$  theo giá trị  $g_{best}$  và  $l_{best}$ .
- Đề xuất thủ tục LocalSearch để chương trình thoát ra khỏi cực trị địa phương bằng cách chuyển các cá thể tới một miền không gian tìm kiếm mới.
- Đề xuất thuật toán LOSPSO sử dụng phương thức cập nhật vị trí cá thể và thủ tục LocalSearch để tìm kiếm lời giải cho bài toán lập lịch thực thi luồng công việc trong môi trường đám mây.

Những kết quả thực nghiệm được tiến hành với nhiều bộ dữ liệu thực nghiệm khác nhau đã chứng tỏ chất lượng lời giải tìm được bởi thuật toán đề xuất tốt hơn so với các thuật toán đối chứng là thuật toán PSO\_H, thuật toán EGA và thuật toán Round Robin. Về hướng công việc tiếp theo chúng tôi dự định đề xuất một kiến trúc lân cận mới phù hợp với bài toán nhằm nâng cao khả năng tìm kiếm tổng thể qua đó nhằm đạt được lời giải có chất lượng tốt hơn.

## TÀI LIỆU THAM KHẢO

- [1] G. B. Berriman, et al, *Montage: A Grid Enabled Engine for Delivering Custom Science-Grade Mosaics On Demand*", in *SPIE Conference*, 2004.
- [2] P. Maechling, et al, *SCEC CyberShake Workflows, Automating Probabilistic Seismic Hazard Analysis Calculations*", Springer, 2006.
- [3] "USC Epigenome Center". <http://epigenome.usc.edu>. [Online]. <http://epigenome.usc.edu>
- [4] LIGO Project. LIGO - Laser Interferometer Gravitational Wave Observatory. [Online]. <http://www.ligo.caltech.edu>.
- [5] Buyya R., Ranjan R., Calheiros R.N. - Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities, *Proceedings of the Conference on High Performance Computing and Simulation (2009)* 1–11. (<http://www.cloudbus.org/cloudsim/>).
- [6] J.D. Ullman, *NP-complete scheduling problems*, *Journal of Computer and System Sciences*, pages 384-393, volume 10, issue 3, 1975
- [7] N.S.Grigoreva, *Branch and Bound Method for Scheduling Precedence Constrained Tasks on Parallel Identical Processors*, *Proceedings of the World Congress on Engineering*, Vol II, 2014
- [8] R. Rajkumar, T. Mala, *Achieving Service Level Agreement in Cloud Environment using Job Prioritization in Hierarchical Scheduling*, *Proceeding of International Conference on Information System Design and Intelligent Application*, 2012, vol 132, pp 547-554.
- [9] S. Singh, M.Kalra, *Task scheduling optimization of independent tasks in cloud computing using enhanced genetic algorithm*, *International Journal of Application or Innovation in Engineering & Management*, V.3, Issue 7, 2014.
- [10] S. Pandey, L. Wu1, S. M. Guru, R. Buyya1, *A Particle Swarm Optimization (PSO)-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments*, *Proc. of 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pages 400-407, 2010
- [11] Q. Cao, W. Gong and Z. Wei, *An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing*, In *Proceedings of*

Third International Conference on Bioinformatics and Biomedical Engineering, 2009, pp.1-3

- [12] S.Selvi, Dr. D.Manimegalai, Dr.A.Suruliandi, *Efficient Job Scheduling on Computational Grid with Differential Evolution Algorithm*, , International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April, 201
- [13] Q. XU, L.WANG, HE. Baomin, N.WANG, *Modified Opposition-Based Differential Evolution for Function Optimization*, Journal of Computational Information Systems, pp. 1582-1591, 2011.
- [14] A. E. M. Zavala, “*EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IIA Comparison, A Comparison Study of PSO Neighborhoods*”, Springer Verlag Berlin Heidelberg, pages 251-295, ISBN 978-3-642-32725-4, 2013.
- [15] H.Matsushita, *Swarm-Based Optimization with Dynamically-Changing Topology*, IEEE Workshop on Nonlinear Circuit Networks, pp. 63-66, December 2012.
- [16] P.Hansen, *Variable neighborhood search: Principles and applications*, European Journal of Operational Research 130, pp. 449-467, 2001.
- [17] Vliet J. V., Paganelli F. - Programming Amazon EC2, O'Reilly Media, ISBN 1449393683, 2011.
- [18] Bruce Berriman G., Deelman E. - Montage: A grid enabled engine for delivering custom science-grade mosaics on demand, in SPIE, 2004 (<http://montage.ipac.caltech.edu>)
- [19] M. Mitzenmacher, E. Upfal, “Probability and Computing: Randomized Algorithms and Probabilistic Analysis”, Cambridge University Press (2005).

**Abstract:** Cloud computing is a new trend of information and communication technology that enables resource distribution and sharing at a large scale. The Cloud consists of a collection of virtual machine that promise to provision on-demand computational and storage resources when needed. End-users can access these resources via the Internet and have to pay only for their usage. Scheduling of scientific workflow applications on the Cloud is a challenging problem that has been the focus of many researchers for many years. In this work, we propose a new algorithm for workflow scheduling that is derived from the Particle Swarm Optimization.

## SƠ LƯỢC VỀ TÁC GIẢ

### Phan Thanh Toàn

Sinh năm 1974 tại Thái Nguyên.

Tốt nghiệp ĐH và Thạc sĩ tại trường ĐH Bách khoa Hà Nội, nghiên cứu sinh năm 2012 tại Học viện Khoa học Công nghệ Quân sự. Hiện đang công tác tại trường ĐH Sư phạm Hà Nội.

Lĩnh vực nghiên cứu: phương pháp

tiến hóa, tối ưu, xử lý song song và phân tán.

Điện thoại : 0912.069.762

E-mail: [pttoan@hnue.edu.vn](mailto:pttoan@hnue.edu.vn)

### Nguyễn Thế Lộc

Tốt nghiệp ĐH khoa Toán Tin, ĐH Sư phạm Hà Nội năm 1993, Tốt nghiệp Thạc sĩ CNTT tại trường ĐH Bách khoa Hà Nội, nhận bằng tiến sỹ tại Viện Nghiên cứu Khoa học Công nghệ Nhật Bản JAIST năm 2007.

Lĩnh vực nghiên cứu: mạng máy tính và truyền thông, xử lý song

song và phân tán

Điện thoại : 0988.765.837

E-mail: [locnt@hnue.edu.vn](mailto:locnt@hnue.edu.vn).

Bảng 2. kết quả thực hiện thuật toán với các bộ dữ liệu ngẫu nhiên

| Kí hiệu | RRTSM |      |      | PSO_H |      |      | LOSPSO |      |      | EGA |      |      |
|---------|-------|------|------|-------|------|------|--------|------|------|-----|------|------|
|         | STD   | Mean | Best | STD   | Mean | Best | STD    | Mean | Best | STD | Mean | Best |
| T532    | -     | 18.7 | 18.7 | 4.7   | 9.9  | 6.8  | 0.0    | 6.8  | 6.8  | 3.4 | 9.8  | 6.8  |
| T1031   | -     | 33.1 | 33.1 | 2.4   | 20.4 | 16.4 | 1.1    | 18.3 | 16.2 | 1.2 | 20.4 | 16.4 |
| T1032   | -     | 16   | 16   | 2.0   | 8.2  | 3.5  | 1.2    | 5.1  | 3.5  | 1.8 | 8.1  | 3.5  |
| T1035   | -     | 18.9 | 18.9 | 2.5   | 9.6  | 4.2  | 1.8    | 6.7  | 2.9  | 2.3 | 9.5  | 4.9  |
| T1051   | -     | 23.7 | 23.7 | 1.8   | 19.5 | 16.4 | 1.0    | 17.6 | 14.8 | 1.4 | 19.1 | 16.1 |
| T1054   | -     | 25.2 | 25.2 | 2.0   | 20.7 | 17.9 | 0.9    | 19.0 | 17.5 | 1.3 | 20.5 | 17.5 |
| T2081   | -     | 72.7 | 72.7 | 5.2   | 44.2 | 35.0 | 2.7    | 37.8 | 32.3 | 3.7 | 41.9 | 35.0 |
| T2083   | -     | 20.3 | 20.3 | 2.2   | 21.2 | 18.8 | 0.5    | 19.4 | 18.4 | 1.4 | 20.3 | 19.1 |
| T2084   | -     | 72.0 | 72.0 | 6.1   | 44.7 | 37.1 | 2.7    | 39.3 | 34.8 | 3.2 | 42.5 | 37.5 |
| T2086   | -     | 32.5 | 32.5 | 1.5   | 26.2 | 23.8 | 1.2    | 21.4 | 17.8 | 1.4 | 25.4 | 22.5 |

Bảng 3. kết quả thực hiện thuật toán với các bộ dữ liệu từ ứng dụng Montage

| Kí hiệu | RRTSM |       |       | PSO_H |       |       | LOSPSO |       |       | EGA |       |       |
|---------|-------|-------|-------|-------|-------|-------|--------|-------|-------|-----|-------|-------|
|         | STD   | Mean  | Best  | STD   | Mean  | Best  | STD    | Mean  | Best  | STD | Mean  | Best  |
| M2032   | -     | 162.7 | 162.7 | 4.9   | 142.1 | 130.6 | 3.6    | 128.4 | 130.0 | 5.3 | 135.5 | 132.1 |
| M2051   | -     | 146.6 | 146.6 | 5.4   | 132   | 122.8 | 3.3    | 123.4 | 118.7 | 4.5 | 131.7 | 125.3 |
| M2531   | -     | 465.0 | 465.0 | 18.7  | 373.4 | 341.5 | 13.0   | 346.4 | 338.1 | 7.0 | 352.7 | 340.4 |
| M2532   | -     | 183.8 | 183.8 | 3.9   | 110.7 | 102.5 | 1.5    | 104.7 | 101.7 | 1.4 | 112.1 | 104.5 |
| M2533   | -     | 322.9 | 322.9 | 4.0   | 315.4 | 311.7 | 0.5    | 312.5 | 311.6 | 0.5 | 315.2 | 311.8 |
| M2581   | -     | 300.6 | 300.6 | 15    | 260.3 | 235.1 | 6.1    | 236.1 | 228.4 | 6.5 | 246.2 | 234.9 |
| M2582   | -     | 133.9 | 133.9 | 5.1   | 84.8  | 75.9  | 4.7    | 81.4  | 74.3  | 2.9 | 85.1  | 77.3  |
| M2583   | -     | 236.5 | 236.5 | 8.3   | 239   | 225   | 5.3    | 224.3 | 215.7 | 4.7 | 233.5 | 222.4 |
| M5081   | -     | 155.8 | 155.8 | 6.3   | 108.0 | 95.0  | 5.5    | 101.7 | 91.1  | 3.2 | 107.8 | 96.4  |
| M5082   | -     | 82.1  | 82.1  | 0.9   | 14.0  | 12.1  | 0.8    | 12.6  | 11.1  | 0.5 | 14.0  | 11.8  |
| M5083   | -     | 101.7 | 101.7 | 4.3   | 98.3  | 88.8  | 4.3    | 90.0  | 80.8  | 1.8 | 95.3  | 87.5  |

Bảng 2. kết quả thực hiện thuật toán với các bộ dữ liệu ngẫu nhiên

| Kí hiệu | RRTSM |      |      | PSO_H |      |      | LOSPSO |      |      | EGA |      |      |
|---------|-------|------|------|-------|------|------|--------|------|------|-----|------|------|
|         | STD   | Mean | Best | STD   | Mean | Best | STD    | Mean | Best | STD | Mean | Best |
| T532    | -     | 18.7 | 18.7 | 4.7   | 9.9  | 6.8  | 0.0    | 6.8  | 6.8  | 3.4 | 9.8  | 6.8  |
| T1031   | -     | 33.1 | 33.1 | 2.4   | 20.4 | 16.4 | 1.1    | 18.3 | 16.2 | 1.2 | 20.4 | 16.4 |
| T1032   | -     | 16   | 16   | 2.0   | 8.2  | 3.5  | 1.2    | 5.1  | 3.5  | 1.8 | 8.1  | 3.5  |
| T1035   | -     | 18.9 | 18.9 | 2.5   | 9.6  | 4.2  | 1.8    | 6.7  | 2.9  | 2.3 | 9.5  | 4.9  |
| T1051   | -     | 23.7 | 23.7 | 1.8   | 19.5 | 16.4 | 1.0    | 17.6 | 14.8 | 1.4 | 19.1 | 16.1 |
| T1054   | -     | 25.2 | 25.2 | 2.0   | 20.7 | 17.9 | 0.9    | 19.0 | 17.5 | 1.3 | 20.5 | 17.5 |
| T2081   | -     | 72.7 | 72.7 | 5.2   | 44.2 | 35.0 | 2.7    | 37.8 | 32.3 | 3.7 | 41.9 | 35.0 |
| T2083   | -     | 20.3 | 20.3 | 2.2   | 21.2 | 18.8 | 0.5    | 19.4 | 18.4 | 1.4 | 20.3 | 19.1 |
| T2084   | -     | 72.0 | 72.0 | 6.1   | 44.7 | 37.1 | 2.7    | 39.3 | 34.8 | 3.2 | 42.5 | 37.5 |
| T2086   | -     | 32.5 | 32.5 | 1.5   | 26.2 | 23.8 | 1.2    | 21.4 | 17.8 | 1.4 | 25.4 | 22.5 |

Bảng 3. kết quả thực hiện thuật toán với các bộ dữ liệu từ ứng dụng Montage

| Kí hiệu | RRTSM |       |       | PSO_H |       |       | LOSPSO |       |       | EGA |       |       |
|---------|-------|-------|-------|-------|-------|-------|--------|-------|-------|-----|-------|-------|
|         | STD   | Mean  | Best  | STD   | Mean  | Best  | STD    | Mean  | Best  | STD | Mean  | Best  |
| M2032   | -     | 162.7 | 162.7 | 4.9   | 142.1 | 130.6 | 3.6    | 128.4 | 130.0 | 5.3 | 135.5 | 132.1 |
| M2051   | -     | 146.6 | 146.6 | 5.4   | 132   | 122.8 | 3.3    | 123.4 | 118.7 | 4.5 | 131.7 | 125.3 |
| M2531   | -     | 465.0 | 465.0 | 18.7  | 373.4 | 341.5 | 13.0   | 346.4 | 338.1 | 7.0 | 352.7 | 340.4 |
| M2532   | -     | 183.8 | 183.8 | 3.9   | 110.7 | 102.5 | 1.5    | 104.7 | 101.7 | 1.4 | 112.1 | 104.5 |
| M2533   | -     | 322.9 | 322.9 | 4.0   | 315.4 | 311.7 | 0.5    | 312.5 | 311.6 | 0.5 | 315.2 | 311.8 |
| M2581   | -     | 300.6 | 300.6 | 15    | 260.3 | 235.1 | 6.1    | 236.1 | 228.4 | 6.5 | 246.2 | 234.9 |
| M2582   | -     | 133.9 | 133.9 | 5.1   | 84.8  | 75.9  | 4.7    | 81.4  | 74.3  | 2.9 | 85.1  | 77.3  |
| M2583   | -     | 236.5 | 236.5 | 8.3   | 239   | 225   | 5.3    | 224.3 | 215.7 | 4.7 | 233.5 | 222.4 |
| M5081   | -     | 155.8 | 155.8 | 6.3   | 108.0 | 95.0  | 5.5    | 101.7 | 91.1  | 3.2 | 107.8 | 96.4  |
| M5082   | -     | 82.1  | 82.1  | 0.9   | 14.0  | 12.1  | 0.8    | 12.6  | 11.1  | 0.5 | 14.0  | 11.8  |
| M5083   | -     | 101.7 | 101.7 | 4.3   | 98.3  | 88.8  | 4.3    | 90.0  | 80.8  | 1.8 | 95.3  | 87.5  |