

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO ĐỀ TÀI**  
**ỨNG DỤNG GIẢI THUẬT TIẾN HÓA TRONG BÀI**  
**TOÁN LẬP LỊCH FOG COMUTING**

**HỌC PHẦN: CÁC THUẬT TOÁN CƠ BẢN TRONG TÍNH TOÁN**  
**TIẾN HOÁ**

**Giảng viên hướng dẫn: PGS.TS Huỳnh Thị Thanh Bình**

**Nhóm: 1**

<b>Chu Quốc Quân</b>	<b>20143627</b>	<b>CNTT 2.2 K59</b>
<b>Bạch Việt Dũng</b>	<b>20140767</b>	<b>CNTT 2.2 K59</b>
<b>Nguyễn Việt Tiến</b>	<b>20144485</b>	<b>CNTT 2.1 K59</b>

**Hà Nội, ngày 15 tháng 5 năm 2018**

**MỤC LỤC**

<b>LỜI MỞ ĐẦU .....</b>	<b>3</b>
<b>PHÂN CÔNG NHIỆM VỤ CHO CÁC THÀNH VIÊN TRONG NHÓM .....</b>	<b>5</b>
<b>I. BÀI TOÁN LẬP LỊCH CÔNG VIỆC TRONG HỆ THỐNG ĐIỆN TOÁN SƯƠNG MÙ – ĐÁM MÂY .....</b>	<b>6</b>
1. Giới thiệu về Điện toán sương mù.....	6
2. Bài toán lập lịch công việc trong Điện toán đám mây - sương mù .....	7
3. Mô hình hóa bài toán.....	8
4. Mục tiêu bài toán .....	9
<b>II. CÁC GIẢI THUẬT ĐỀ XUẤT TRONG BÀI TOÁN LẬP LỊCH FOG COMPUTING .....</b>	<b>11</b>
1. Giải thuật Di truyền (Genetic Algorithm).....	11
2. Các giải thuật Tìm kiếm Cục Bộ (Local Search) .....	13
2.1. Giải thuật Leo Đồi (Hill Climbing) .....	13
2.2. Giải thuật Tabu (Tabu Search).....	14
2.3. Giải thuật Leo Đồi Cải Tiến – Chấp nhận hy sinh.....	15
2.4. Giải thuật Luyện Kim (Simulated Annealing).....	17
2.5. Giải thuật Tăng/Giảm Trần Tuyệt Tính (Degraded Ceiling).....	19
3. Giải thuật Kết hợp Di truyền với Tìm Kiếm Cục Bộ (Combination of GA and Local Search) .....	21
3.1. Giải thuật Mã Giả Kết hợp Di truyền với Tìm Kiếm Cục Bộ .....	21
<b>III. THỰC NGHIỆM ĐÁNH GIÁ CÁC GIẢI THUẬT .....</b>	<b>22</b>
1. Thực nghiệm – Giải thuật Leo Đồi.....	22
2. Thực nghiệm – Giải thuật Leo Đồi Cải Tiến.....	23
3. Thực nghiệm – Giải thuật Luyện Kim.....	24
4. Thực nghiệm – Giải thuật Tăng-Giảm Trần Tuyệt Tính .....	25
5. Thực nghiệm – Giải thuật Di truyền (GA) .....	25
6. Thực nghiệm – Giải thuật Kết hợp GA và Luyện Kim.....	26
7. Thực nghiệm – Giải thuật Kết hợp GA và Leo Đồi Cải Tiến.....	26

<b>8. Thực nghiệm – Giải thuật Kết hợp GA và Tăng giảm tần tuyến tính.....</b>	<b>27</b>
<b>9. Thực nghiệm – Giải thuật Kết hợp GA và Leo Đồi.....</b>	<b>28</b>
<b>Tài liệu tham khảo .....</b>	<b>29</b>

## **LỜI MỞ ĐẦU**

Fog Computing là một trong những hướng nghiên cứu và ứng dụng mới mẽ đang được rất nhiều nhà khoa học hàng đầu quan tâm trong vài năm trở lại đây. Sự ra đời của công nghệ Fog Computing dường như là một tính tất yếu cần thiết cho xu hướng phát triển của nền tảng Internet of Thing trong phát triển tiềm năng các hạ tầng quản lý thông minh từ xa, các hệ thống điều khiển tự động trong thời gian thực, và hàng ngàn những lĩnh vực, dịch vụ, ứng dụng cần thiết khác trong cuộc sống. Ý tưởng về Fog Computing đã được nảy nở từ các yếu điểm chính về mặt thời gian đang dần được bộc lộ tại các hệ thống Cloud Computing trong guồng gia tăng chóng mặt về dữ liệu, cũng như các thiết bị tham gia môi trường tương tác mạng Internet. Điện toán sương mù là một sự kết hợp cần thiết mà dung hòa được các yếu tố cốt lõi về mặt tốc độ xử lý, dung lượng bộ nhớ, cũng như khoảng cách trong không gian địa lý... từ 2 phía “rìa” trong môi trường mạng : Cloud Servers và End Devices.

Với tính cấp thiết cùng với tiềm năng ứng dụng cao công nghệ này vào trong thực tế, nhóm nghiên cứu đã tìm hiểu, phát biểu và mô hình hóa một cách cụ thể bài toán Fog Computing, trên cơ sở đó tiến hành đề xuất và phát triển các thuật toán lập lịch dựa trên ý tưởng chính trong Tính toán Tiến hóa và Tìm kiếm cục bộ. Thông qua quá trình thực nghiệm, nhóm đã bước đầu thu được những kết quả khá tích cực và khả quan từ các ý tưởng kết hợp này. Nhóm sẽ trình bày cụ thể cơ sở lý thuyết kèm theo thực nghiệm đánh giá thu được từ bài toán Fog Computing (nêu trên) trong khuôn khổ nội dung Báo Cáo môn học này.

Nội dung báo cáo được nhóm trình bày theo 3 chương chính :

Chương 1 : Tổng quan về Fog Computing và Phát biểu, mô hình hóa bài toán

Chương 2 : Trình bày và phân tích các giải thuật đề xuất cho bài toán

**Chương 3 : Thực nghiệm và Đánh giá chất lượng cho từng thuật toán nghiên cứu**

Nhóm xin gửi lời cảm ơn chân thành tới PGS.TS Huỳnh Thị Thanh Bình cùng nhóm nghiên cứu Fog Computing. Trong thời gian vừa qua, chúng em đã được hỏi học rất nhiều điều từ cô và các anh chị trong nhóm nghiên cứu, từ những kiến thức bổ ích trong học phần Tính toán Tiến hóa, tới các công cụ, phương pháp quản lý mã nguồn, hơn thế nữa là những kỹ năng làm việc nhóm hiệu quả mà chúng em đã tự rút ra được cho mình thông qua các buổi làm việc, trò chuyện với cô và nhóm nghiên cứu,...

Tài liệu báo cáo này là một sự tích hợp, ghi nhận chi tiết cụ thể quá trình nghiên cứu và kết quả đạt được của nhóm trong quá trình tìm hiểu và phát triển, do vậy, thật khó để tránh khỏi những sai sót về cú pháp, hay tính logic trong tài liệu. Chúng em rất mong nhận được sự góp ý, phản hồi từ phía cô để chúng em có thể cập nhật và hoàn thiện báo cáo ngày một tốt hơn.

Chúng em xin chân thành cảm ơn.

Nhóm sinh viên nghiên cứu

**PHÂN CÔNG NHIỆM VỤ CHO CÁC THÀNH VIÊN TRONG NHÓM**

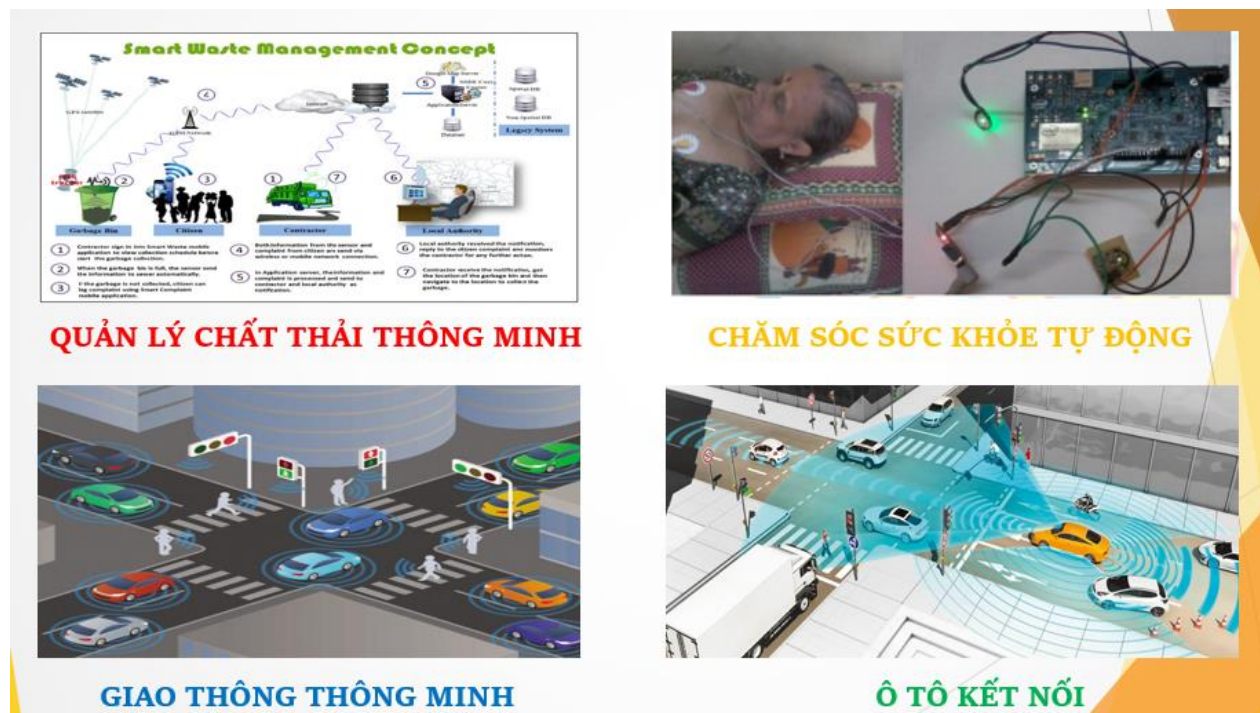
<b>HỌ TÊN</b>	<b>MSSV</b>	<b>NHIỆM VỤ ĐƯỢC GIAO</b>	<b>MỨC ĐỘ HOÀN THÀNH</b>
Chu Quốc Quân	20143627	<b>Nghiên cứu và Lập trình các giải thuật Tăng/Giảm Trần Tuyến Tính, Kết hợp GA-Leo Đồi Cải Tiến, GA-Tăng Giảm Trần</b>  <b>Thông kê thực nghiệm với các giải thuật tìm hiểu</b>  <b>Viết báo cáo và slide phần tìm hiểu</b>	Tốt
Bạch Việt Dũng	20140767	<b>Nghiên cứu và Lập trình các giải thuật Genetic Algorithm, Leo Đồi Cải Tiến, Luyện Kim, Kết hợp GA-Luyện Kim</b>  <b>Thông kê thực nghiệm với các giải thuật tìm hiểu</b>  <b>Viết báo cáo và slide phần tìm hiểu</b>	Tốt
Nguyễn Việt Tiến	20144485	<b>Nghiên cứu và Lập trình các giải thuật Leo Đồi, Tabu Search, Kết hợp GA – Giải thuật Leo Đồi, Kết hợp GA-Tabu Search</b>  <b>Thông kê thực nghiệm với các giải thuật tìm hiểu</b>  <b>Viết báo cáo và slide phần tìm hiểu</b>	Tốt

## **I. BÀI TOÁN LẬP LỊCH CÔNG VIỆC TRONG HỆ THỐNG ĐIỆN TOÁN SƯƠNG MÙ – Đám Mây**

### **1. Giới thiệu về Điện toán sương mù**

Internet of Things (IoT) là một xu hướng mới trong phạm trù công nghệ thông tin và truyền thông. Đó là một liên mạng mà ở đó các thiết bị xung quanh chúng ta đều được kết nối Internet. Các thiết bị thông minh như camera thông minh, các thiết bị đeo, cảm biến môi trường, các thiết bị nhà thông minh, phương tiện, ... được định danh duy nhất trong mạng và có khả năng tương tác, thu thập và trao đổi thông tin với nhau.

Số thiết bị được kết nối Internet hiện nay đã nhiều hơn số lượng người trên trái đất. Theo thống kê của công ty Information Handling Services (IHS) Markit, số thiết bị IoT được thiết lập năm 2015 là 15.4 tỉ thiết bị, dự đoán tương lai đến năm 2020 con số này là 30.7 tỉ và năm 2025 sẽ là 75.4 tỉ thiết bị





## **2. Bài toán lập lịch công việc trong Điện toán đám mây - sương mù**

Trong thực tế, nhiều ứng dụng offloading bao gồm nhiều mô-đun thực hiện các task khác nhau, các task có thể độc lập hoặc phụ thuộc lẫn nhau (8). Với các trường hợp trước, tất cả các task có thể được offload và xử lý một cách đồng thời, tuy nhiên trường hợp thứ hai thì ứng dụng bao gồm các task mà cần input từ các task khác nên việc offload song song không thể thực hiện được. Các ứng dụng trong trường hợp thứ hai thường được mô hình hóa như một luồng công việc biểu diễn bởi DAG, trong đó các đỉnh biểu diễn cho các task khác nhau còn các cạnh xác định các ràng buộc ưu tiên giữa chúng. Một ví dụ về ứng dụng dựa trên DAG là xử lý sự kiện phức tạp (CEP), một kỹ thuật phân tích dữ liệu lớn cho IoT. Các hệ thống dựa trên CEP biểu diễn dữ liệu đầu vào dưới dạng luồng sự kiện và thu nhận quy trình phân tích xác định người dùng bao gồm các truy vấn CEP (task) cần thực hiện trên các luồng này để có thể thu được sự kiện phức tạp về ngữ nghĩa từ một loạt sự kiện đơn giản (9,10). Với việc hệ thống cloud-fog computing ngày càng phổ biến, các ứng dụng dựa trên DAG như vậy sẽ được phân phối và nhúng vào trong môi trường với nhiều thiết bị máy tính kết nối với các tính năng không đồng nhất. Khi truyền dữ liệu từ điểm xuất phát của nó (ví dụ: sensor) sang nền cloud, nó đi qua nhiều thiết bị từ cạnh mạng tới cloud, mỗi điểm trong số đó là một mục tiêu tiềm năng để tính toán offloading. Thách thức chính nằm ở việc lập kế hoạch các task ứng dụng trong một nhóm các node xử lý trong môi trường cloud và fog, xem xét các mối liên quan giữa các mối phụ thuộc để tối ưu hóa một số mục tiêu được xác định trước. Trước đây, nhiều thuật toán lập lịch trình đã được đề xuất cho tính toán không đồng nhất, với mục tiêu chính là giảm thiểu thời gian thực hiện các nhiệm vụ, mà không phải lo lắng về những chi phí khi sử dụng các tài nguyên máy tính (11-14). Tuy nhiên, với sự xuất hiện của CC, trong đó một phần của việc thực hiện ứng dụng được thuê ngoài đến các tài nguyên máy tính của các nhà cung cấp khác nhau (CPs) và khách hàng đám mây (CCS) được tính dựa trên số máy ảo và giờ sử dụng, gần đây đã có một số nỗ lực được thực hiện để giảm chi phí sử dụng dịch vụ đám mây (15-22). Một kế hoạch công việc, có thể giảm thiểu thời gian hoàn thành công việc nhưng tương ứng với một lượng lớn chi phí tiền tệ, không phải là một giải pháp tối ưu cho CCS.

Bài báo xem xét việc lập kế hoạch công việc trong một nền tảng tính toán phân tán mới dựa trên sự giữa CC và FC để thực hiện các ứng dụng DAG dựa trên các ứng dụng giảm tải (offloading) quy mô lớn. Cụ thể, nền tảng tính toán có nguồn gốc từ cơ sở hạ tầng của các FN (ví dụ như bộ định tuyến, cổng, máy trạm, máy quay video IP) ở phía trước của CCS và được mở rộng bởi các máy ảo (các cloud node) cung cấp như các dịch vụ trong điện toán đám mây. Đề xuất một ứng dụng lập kế hoạch heuristic mới, thuật toán Cost-Makespan aware Scheduling (CMaS) để giải quyết vấn đề lập kế hoạch trong nền tảng này. Bên cạnh hiệu quả tổng thể, đề xuất của bài báo cũng tính đến điều kiện mạng trong



môi trường cloud-fog và chi phí tiền tệ tính cho CCs. Do đó, mục tiêu chính của thuật toán CMaS là đạt được sự cân bằng tốt giữa thời gian thực hiện ứng dụng và chi phí cho việc sử dụng tài nguyên cloud. Hơn nữa, chúng tôi trình bày một chiến lược phân bổ các task có trách nhiệm tinh chỉnh các lịch biểu đầu ra của thuật toán CMaS để đáp ứng những hạn chế do người dùng xác định, do đó cải thiện QoS của hệ thống. Cách tiếp cận trong bài báo được thực nghiệm đánh giá và so sánh với một số mô hình hiện có. Các kết quả chứng minh rằng phương pháp này có thể đảm bảo QoS và có hiệu quả chi phí tốt hơn các cách tiếp cận khác.

### 3. Mô hình hóa bài toán

Ở phần này, chúng ta sẽ mô hình hóa bài toán lập lịch công việc trong môi trường điện toán đám mây – sương mù.

Khi người dùng gửi các yêu cầu xử lý lên tầng sương mù, các yêu cầu này sẽ được phân tích thành các công việc nhỏ và không phụ thuộc vào nhau để được phân tán xử lý trên hạ tầng điện toán đám mây – sương mù, mỗi công việc đã được phân tích và có các thuộc tính: số câu lệnh cần thực hiện, dung lượng bộ nhớ yêu cầu, kích thước các tệp vào ra. Giả sử, tại một thời điểm, một tập gồm ‘ $n$ ’ công việc độc lập được gửi đến để được phân chia xử lý. Ta biểu diễn tập công việc đó như sau:

$$Tasks = \{T_1, T_2, T_3, \dots, T_n\}$$

Hạ tầng điện toán đám mây – sương mù bao gồm các máy xử lý: cloud node (máy chủ hoặc có thể là máy ảo trên tầng đám mây) và fog node (các thiết bị có khả năng xử lý ở tầng sương mù), các máy này có các thuộc tính như: tốc độ xử lý, chi phí sử dụng CPU, chi phí sử dụng bộ nhớ, chi phí sử dụng băng thông. Các cloud node thường có tốc độ xử lý mạnh mẽ hơn, tuy nhiên chi phí sử dụng sẽ lớn hơn so với các fog node. Giả sử, hạ tầng điện toán đám mây – sương mù có ‘ $m$ ’ máy tham gia xử lý, được biểu diễn như sau:

$$Nodes = \{N_1, N_2, N_3, \dots, N_m\}$$

Mỗi công việc  $T_k (T_k \in Tasks)$  sẽ được phân công xử lý trên một máy  $N_i (N_i \in Nodes)$  và được biểu diễn là  $T_k^i$ . Một máy có thể được phân công thực hiện một hay nhiều công việc. Giả sử máy  $N_i$  được phân công thực hiện một tập các công việc như sau:

$$N_i Tasks = \{T_x^i, T_y^i, \dots, T_z^i\}$$

Bài toán lập lịch công việc trên môi trường điện toán đám mây – sương mù được đưa về là tìm kiếm một tập như sau:

$$NodeTasks = \{T_1^a, T_2^b, T_3^c, \dots, T_n^p\}$$

#### 4. Mục tiêu bài toán

Bài toán lập lịch công việc trên môi trường điện toán đám mây – sương mù hướng tới nhiều mục tiêu nhằm lợi ích cho người dùng dịch vụ và nhà cung cấp dịch vụ. Lợi ích của người dùng dịch vụ tương ứng với thời gian thực hiện (makespan), ngân sách (budget), thời hạn (deadline), bảo mật (security), và chi phí (cost). Mặt khác, mục đích của nhà cung cấp dịch vụ là cân bằng tải (load balancing), tận dụng tài nguyên (resource utilization) và hiệu quả năng lượng (energy efficiency).

Mục tiêu của bài toán này hướng tới lợi ích cho người dùng dịch vụ khi tập trung tối ưu hóa thời gian thực hiện tất cả các công việc (makespan) và chi phí thực hiện (cost).

➤ **Thời gian thực hiện (makespan):**

- Thời gian thực hiện ( $EXT$ ) các công việc ( $r$  công việc) được phân công xử lý trên máy  $N_i$  là:

$$EXT(N_i) = \sum_{T_k^i \in N_i Tasks} T_k^i . ExeTime = \frac{\sum_{T_k^i \in N_i Tasks} T_k . length}{N_i . CPU\_rate}$$

Trong đó,  $T_k^i . ExeTime$  là thời gian máy  $N_i$  cần để hoàn thành công việc  $T_k$ , được xác định dựa trên số câu lệnh  $T_k . length$  của công việc và tốc độ xử lý  $N_i . CPU\_rate$  của máy  $N_i$ :

$$T_k^i . ExeTime = \frac{T_k . length}{N_i . CPU\_rate}$$

- *Makespan* là thời gian để hệ thống hoàn thành tất cả các công việc, được xác định từ thời điểm tập các công việc được gửi đến tới thời điểm công việc cuối cùng được hoàn thành, hay chính là thời điểm máy cuối cùng kết thúc xử lý. *Makespan* được xác định theo công thức:

$$Makespan = \underset{1 \leq i \leq m}{Max}\{EXT(N_i)\}$$

- *Min\_Makespan* là giá trị nhỏ nhất của *Makespan*, là thời gian nhỏ nhất để hệ thống hoàn thành tất cả công việc. Lí tưởng nhất là khi các công việc được phân chia các công việc phù hợp với khả năng xử lý của các máy để các máy có thể kết thúc tại cùng một thời điểm. *Min\_Makespan* được xác định như sau:

$$Min\_Makespan = EXT(N_1) = EXT(N_2) = \dots = EXT(N_n) = \frac{\sum_{T_k \in Tasks} T_k.length}{\sum_{N_i \in Nodes} N_i.CPU\_rate}$$

➤ **Chi phí thực hiện (cost):**

- Một công việc được thực thi trên môi trường điện toán đám mây – sương mù sẽ tồn các chi phí sau: chi phí xử lý (processing cost), chi phí sử dụng bộ nhớ (memory cost), và chi phí sử dụng băng thông (bandwidth cost). Cách xác định chi phí thực hiện của công việc  $T_k$  khi được thực hiện trên máy  $N_i$  được thể hiện qua công thức:

$$Cost(T_k^i) = c_{proc}(T_k^i) + c_{mem}(T_k^i) + c_{bw}(T_k^i)$$

Trong đó:

- $c_{proc}(T_k^i) = c_1 * T_k^i.ExeTime$  là chi phí xử lý hay chính là chi phí sử dụng CPU khi máy  $N_i$  thực hiện công việc  $T_k$  với  $T_k^i.ExeTime$  là thời gian xử lý được xác định theo công thức ( \* ),  $c_1$  là chi phí trên một đơn vị thời gian xử lý trên máy  $N_i$ .
- $c_{mem}(T_k^i) = c_2 * T_k^i.Mem$  là chi phí sử dụng bộ nhớ (RAM) khi máy  $N_i$  thực hiện công việc  $T_k$ , với  $T_k^i.Mem$  là dung lượng bộ nhớ công việc  $T_k^i$  yêu cầu khi nó được thực hiện,  $c_2$  là chi phí sử dụng bộ nhớ trên một đơn vị dữ liệu đối với máy  $N_i$ .
- $c_{bw}(T_k^i) = c_3 * T_k^i.Bw$  là chi phí sử dụng băng thông khi máy  $N_i$  thực hiện công việc  $T_k$ , với  $T_k^i.Bw$  là tổng băng thông sử dụng, được xác định bằng tổng dung lượng của file vào và file ra khi công việc  $T_k^i$  được thực hiện,  $c_3$  là chi phí truyền dữ liệu trên một đơn vị dữ liệu đối với máy  $N_i$ .
- Khi tất cả các công việc được thực hiện trên môi trường điện toán đám mây – sương mù, tổng chi phí thực hiện ( $TotalCost$ ) được xác định theo ():

$$TotalCost = \sum_{T_k^i \in NodeTasks} Cost(T_k^i)$$

- $Min\_TotalCost$  là tổng chi phí thực hiện nhỏ nhất có thể đạt được khi tập tất cả các công việc  $Tasks$  được xử lý trên môi trường điện toán đám mây – sương mù, tổng chi phí nhỏ nhất khi tất cả các công việc được thực hiện trên máy xử lý với chi phí nhỏ nhất đối với nó. Giả sử,  $Min\_Cost(T_k)$  là chi phí thực hiện nhỏ nhất của công việc  $T_k$ ,  $Min\_TotalCost$  được xác định theo công thức:

$$\text{Min\_TotalCost} = \sum_{T_k \in \text{Tasks}} \text{Min\_Cost}(T_k) = \sum_{T_k \in \text{Tasks}} \left[ \text{Min}_{1 \leq i \leq m} (\text{Cost}(T_k^i)) \right]$$

➤ **Hàm mục tiêu (Objective function):**

- Mục tiêu của bài toán là tập trung tối ưu hóa thời gian xử lý tất cả các công việc và chi phí thực hiện. Tác giả định nghĩa hàm mục tiêu (  $F$  ) tính toán sự cân bằng giữa thời gian (  $Makespan$  ) và chi phí thực hiện (  $TotalCost$  ) bằng công thức (\*):

$$F = \alpha * \frac{\text{Min\_Makespan}}{\text{Makespan}} + (1 - \alpha) * \frac{\text{Min\_TotalCost}}{\text{TotalCost}}$$

Trong đó,  $\alpha$  là hệ số cân bằng giữa thời gian và chi phí thực hiện.

## **II. CÁC GIẢI THUẬT ĐỀ XUẤT TRONG BÀI TOÁN LẬP LỊCH FOG COMPUTING**

### **1. Giải thuật Di truyền (Genetic Algorithm)**

#### **1.1. Giải thuật Mã Giả**

Khởi tạo quần thể với N cá thể xuất phát

Đánh giá và sắp xếp chất lượng các cá thể trong quần thể

Khởi tạo giá trị Sức ép chọn lọc PS (nếu có)

Khởi tạo Tỷ lệ số 1 trong Template ngẫu nhiên (Toán tử lai ghép) (nếu có)

Khởi tạo kích thước tập Offspring (con cháu) : M

Khởi tạo tỷ lệ Đột biến : R

Khởi tạo số lượng thế hệ tối đa (số vòng lặp – điều kiện dừng)

❖ Vòng lặp các thế hệ :

A. Toán tử chọn lọc - Chọn lọc tập Offsprings gồm M cá thể từ quần thể theo một trong các chiến lược :

- 1. Chọn ngẫu nhiên, cho phép trùng lặp M cá thể
- 2. Chọn ngẫu nhiên, không cho phép trùng lặp M cá thể
- 3. Chọn M cá thể có Fitness (Độ phù hợp) cao nhất
- 4. Chọn M cá thể theo sức ép chọn lọc PS

- B. Toán tử Lai ghép - Đưa tập Offspring gồm M cá thể trên qua một trong các chiến lược lai ghép sau :
- 1. Lai ghép 1 điểm cắt
  - 2. Lai ghép 2 điểm cắt
  - 3. Lai ghép dựa trên Template ngẫu nhiên chứa miền giá trị {0,1}
- C. Toán tử Đột biến - Lựa chọn ngẫu nhiên một số lượng cá thể từ tập Offsprings theo tỷ lệ đột biến R qua chiến lược Đột biến sau :
- 1. Chọn ngẫu nhiên một Gene trong cá thể cần đột biến, thay đổi giá trị bất kỳ (nằm trong miền giá trị) cho Gene đó.
- D. Toán tử Thay thế - Chọn lọc Quần thể cho thế hệ tiếp theo theo chiến lược
- 1. Hợp nhất Quần thể ban đầu với Tập các cá thể trong Offsprings. Đánh giá quần thể hợp nhất và lựa chọn N cá thể tốt nhất trong số đó làm quần thể mới cho thế hệ tiếp theo.

Kết thúc vòng lặp của 1 thế hệ

Hiển thị lời giải tốt nhất ứng với cá thể tốt nhất trong Quần thể sau cùng này.

**1.2. Phân tích Giải thuật Di truyền**

- Giải thuật di truyền thường hội tụ khá sớm (tại các vòng lặp 300-400)
- Cần sử dụng kích thước quần thể đủ lớn nhằm hạn chế sự đồng hóa sớm trong quần thể (trạng thái mà tại đó các cá thể trong quần thể đều giống hệt nhau)
- Nên chọn lựa giá trị Sức ép chọn lọc phù hợp (không quá lớn, hoặc quá nhỏ). Nếu sức ép chọn lọc lớn, khiến cho quần thể nhanh chóng bị đồng nhất, dễ lạc vào đỉnh cục bộ tìm kiếm gây ra hiện tượng hội tụ “non” (quá sớm trong khi có thể cải thiện được thêm về fitness). Nếu sức ép chọn lọc quá nhỏ, nghĩa là mọi cá thể trong quần thể đều có khả năng bị chọn lọc với xác suất như nhau, hay cá thể tồi cũng có xác suất được lựa chọn như cá thể tốt. Điều này gây ra hiện tượng “random walk” (định hướng một cách ngẫu nhiên- random), khiến cho giải thuật không cải thiện tốt được chất lượng của lời giải.

- Các chiến lược Lai ghép, Đột biến đóng vai trò quan trọng, giúp cải thiện được đáng kể chất lượng của giải thuật

## **2. Các giải thuật Tìm kiếm Cục Bộ (Local Search)**

### **2.1. Giải thuật Leo Đồi (Hill Climbing)**

#### **2.1.1. Giải thuật Mã Giả**

Khởi tạo từ một lời giải xuất phát ngẫu nhiên

Đánh giá Fitness (Độ phù hợp) của các thể

Khởi tạo danh sách chứa láng giềng. Láng giềng được định nghĩa theo 2 chiến lược : Gán giá trị cho một Gene (Assigning Neighbor) và Hoán đổi vị trí của 2 Gene (Swapping Neighbor)

Khởi tạo biến ghi nhận kỷ lục bestFitness

Khởi tạo Tổng số vòng lặp (Điều kiện dừng)

Vòng lặp :

- Kết nạp hàng xóm Assigning Neighbor vào Tập láng giềng tiềm năng theo chiến lược Leo đồi, trong đó với mỗi hàng xóm được kết nạp phải có Fitness lớn hơn (cải thiện hơn) so với Fitness hiện tại
- Kết nạp hàng xóm Swapping Neighbor vào Tập láng giềng tiềm năng theo chiến lược Leo đồi, trong đó với mỗi hàng xóm được kết nạp phải có Fitness lớn hơn (cải thiện hơn) so với Fitness hiện tại.

If (Tập láng giềng tiềm năng != Rỗng)

Lựa chọn ngẫu nhiên một cá thể trong Tập láng giềng tiềm năng. Cập nhật trạng thái cho cá thể hiện tại (theo láng giềng chọn được). Cập nhật Kỷ lục mới (nếu láng giềng mới tốt hơn kỷ lục hiện tại)

Else

Thực hiện cơ chế Restart – Tạo ngẫu nhiên một cá thể xuất phát mới để thực hiện tìm kiếm lại.

Kết thúc vòng lặp

Trả lại cá thể đang nắm giữ kỷ lục tốt nhất của bài toán

### **2.1.2. Phân tích Giải thuật Leo Đồi**

- Giải thuật Leo Đồi sử dụng chiến lược “đi theo hướng tốt” tới những cá thể láng giềng tốt hơn (cải thiện hơn) so với lời giải hiện tại.
- Giải thuật Leo Đồi thường hoạt động hiệu quả khi đã tìm được một lời giải xuất phát đủ tốt hay các lời giải đang nằm ở sườn đồi, điều này khiến cho giải thuật có thể cải thiện được chất lượng của lời giải lên tới đỉnh đồi. Ngược lại, giải thuật Leo Đồi sẽ hoạt động kém với các lời giải xuất phát chất lượng kém (tại các chân đồi), vì giải thuật Leo Đồi sẽ nhanh chóng bị lạc vào đỉnh cục bộ và sớm bị Restart (trạng thái không còn hàng xóm tiếp theo để đi)
- Thường áp dụng giải thuật Leo Đồi trong các pha cuối trong các chiến lược “đa pha” (gồm nhiều pha, chẳng hạn với chiến lược 3 pha : Pha 1 - Tìm một lời giải xuất phát đã triệt tiêu ràng buộc, Pha 2 – Cải thiện chất lượng lời giải với các giải thuật chấp nhận hy sinh hoặc các giải thuật di truyền hoặc bầy đàn, Pha 3 – Sử dụng Leo Đồi nhằm cải thiện sâu hơn chất lượng lời giải từ pha 2)

## **2.2. Giải thuật Tabu (Tabu Search)**

### **2.2.1. Giải thuật Mã Giả**

Khởi tạo từ một lời giải xuất phát ngẫu nhiên

Đánh giá Fitness (Độ phù hợp) của các thể

Khởi tạo danh sách chứa láng giềng. Láng giềng được định nghĩa theo 2 chiến lược : Gán giá trị cho một Gene (Assigning Neighbor) và Hoán đổi vị trí của 2 Gene (Swapping Neighbor)

Khởi tạo Danh sách Assigning Tabu ghi nhận các bước đi trong quá khứ của láng giềng Assigning Neighbor

Khởi tạo Danh sách Swapping Tabu ghi nhận các bước đi trong quá khứ của láng giềng Swapping Neighbor

Khởi tạo biến ghi nhận kỷ lục bestFitness

Khởi tạo Tổng số vòng lặp (Điều kiện dừng)

Vòng lặp :



- Kết nạp hàng xóm Assigning Neighbor vào Tập láng giềng tiềm năng theo chiến lược Leo đồi, trong đó với mỗi hàng xóm được kết nạp phải
  - Có Fitness lớn hơn (cải thiện hơn) so với Fitness hiện tại
  - Hoặc có Fitness bằng với Fitness hiện tại nhưng không chứa trong Assigning Tabu
- Kết nạp hàng xóm Swapping Neighbor vào Tập láng giềng tiềm năng theo chiến lược Leo đồi, trong đó với mỗi hàng xóm được kết nạp phải
  - Có Fitness lớn hơn (cải thiện hơn) so với Fitness hiện tại.
  - Hoặc có Fitness bằng với Fitness hiện tại nhưng không chứa trong Swapping Tabu

If (Tập láng giềng tiềm năng != Rỗng)

Lựa chọn ngẫu nhiên một cá thể trong Tập láng giềng tiềm năng. Cập nhật trạng thái cho cá thể hiện tại (theo láng giềng chọn được). Cập nhật Kỷ lục mới (nếu láng giềng mới tốt hơn kỷ lục hiện tại)

Else

Thực hiện cơ chế Restart cá thể mới (khởi tạo ngẫu nhiên cá thể mới) đồng thời Restart Assign Tabu và Swapping Tabu

Kết thúc vòng lặp

Trả lại cá thể đang nắm giữ kỷ lục tốt nhất của bài toán

### **2.2.2. Phân tích Giải thuật Tabu**

- Là một giải thuật Meta-Heuristic ra đời nhằm khắc phục tình trạng lạc vào miền cục bộ của các giải thuật Heuristic như Leo đồi,...
- Sử dụng một danh sách Tabu nhằm chứa các bước đi trong quá khứ, tránh tình trạng phải đi lại những nước đi đó mà không hề có sự cải thiện thêm nào
- Cần đề xuất chiến lược sinh láng giềng hiệu quả vì láng giềng càng tốt và đa dạng sẽ khiến cho giải thuật có tiềm năng “leo lên” được fitness cao nhất

## **2.3. Giải thuật Leo Đồi Cải Tiến – Chấp nhận hy sinh**

### **2.3.1. Giải thuật Mã Giả Leo Đồi Cải Tiến**

Khởi tạo từ một lời giải xuất phát ngẫu nhiên

Đánh giá Fitness (Độ phù hợp) của các thể

Khởi tạo danh sách chứa láng giềng. Láng giềng được định nghĩa theo 2 chiến lược : Gán giá trị cho một Gene (Assigning Neighbor) và Hoán đổi vị trí của 2 Gene (Swapping Neighbor)

Khởi tạo biến ghi nhận kỷ lục bestFitness

Khởi tạo mức độ hy sinh Sacrifice (cỡ  $10^{-3}$ )

Khởi tạo Tổng số vòng lặp (Điều kiện dừng)

Vòng lặp :

- Kết nạp hàng xóm Assigning Neighbor vào Tập láng giềng tiềm năng theo chiến lược mà trong đó với mỗi hàng xóm được kết nạp phải
  - Có Fitness lớn hơn (cải thiện hơn)  $> (\text{bestFitness} - \text{Sacrifice})$
- Kết nạp hàng xóm Swapping Neighbor vào Tập láng giềng tiềm năng theo chiến lược mà trong đó với mỗi hàng xóm được kết nạp phải
  - Có Fitness lớn hơn (cải thiện hơn)  $> (\text{bestFitness} - \text{Sacrifice})$

If (Tập láng giềng tiềm năng  $\neq$  Rỗng)

Lựa chọn ngẫu nhiên một cá thể trong Tập láng giềng tiềm năng. Cập nhật trạng thái cho cá thể hiện tại (theo láng giềng chọn được). Cập nhật Kỷ lục mới (nếu láng giềng mới tốt hơn kỷ lục hiện tại)

Else

Thực hiện cơ chế Restart cá thể mới (khởi tạo ngẫu nhiên cá thể mới).

Giảm mức độ Hy sinh sau mỗi 300 vòng lặp :  $\text{Sacrifice} = \text{Sacrifice} * 0.9$

Kết thúc vòng lặp

Trả lại cá thể đang nắm giữ kỷ lục tốt nhất của bài toán

### **2.3.2. Phân tích Giải thuật Leo Đồi Cải Tiến**

- Là một sự cải tiến nhỏ mà cần thiết cho Giải thuật Leo Đồi với ý tưởng chính : Tại mỗi bước kết nạp láng giềng, thì thay vì lựa chọn các láng giềng tốt hơn theo cơ chế của Leo Đồi, ta sẽ lựa chọn các láng giềng có giá trị fitness  $\geq$  (Kỷ Lục – Mức độ hy sinh). Yếu tố Mức độ Hy sinh cho thấy một sự “nhún nhường”, chấp nhận “xuống đồi” nhằm thoát khỏi các đỉnh cục bộ để tìm ra những hướng đi mới giàu tiềm năng hơn

## **2.4. Giải thuật Luyện Kim (Simulated Annealing)**

### **2.4.1. Giải thuật Mã Giả Luyện Kim**

Khởi tạo từ một lời giải xuất phát ngẫu nhiên

Đánh giá Fitness (Độ phù hợp) của các thể

Khởi tạo danh sách chứa láng giềng. Láng giềng được định nghĩa theo 2 chiến lược : Gán giá trị cho một Gene (Assigning Neighbor) và Hoán đổi vị trí của 2 Gene (Swapping Neighbor)

Khởi tạo danh sách chứa những láng giềng phá kỷ lục hiện tại

Khởi tạo biến ghi nhận kỷ lục bestFitness

Khởi tạo mức độ nhiệt độ ban đầu :  $T_0$

Khởi tạo nhiệt độ kết thúc :  $T_f$

Khởi tạo nhiệt độ hiện tại :  $currentFitness = T_0$

Khởi tạo Tổng số vòng lặp (Điều kiện dừng)

Tính mức độ giảm nhiệt độ theo từng vòng lặp :  $a = (\ln(T_0) - \ln(T_f)) / (\text{Tổng số vòng lặp})$

Vòng lặp <thỏa mãn điều kiện dừng> :

- Kết nạp hàng xóm Assigning Neighbor vào Tập láng giềng phá kỷ lục theo chiến lược mà trong đó với mỗi hàng xóm được kết nạp phải
  - Cải thiện hơn (phá kỷ lục) so với Kỷ lục hiện tại

- Kết nạp hàng xóm Assigning Neighbor vào Tập láng giềng tiềm năng theo chiến lược mà trong đó với mỗi hàng xóm được kết nạp phải

- Thỏa mãn công thức hy sinh :

$$\text{Math.random} \leq e^{\frac{-(\text{currentFitness} - \text{neighborFitness})}{\text{currentTemperature}}}$$

- Kết nạp hàng xóm Swapping Neighbor vào Tập láng giềng phá kỷ lục theo chiến lược mà trong đó với mỗi hàng xóm được kết nạp phải

- Cải thiện hơn (phá kỷ lục) so với Kỷ lục hiện tại

- Kết nạp hàng xóm Swapping Neighbor vào Tập láng giềng tiềm năng theo chiến lược mà trong đó với mỗi hàng xóm được kết nạp phải

- Thỏa mãn công thức hy sinh :

$$\text{Math.random} \leq e^{\frac{-(\text{currentFitness} - \text{neighborFitness})}{\text{currentTemperature}}}$$

If (Tập kỷ lục mới != Rỗng)

Lựa chọn ngẫu nhiên một cá thể trong Tập láng giềng Kỷ lục mới này. Cập nhật trạng thái cho cá thể hiện tại (theo kỷ lục mới chọn được). Cập nhật Kỷ lục mới

Else if (Tập láng giềng tiềm năng != Rỗng)

Lựa chọn ngẫu nhiên một cá thể trong Tập láng giềng tiềm năng. Cập nhật trạng thái cho cá thể hiện tại (theo láng giềng chọn được). Cập nhật Kỷ lục mới (nếu láng giềng mới tốt hơn kỷ lục hiện tại)

Else

Thực hiện cơ chế Restart cá thể mới (khởi tạo ngẫu nhiên cá thể mới).

Giảm nhiệt độ theo mỗi vòng lặp :  $\text{currentTemperature} *= a$

Kết thúc vòng lặp

Trả lại cá thể đang nắm giữ kỷ lục tốt nhất của bài toán

#### **2.4.2. Phân tích Giải thuật Luyện Kim**

- Có thể nói rằng, tính hiệu quả của giải thuật Luyện Kim tới từ những yếu tố sau:
  - Nhiệt độ ban đầu
  - Nhiệt độ kết thúc
  - Lịch trình làm đông lạnh (giảm nhiệt độ)
  - Chiến lược (công thức) chấp nhận hy sinh hiệu quả
- Qua quá trình nghiên cứu và cài đặt, nhóm nhận thấy rằng : đối với giải thuật Luyện Kim ta cần phải khảo sát để tìm ra được một dải nhiệt độ phù hợp nhất (giúp đánh giá đúng bản chất nhất về các hướng di chuyển tiềm năng), đồng thời cần phải đề xuất một chiến lược giảm nhiệt phù hợp với từng giai đoạn. Có những khoảng nhiệt độ giúp cải thiện rất tốt chất lượng của lời giải, nhưng có những lời giải thì không. Hay với mỗi giai đoạn khác nhau cần nên đề xuất chiến lược tăng-giảm tốc độ giảm nhiệt độ phù hợp nhất với bản thân giai đoạn đó.
- Nhóm nhận thấy đây là một giải thuật gây khó khăn nhiều nhất trong quá trình điều chỉnh tham số hay hiểu được rõ ràng bản chất và cơ chế làm việc thật sự của giải thuật

#### **2.5. Giải thuật Tăng/Giảm Trần Tuyến Tính (Degraded Ceiling)**

##### **2.5.1. Giải thuật Mã Giả :**

Khởi tạo từ một lời giải xuất phát ngẫu nhiên

Đánh giá Fitness (Độ phù hợp) của các thể : currentFitness

Khởi tạo danh sách chứa láng giềng. Láng giềng được định nghĩa theo 2 chiến lược : Gán giá trị cho một Gene (Assigning Neighbor) và Hoán đổi vị trí của 2 Gene (Swapping Neighbor)

Khởi tạo danh sách chứa những láng giềng phá kỷ lục hiện tại

Khởi tạo biến ghi nhận kỷ lục bestFitness

Thiết lập giá trị Fitness tốt nhất mong muốn : desiredFitness

Khởi tạo mức trần hiện tại bằng với Fitness của lời giải xuất phát :  
 $\text{currentCeiling} = \text{currentFitness}$

Khởi tạo Tổng số vòng lặp (Điều kiện dừng)

Tính mức độ tăng trần theo từng vòng lặp :  $a = (\text{desiredFitness} - \text{currentFitness}) / (\text{Tổng số vòng lặp})$

Vòng lặp <thỏa mãn điều kiện dừng> :

- Kết nạp hàng xóm Assigning Neighbor vào Tập láng giềng phá kỷ lục theo chiến lược mà trong đó với mỗi hàng xóm được kết nạp phải
  - Cải thiện hơn (phá kỷ lục) so với Kỷ lục hiện tại
- Kết nạp hàng xóm Assigning Neighbor vào Tập láng giềng tiềm năng theo chiến lược mà trong đó với mỗi hàng xóm được kết nạp phải
  - Thỏa mãn công thức hy sinh :  
 $\text{neighborFitness} \geq \text{currentCeiling}$
- Kết nạp hàng xóm Swapping Neighbor vào Tập láng giềng phá kỷ lục theo chiến lược mà trong đó với mỗi hàng xóm được kết nạp phải
  - Cải thiện hơn (phá kỷ lục) so với Kỷ lục hiện tại
- Kết nạp hàng xóm Swapping Neighbor vào Tập láng giềng tiềm năng theo chiến lược mà trong đó với mỗi hàng xóm được kết nạp phải
  - Thỏa mãn công thức hy sinh :  
 $\text{neighborFitness} \geq \text{currentCeiling}$

If (Tập kỷ lục mới != Rỗng)

Lựa chọn ngẫu nhiên một cá thể trong Tập láng giềng Kỷ lục mới này.  
Cập nhật trạng thái cho cá thể hiện tại (theo kỷ lục mới chọn được).  
Cập nhật Kỷ lục mới

Else if (Tập láng giềng tiềm năng != Rỗng)

Lựa chọn ngẫu nhiên một cá thể trong Tập láng giềng tiềm năng. Cập nhật trạng thái cho cá thể hiện tại (theo láng giềng chọn được). Cập nhật Kỷ lục mới (nếu láng giềng mới tốt hơn kỷ lục hiện tại)

Else

Thực hiện cơ chế Restart cá thể mới (khởi tạo ngẫu nhiên cá thể mới).

Tăng trần theo mỗi vòng lặp :  $\text{currentCeiling} += a$

Kết thúc vòng lặp

Trả lại cá thể đang nắm giữ kỷ lục tốt nhất của bài toán

### **2.5.2. Phân tích Giải thuật Tăng/Giảm Trần Tuyến Tính**

- Ý tưởng cốt lõi trong giải thuật Tăng/Giảm Trần Tuyến Tính đó là sử dụng một ngưỡng Trần (theo Fitness) tăng dần, đều đặn theo thời gian. Ban đầu, giá trị Trần được thiết lập bằng với Fitness của lời giải xuất phát. Trong quá trình tìm kiếm cục bộ, fitness của lời giải sẽ ngày một cải thiện và đặc biệt phải luôn lớn hơn giá trị của mức Trần hiện tại. Trong khoảng thời gian từ lúc giá trị Trần nhỏ hơn, tới khi nó vượt lên bằng với Fitness hiện tại thì đó cũng chính là giai đoạn mà giải thuật đang tận dụng chính khoảng thời gian này để thử nhiều hướng đi khác nhau và lựa chọn ra các hướng đi tiềm năng sắp tới cho riêng mình.
- Việc chọn lựa Desired Fitness (Độ phù hợp mong muốn) mà bản chất của dữ liệu với sự nỗ lực giải thuật có thể đạt tới. Nếu lựa chọn Desired Fitness quá lớn (vượt quá khả năng cho phép của dữ liệu), nghĩa là giải thuật có ít thời gian lựa chọn hay tìm kiếm các hướng đi tiềm năng, điều này khiến cho giải thuật nhanh chóng lạc vào đỉnh cục bộ, gây ra hiện tượng restart từ khá sớm. Còn ngược lại, nếu chọn lựa Desired Fitness quá nhỏ, tức đang đánh giá quá thấp về khả năng của dữ liệu, thì giải thuật sẽ dừng lại một fitness không tốt trong khi khả năng của dữ liệu vẫn có thể tăng thêm được nữa. Như vậy, ta cần phải chọn lựa giá trị Desired Fitness phù hợp với đúng bản chất của dữ liệu thông qua thực nghiệm và đánh giá.

## **3. Giải thuật Kết hợp Di truyền với Tìm Kiếm Cục Bộ (Combination of GA and Local Search)**

### **3.1. Giải thuật Mã Giả Kết hợp Di truyền với Tìm Kiếm Cục Bộ**



Pha 1 : Áp dụng Giải thuật Di truyền (Genetic Algorithm) đối với quần thể

- Số lượng cá thể : 3000
- Số lượng thế hệ : 400
- Toán tử chọn lọc : Có sử dụng sức ép chọn lọc 2.0
- Toán tử lai ghép : Sử dụng Chiến lược Template ngẫu nhiên với tỉ lệ 90% số lượng 1
- Toán tử đột biến : Chiến lược đột biến cơ bản (chọn 1 giá trị bất kỳ cho một gene bất kỳ)
- Toán tử thay thế : Chọn tập cá thể mới tốt nhất từ quần thể ban đầu với tập Offsprings

Pha 2 : Chọn ra 5 cá thể tốt nhất từ GA. Đưa 5 cá thể này qua các giải thuật Local Search (như Leo Đồi, Luyện Kim, Tăng Tràn, Leo Đồi Cải Tiến,...) với giới hạn 300 vòng lặp, nhằm đẩy cá thể leo lên tới đỉnh đồi trong tìm kiếm cục bộ.

### **3.2. Phân tích Giải thuật Kết hợp Di truyền với Tìm Kiếm Cục Bộ**

- Chúng ta có thể nhận thấy rằng : Ưu điểm về Thời gian của GA cũng chính là nhược điểm trong thời gian chạy của Local Search. Hơn nữa, nhược điểm trong việc cải thiện tốt hơn nữa Fitness trong GA lại là ưu điểm (vốn có) của một giải thuật Local Search
- Giải thuật Di truyền thường hội tụ từ khá sớm (vòng lặp thứ 300-400) và cho kết quả fitness chung cuộc ở mức độ khá tốt, nhưng chưa thực sự có khả năng đẩy cá thể lên tới đỉnh điểm của ngọn đồi (giá trị fitness tốt nhất). Vì vậy, ta sẽ kết hợp Local Search như một pha thứ 2 giúp cải thiện các lời giải tốt nhất từ pha GA. Chúng ta có thể hy vọng về tính hiệu quả trong sự kết hợp này, và thực vậy qua thực nghiệm nhóm đã nhận thấy bộ kết hợp (tuy đơn giản này) nhưng lại đem lại hiệu quả đáng kinh ngạc, đặc biệt trong một thời gian đủ ngắn.

## **III. THỰC NGHIỆM ĐÁNH GIÁ CÁC GIẢI THUẬT**

### **1. Thực nghiệm – Giải thuật Leo Đồi**

Vòng lặp	Fitness	Thời gian	Ghi chú
12000	0.60346013	24.975	Restart tại 398
12000	0.610521756	28.85	Restart tại 412
12000	0.631072733	26.874	Restart tại 423
12000	0.617942042	10.274	Restart tại 482
12000	0.614241659	25.439	Restart tại 410
<b>Trung bình</b>	<b>0.612</b>	<b>23.2824</b>	

## **2. Thực nghiệm – Giải thuật Leo Đồi Cải Tiến**

Vòng lặp	Fitness	Thời gian	Mức độ hy sinh	Ghi chú
15000	0.871953198	985.138	0.008 và giảm dần 10% sau 300 vòng lặp	- Mức độ hy sinh 0.008 hơi cao khiến cho danh sách hàng xóm cho vòng lặp tiếp theo bị "loãng" (chứa nhiều cá thể kém chất lượng) khiến giải thuật khó chọn lọc hướng tốt lên đỉnh đồi.
15000	0.879436793	1002.935 (16 phút)	0.005 và giảm dần 10% sau 300 vòng lặp	- Mức độ hy sinh 0.005 khá phù hợp qua thực nghiệm
15000	0.879614922	926.071 (15 phút)	0.005 và giảm dần 10% sau 300 vòng lặp	- Tại vòng lặp 5900 đã có thể lên được Fitness 0.87 - Tại vòng lặp 9000 Fitness mới đẩy lên được 0.877 (khi gần lên tới đỉnh, quá trình cải thiện fitness diễn ra chậm hơn, chững lại) - Tại vòng lặp thứ 13600, Fitness đẩy lên 0.8792 - Tại vòng lặp thứ 15000, Fitness đẩy lên 0.8796

15000	0.84044063	949.789	0.002 và giảm dần 10% sau 300 vòng lặp	<ul style="list-style-type: none"> <li>- Mức độ hy sinh nhỏ 0.002. Xu hướng chọn hàng xóm tốt cao hơn, dẫn tới lạc vào đỉnh cục bộ. Tức kết quả (chung cuộc) thu được kém hơn.</li> <li>- Có dấu hiệu hội tụ (chững lại) với Fitness 0.839 tại những vòng lặp 11000. Không có sự thay đổi đáng kể tới vòng lặp thứ 15000 với Fitness cao nhất chỉ là : 0.840</li> </ul>
<b>Trung bình</b>	<b>0.879 (xét với trường hợp Độ hy sinh phù hợp 0.005)</b>			Cần qua thực nghiệm để lựa chọn Mức độ Hy sinh phù hợp : 0.005

### 3. Thực nghiệm – Giải thuật Luyện Kim

Vòng lặp	Fitness	Thời gian	Nhiệt độ	Ghi chú
15000	0.881052606	918.712 (15 phút)	Duy trì nhiệt độ cỡ 0.0005 (cỡ $10^{-4}$ )	<ul style="list-style-type: none"> <li>- Đã lên được kỷ lục 0.881 tại các vòng lặp thứ 9000. Sau vòng lặp thứ 9000 gần như không có thêm một sự cải thiện kỷ lục nào nữa. Lúc này, thuật toán rơi vào trạng thái "xuống đồi" nhằm "dò la" những cá thể kém hơn để tìm ra một hướng đi tốt hơn Kỷ lục trước đó.</li> <li>- Hiếm xảy ra tình trạng Restart (danh sách hàng xóm cho bước tiếp theo thường khác 0, do chiến lược chấp nhận "hy sinh" của giải thuật Luyện Kim)</li> </ul>
15000	0.880020188	929.401 (15 phút)	Duy trì nhiệt độ cỡ 0.0005 (cỡ $10^{-4}$ )	
15000	0.880755151	892.628 (14 phút)	Duy trì nhiệt độ cỡ 0.0005 (cỡ $10^{-4}$ )	

<b>Trung bình</b>	<b>0.88</b>	<b>15 phút</b>		
-------------------	-------------	----------------	--	--

#### **4. Thực nghiệm – Giải thuật Tăng-Giảm Trần Tuyến Tính**

<b>Vòng lặp</b>	<b>Fitness</b>	<b>Thời gian</b>	<b>Fitness mong muốn</b>	<b>Ghi chú</b>
15000	0.877231302	865.674 (14 phút)	0.9	Restart tại vòng lặp thứ 14000
15000	0.868493711	846.013 (14 phút)	0.9	Restart tại vòng lặp thứ 13988
15000	0.879796056	786.439 (13 phút)	0.95	Restart tại vòng lặp thứ 12760
15000	0.879108311	759.934 (13 phút)	0.95	Restart tại vòng lặp thứ 12763
15000	0.874195538	749.72 (13 phút)	0.95	Restart tại vòng lặp thứ 12447
<b>Trung bình</b>	<b>0.875</b>	<b>13.35 phút</b>		- Tăng Desired Fitness cao (hơn nhiều so với khả năng tốt nhất của cá thể) khiến cho Restart sớm xảy ra hơn - Kỷ lục (fitness) được cải theo mức độ tăng tuyến tính của trần (ceiling), nên quá trình hội tụ (lên đỉnh đồi) diễn ra chậm (cỡ 13000-14000 vòng lặp)

#### **5. Thực nghiệm – Giải thuật Di truyền (GA)**

Thông số - Kích thước quần thể 3000, Kích thước Offsprings :  $0.9 \times 3000$ , Tỷ lệ đột biến : 0.1, Số vòng lặp : 300-400, Sức ép : 2.0, Tỷ lệ One-Template: 0.9

<b>Vòng lặp</b>	<b>Fitness</b>	<b>Thời gian</b>	<b>Ghi chú</b>
300	0.859142601	8.09	
300	0.854313153	7.677	
400	0.854891902	10.32	
400	0.857068538	10.274	
400	0.860231456	10.633	

<b>Trung bình</b>	<b>0.8568</b>	<b>9.399</b>	- Hội tụ khá sớm (tại các vòng lặp thứ 300-400) - Yêu cầu quần thể đủ lớn (3000) - Khi đạt tới Fitness cao, GA với toán tử Lai ghép và Đột biến tỏ ra "khó khăn" trong việc cải thiện tốt hơn lên đỉnh đồi vì chiến lược chưa mềm dẻo của bản thân toán tử lai ghép và đột biến
-------------------	---------------	--------------	---

**6. Thực nghiệm – Giải thuật Kết hợp GA và Luyện Kim**

- Thông số - GA : Quần thể 3000, Offspring : 0.9, Đột biến : 0.1, Số vòng lặp : 300-400, Sức ép : 2.0, Tỷ lệ One-Template: 0.9
- Thông số - Luyện Kim : Nhiệt độ ban đầu 0.0005 (giữ không thay đổi)

Pha	Giải thuật	Vòng lặp	Fitness đạt được	Thời gian chạy	Ghi chú
1	GA	400	0.854793659	10.68	Hội tụ ở những vòng lặp 300
2	Luyện Kim	300	0.881489292	18.76	Ít khả năng bị Restart, chạy hết 300 vòng lặp. Nhưng hội tụ (chững lại) ở những vòng lặp 200-220
			0.877578742	18.13	
			0.879061184	18.12	
			0.87778438	18.13	
			0.878521546	17.98	
Trung Bình			0.878887029	28.91	
Tốt nhất			0.881489292		

**7. Thực nghiệm – Giải thuật Kết hợp GA và Leo Đồi Cải Tiến**

- Thông số - GA : Quần thể 3000, Offspring : 0.9, Đột biến : 0.1, Số vòng lặp : 300-400, Sức ép : 2.0, Tỷ lệ One-Template: 0.9
- Thông số - Leo Đồi Cải Tiến (Sacrificed Hill Climbing) : Mức độ hy sinh 0.00001 ( $10^{-5}$ ), sau 300 lần giảm 10% mức độ hy sinh

Pha	Giải thuật	Vòng lặp	Fitness đạt được	Thời gian chạy	Ghi chú
1	GA	400	0.851899838	10.64	Hội tụ ở những vòng lặp 300
2	Leo Đồi Cải Tiến	300	0.868986964	19.06	- Quá trình cải thiện fitness chậm, "không tập trung" do chiến lược chấp nhận hy sinh của nó. - Nếu áp dụng mức độ hy sinh 0.005 thì kết quả trung bình fitness thu được 0.86. Còn nếu giảm sâu hơn mức độ hy sinh xuống 0.00001 thì kết quả trung bình thu được là 0.865
			0.87050214	19.08	
			0.864820581	20.09	
			0.859057956	18.78	
			0.863252088	18.69	
Trung Bình			0.865323946	29.78	
Tốt nhất			0.87050214		

**8. Thực nghiệm – Giải thuật Kết hợp GA và Tăng giảm trần tuyến tính**

- Thông số - GA : Quần thể 3000, Offspring : 0.9, Đột biến : 0.1, Số vòng lặp : 300-400, Sức ép : 2.0, Tỷ lệ One-Template: 0.9
- Thông số - Tăng/Giảm Trần (Degraded Ceiling) : Desired Fitness = 0.9

Pha	Giải thuật	Vòng lặp	Fitness đạt được	Thời gian chạy	Ghi chú
1	GA	400	0.852500439	11.25	Hội tụ ở những vòng lặp 300
2	Tăng giảm trần	300	0.876263942	10.17	- Dễ bị Restart. Thường Restart tại vòng lặp thứ 170-175 - Cần phải chọn tham số Desired Fitness phù hợp (phản ánh đúng khả năng Fitness của dữ liệu)
			0.880822867	15.39	
			0.880235187	10.94	
			0.879983962	10.90	
			0.880456068	11.03	
Trung Bình			0.879552405	22.74	
Tốt nhất			0.880822867		

**9. Thử nghiệm – Giải thuật Kết hợp GA và Leo Đồi**

- Thông số - GA : Quần thể 3000, Offspring : 0.9, Đột biến : 0.1, Số vòng lặp : 300-400, Sức ép : 2.0, Tỷ lệ One-Template: 0.9
- Thông số - Leo Đồi : Số vòng lặp 300

Pha	Giải thuật	Vòng lặp	Fitness đạt được	Thời gian chạy	Ghi chú
1	GA	400	0.848841248	10.320	Hội tụ ở những vòng lặp 300
2	Hill Climbing	300	0.879268791	13.657	Hội tụ tại vòng lặp 120-130
			0.880288932	16.098	
			0.881104274	12.224	
			0.878771973	12.303	
			0.876626793	11.712	
Trung Bình			0.879212153	23.519	
Tốt nhất			0.881104274		



**Tài liệu tham khảo**

1. Tài liệu Mô hình hóa bài toán Điện toán Sương Mù – Tác giả : Nhóm nghiên cứu Fog Computing
2. Cuốn Metaheuristics for Hard Optimization – Tác giả : J. Drezo A. P ´ etrowski và P. Siarry E. Taillard