

ĐA TÁC VỤ TIẾN HÓA: KỸ THUẬT TỐI ƯU HÓA MỚI

Lại Thị Nhung^a, Nguyễn Thị Hòa^a, Phạm Văn Hạnh^b,
Lê Đăng Nguyên^c, Lê Trọng Vĩnh^{d*}

^aKhoa Khoa học Cơ bản, Trường Đại học Điều dưỡng Nam Định, Nam Định, Việt Nam

^bTrung tâm Tin học, Trường Đại học Luật Hà Nội, Hà Nội, Việt Nam

^cPhòng Khảo thí và Đảm bảo chất lượng, Trường Đại học Hải Phòng, Hải Phòng, Việt Nam

^dKhoa Toán - Cơ - Tin học, Trường Đại học Khoa học Tự nhiên,

Đại học Quốc gia Hà Nội, Hà Nội, Việt Nam

*Tác giả liên hệ: Email: vinhlt@vnu.edu.vn

Lịch sử bài báo

Nhận ngày 01 tháng 03 năm 2018

Chỉnh sửa ngày 01 tháng 05 năm 2018 | Chấp nhận đăng ngày 14 tháng 05 năm 2018

Tóm tắt

Trong vài thập kỷ vừa qua, các thuật toán tiến hóa (Evolutionary Algorithms - EA) đã được áp dụng thành công để giải các bài toán tối ưu khác nhau trong khoa học và kỹ thuật. Các vấn đề này thường được phân loại vào hai nhóm: i) Tối ưu hóa đơn mục tiêu (single-objective optimization - SOO), trong đó mỗi điểm trong không gian tìm kiếm của bài toán được ánh xạ thành một giá trị mục tiêu vô hướng; và ii) Tối ưu hóa đa mục tiêu (multi-objective optimization-MOO), trong đó mỗi một điểm trong không gian tìm kiếm của bài toán được ánh xạ thành một vec-tơ (các giá trị) mục tiêu. Trong bài báo này, chúng tôi sẽ giới thiệu một loại thứ ba hoàn toàn mới đó là đa tác vụ tiến hóa (evolutionary multitasking), cho phép giải đồng thời nhiều bài toán tối ưu khác nhau trên một quần thể duy nhất và được gọi là tối ưu hóa đa nhân tố (multifactorial optimization - MFO).

Từ khóa: Các thuật toán tiến hoá; Đa tác vụ tiến hoá.

Mã số định danh bài báo: <http://tckh.dlu.edu.vn/index.php/tckhdh/article/view/428>

Loại bài báo: Bài báo nghiên cứu gốc có bình duyệt

Bản quyền © 2018 Các tác giả.

Cấp phép: Bài báo này được cấp phép theo CC BY-NC-ND 4.0

EVOLUTIONARY MULTITASKING: A NEW OPTIMIZATION TECHNIQUE

Lai Thi Nhung^a, Nguyen Thi Hoa^a, Pham Van Hanh^b,
Le Dang Nguyen^c, Le Trong Vinh^{d*}

^aThe Faculty of Science, Namdinh University of Nursing, Namdinh, Vietnam

^bThe Center of Information, Hanoi Law University, Hanoi, Vietnam

^cThe Department of Quality Assurance and Examination, Haiphong University, Haiphong, Vietnam

³The Faculty of Mathematics, Mechanics, and Informatics, Hanoi University of Science,
Vietnam National University, Hanoi, Vietnam

* Corresponding author: Email: vinhlt@vnu.edu.vn

Article history

Received: March 01st, 2018

Received in revised form: May 01st, 2018 | Accepted: May 14th, 2018

Abstract

In the last decades, evolutionary algorithms (EAs) have been successfully applied to solve various optimization problems in science and technology. These issues are usually categorized into two groups: i) Single-objective optimization (SOO), where each point in the search space of the problem is mapped to a target value scalar; and ii) Multi-objective optimization (MOO), where each point in the search space of the problem is mapped to a target vector. In this paper, we will introduce a completely new kind of third-party evolutionary multitasking, which allows simultaneous optimization of different optimization problems on a single population and is called multifactorial optimization (MFO).

Keywords: Evolutionary Algorithms; Evolutionary Multitasking.

Article identifier: <http://tckh.dlu.edu.vn/index.php/tckhdh/article/view/428>

Article type: (peer-reviewed) Full-length research article

Copyright © 2018 The author(s).

Licensing: This article is licensed under a CC BY-NC-ND 4.0

1. GIỚI THIỆU

Các thuật toán tiến hóa EA (Back, Hammel, & Schwefel, 1997; Coello, 2006; & Fonseca & Fleming, 2007) là các kỹ thuật tối ưu Heuristics dựa trên nguyên lý của Darwin về sự lựa chọn tự nhiên. Thuật toán bắt đầu với nhóm các cá thể (gọi là quần thể) trải qua các thao tác (lai ghép, đột biến) tương tự như quá trình sinh sản trong tự nhiên để tạo ra thế hệ các con cháu. Tiếp theo đó, các tương tác trên chính các cá thể đó bảo tồn tính di truyền làm cho một số cá thể phù hợp tốt hơn với “môi trường” và loại bỏ đi các cá thể xấu đối với “môi trường”. Từ “môi trường” ở đây được sử dụng như một phép ẩn dụ cho ngữ cảnh của hàm mục tiêu đang được tối ưu hóa. Và trong vài thập kỷ vừa qua, các thuật toán tiến hóa đã được áp dụng thành công để giải các bài toán tối ưu khác nhau trong khoa học, kỹ thuật. Các vấn đề này thường được phân loại vào hai nhóm: i) Tối ưu hóa đơn mục tiêu SOO (*single-objective optimization*), trong đó mỗi điểm trong không gian tìm kiếm của bài toán được ánh xạ thành một giá trị mục tiêu vô hướng; và ii) Tối ưu hóa đa mục tiêu MOO (*multi-objective optimization*), trong đó mỗi một điểm trong không gian tìm kiếm của bài toán được ánh xạ thành một vec-tơ (các giá trị) mục tiêu. Hầu hết sự thiết kế của các thuật toán tiến hóa này đều tập trung vào việc giải *một bài toán tối ưu* tại mỗi thời điểm dựa trên một quần thể mà chưa có sự quan tâm đến việc giải quyết *nhiều bài toán tối ưu khác nhau* đồng thời trên cùng một quần thể. Do vậy, trong bài báo này, chúng tôi sẽ giới thiệu một dạng hoàn toàn mới của các thuật toán tiến hóa đó là đa tác vụ tiến hóa (*evolutionary multitasking*), cho phép giải đồng thời nhiều bài toán tối ưu khác nhau trên một quần thể duy nhất và được gọi là *tối ưu hóa đa nhân tố* MFO (*multifactorial optimization*) (Gupta, Ong, & Feng, 2017).

Cũng giống như các thuật toán tiến hóa cổ điển dựa trên sinh học tiến hóa, MFO cũng được thiết kế dựa trên mô hình của *sự kế thừa đa nhân tố*, trong đó, những đặc điểm phát triển phức tạp của thế hệ con cái (*offspring*) bị ảnh hưởng bởi sự tương tác của các yếu tố di truyền và văn hoá (Cloninger, Rice, & Reich, 1979; Tavarani, & Bennett, 2013). Nói một cách khác, di truyền và văn hoá không thể xem xét một cách độc lập. Ví dụ, trong khi những gì một cá thể học được có thể phụ thuộc vào kiểu gen của nó, sự lựa chọn tác động lên hệ gen có thể được tạo ra hoặc thay đổi bởi sự lan rộng của một đặc điểm văn hoá nào đó. Những đặc điểm văn hoá này thường bắt nguồn từ việc học tập xã hội hoặc từ cha mẹ (*parents*) trải qua một số tập quán và trở thành sở thích nhất định cho con cái của họ. Sự tương đương tính toán của việc thừa kế đa nhân tố, cho mục đích của đa tác vụ tiến hóa hiệu quả, được thiết lập bằng cách xem xét mỗi tác vụ (*task*) tối ưu đóng góp một “môi trường” khác nhau trong đó các cá thể con cháu được tiến hóa. Do đó, MFO dẫn tới sự tồn tại của nhiều *biểu tượng văn hóa* (*culture bias* hay *memes*), là một trào lưu văn hóa hoặc một ý tưởng lan truyền rộng rãi và nhanh chóng trong xã hội. Trong lịch sử, *memes* giống như một “vấn đề văn hóa” lan truyền từ người này sang người khác thông qua việc truyền miệng, bắt chước,... hay “tin tưởng quá” thành truyền thuyết, truyện ngụ ngôn,...) mà mỗi cái tương ứng với một tác vụ tối ưu. Vì vậy, mặc dù cấu trúc cơ bản của MFO sẽ tương tự như của một EA cổ điển, nó được tăng cường với các tính năng được vay mượn từ các mô hình đa thừa kế.

Phần còn lại của bài báo sẽ được tổ chức như sau: Mục 2 sẽ được dùng để trình bày các định nghĩa và khái niệm liên quan đến tối ưu hóa đa nhân tố; Mục 3 sẽ trình bày thuật toán tối ưu hóa đa nhân tố cho phép đa tác vụ tiến hóa; Mục 4 sẽ trình bày MFO qua ví dụ để minh họa; và Cuối cùng là các kết luận và hướng nghiên cứu tiếp theo.

2. CÁC ĐỊNH NGHĨA VÀ KHÁI NIỆM LIÊN QUAN

Trong phần này, chúng tôi sẽ giải thích các khái niệm của MFO và mô tả sự thực hiện của một cá thể trong môi trường đa tác vụ như thế nào. Trước khi đi vào chi tiết, điều quan trọng là phải xem xét mối quan hệ giữa các tác vụ cấu thành trong MFO. Nó là giá trị đề cập đến khái niệm về đa tác vụ tiến hóa không áp đặt bất kỳ hạn chế nghiêm ngặt về mối quan hệ giữa các tác vụ. Cụ thể, các tác vụ có thể là khác biệt, hoặc là các thành phần phụ thuộc lẫn nhau của một vấn đề nhiều thành phần lớn. Tuy nhiên, trong bài báo này, chúng tôi chỉ tập trung vào trường hợp đầu, nghĩa là khi không có kiến thức về bất kỳ sự phụ thuộc giữa các tác vụ. Nói cách khác, chúng tôi xem xét đa nhiệm trên các vấn đề tối ưu hóa mà thường được xem độc lập (như các tác vụ khác biệt). Vì vậy, mục đích của MFO không phải là để tìm ra sự cân bằng tối ưu giữa các hàm mục tiêu cấu thành. Thay vào đó, mục tiêu là tối ưu hóa toàn bộ tác vụ một cách đầy đủ và đồng thời, với sự trợ giúp của sự tìm kiếm dựa trên quần thể.

Với phạm vi nói trên, xem xét tình huống trong đó K tác vụ tối ưu hóa được thực hiện đồng thời. Không mất tính tổng quát, giả sử rằng tất cả các tác vụ đều là bài toán tối thiểu hóa (*minimization problems*). Tác vụ thứ j^{th} ký hiệu là T_j , được xem là có không gian tìm kiếm X_j với hàm mục tiêu là $f_j: X_j \rightarrow R$. Thêm nữa, mỗi một tác vụ có thể bị ràng buộc bởi một số các phương trình và bất phương trình điều kiện phải được thỏa mãn đối với một giải pháp được xem là phù hợp.

Trong bối cảnh như vậy, người ta định nghĩa MFO là một chiến lược đa tác vụ tiến hóa được xây dựng trên sự song song hóa hoàn toàn của việc tìm kiếm dựa vào quần thể với mục đích tìm $\{x_1, x_2, \dots, x_{K-1}, x_K\} = \operatorname{argmin}\{f_1(x), f_2(x), \dots, f_{K-1}(x), f_K(x)\}$, trong đó x_j là một giải pháp phù hợp trong X_j .

Để thiết kế các lời giải cho MFO, cần phải mô hình hóa một kỹ thuật tổng quát cho việc so sánh các cá thể trong một quần thể của một môi trường đa tác vụ. Để làm điều này, đầu tiên phải định nghĩa một tập các thuộc tính cho mọi cá thể p_i , trong đó $i \in \{1, 2, \dots, |P|\}$, trong một quần thể P . Chú ý rằng, các cá thể được mã hóa trong một không gian tìm kiếm duy nhất chứa đựng X_1, X_2, \dots, X_K , và có thể được giải mã thành từng giải pháp cụ thể cho mỗi tác vụ (trong K tác vụ cần tối ưu). Dạng giải mã của cá thể p_i có thể được viết như là $\{x_1^i, x_2^i, \dots, x_K^i\}$, trong đó $x_1^i \in X_1, x_2^i \in X_2, \dots$ và $x_K^i \in X_K$.

Định nghĩa 1 (Factorial Cost: Giá của của một cá thể đối với từng tác vụ): *Factorial cost* Ψ_j^i của cá thể p_i đối với tác vụ T_j là $\Psi_j^i = \lambda \cdot \delta_j^i + f_j^i$, trong đó λ là một hằng số phạt lớn, f_j^i và δ_j^i là giá trị mục tiêu và tổng số sự vi phạm ràng buộc tương ứng của p_i

đối với tác vụ T_j . Do vậy, nếu p_i phù hợp đối với tác vụ T_j (sự vi phạm ràng buộc bằng 0), chúng ta có $\Psi_j^i = f_j^i$.

Định nghĩa 2 (Factorial rank: Xếp hạng của một cá thể đối với từng tác vụ): *Factorial rank* r_j^i của cá thể p_i trên tác vụ T_j là chỉ số (*index*) của p_i trong quần thể P được sắp xếp theo thứ tự tăng dần của Ψ_j^i .

Trong khi gán *Factorial rank*, mỗi khi $\Psi_j^a = \Psi_j^b$ đối với một cặp cá thể p_a và p_b , thứ tự trước sau được lựa chọn ngẫu nhiên. Tuy nhiên, vì hiệu năng của hai cá thể là tương đương đối với tác vụ thứ j^{th} , vì vậy chúng ta gán nhãn chung như là *j-counterparts* (bản sao hay giống nhau đối với tác vụ thứ j^{th}).

Định nghĩa 3 (Scalar Fitness: Sự thích hợp): Danh sách các xếp hạng của một cá thể p_i trên tất cả các tác vụ (K tác vụ) $\{r_1^i, r_2^i, \dots, r_K^i\}$ có thể được biến đổi sang một hình thức đơn giản hơn là giá trị *Scalar Fitness* ϕ_i dựa trên thứ tự xếp hạng tốt nhất của p_i trên tất cả các tác vụ, nghĩa là $\phi_i = 1 / \min_{j \in \{1, 2, \dots, K\}} \{r_j^i\}$.

Định nghĩa 4 (Skill Factor: Tác vụ đầy đủ hoặc tác vụ tốt nhất [nhân tố kỹ năng]): *Skill factor* τ_i của cá thể p_i là tác vụ trong số tất cả các tác vụ của MFO mà cá thể p_i là hiệu quả nhất, nghĩa là $\tau_i = \operatorname{argmin}_j \{r_j^i\}$ trong đó $j \in \{1, 2, \dots, K\}$.

Một khi sự thích hợp của các cá thể được qui theo *Định nghĩa 3*, sự so sánh hiệu năng có thể được thực hiện một cách dễ dàng. Ví dụ, p_a được xem là trội hơn p_b trong ngữ cảnh đa nhân tố nếu $\phi_a > \phi_b$. Nếu hai cá thể có cùng *skill factor* $\tau_a = \tau_b = j$ (phù hợp với tác vụ thứ j^{th} nhất), chúng ta gán nhãn chúng là *strong counterparts* (giống nhau mạnh).

Điều quan trọng cần lưu ý là các thủ tục được mô tả từ trước đến nay để so sánh các cá thể không phải là tuyệt đối. Vì *factorial rank* của một cá thể (và rõ ràng *scalar fitness* và *skill factor* của nó) phụ thuộc vào hiệu suất của mọi cá thể khác trong quần thể, sự so sánh là phụ thuộc vào quần thể thực tế. Tuy nhiên, thủ tục đảm bảo rằng nếu một cá thể p^* ánh xạ tới sự tối ưu hóa toàn cục của một tác vụ bất kỳ thì $\phi^* \geq \phi_i$ với mọi $i \in \{1, 2, \dots, K\}$.

Định nghĩa 5 (Multifactorial optimality: Sự tối ưu đa nhân tố): Một cá thể p^* , với một danh sách các giá trị mục tiêu $\{f_1^*, f_2^*, \dots, f_K^*\}$, được xem là tối ưu hóa trong ngữ cảnh đa nhân tố nếu và chỉ nếu $\exists j \in \{1, 2, \dots, K\}$ sao cho $f_j^* \leq f_j(x_j)$ đối với mọi x_j phù hợp trong X_j .

3. THUẬT TOÁN TỐI ƯU HÓA ĐA NHÂN TỐ

Như đã nói ở trên, thuật toán tiến hóa MFEA (*multifactorial evolutionary algorithm*) cho vấn đề tối ưu hóa đa nhân tố MFO là dựa trên mô hình kế thừa đa nhân tố trong sinh học, vì vậy, nguyên tắc làm việc của thuật toán là dựa trên sự truyền *gen* và các đặc trưng văn hóa từ thế hệ cha mẹ sang thế hệ con cái. Mặc dù cấu trúc cơ bản của thuật toán MFEA là tương tự như thuật toán EA cổ điển, việc thêm vào các đặc trưng văn hóa sẽ biến MFEA thành các lời giải MFO hiệu quả. Sau đây là thuật toán mô tả cấu trúc cơ bản của MFEA.

- *Đầu vào*: Quần thể được khởi tạo ngẫu nhiên cho bài toán;
- *Đầu ra*: Cá thể tối ưu (lời giải) cho bài toán;
- *Chi tiết thuật toán*:
 - 1 Khởi tạo quần thể ban đầu và lưu trữ trong *current-pop* (*P*)
 - 2 Đánh giá mọi cá thể đối với mọi tác vụ tối ưu trong môi trường đa tác vụ
 - 3 Tính toán skill factor τ của mỗi cá thể
 - 4 While (điều kiện dừng chưa thỏa mãn) do
 - 5 Áp dụng các thao tác di truyền tới *current-pop* (*P*) để sinh ra *offspring-pop* (*C*) [Xem Thuật toán 2]
 - 6 Đánh giá các cá thể trong *offspring-pop* chỉ đối với các tác vụ được lựa chọn [Xem Thuật toán 3]
 - 7 Gộp *current-pop* (*P*) và *offspring-pop* (*C*) tạo thành *intermediate-pop* (*PUC*)
 - 8 Cập nhật *scalar fitness* và *skill factor* cho mọi cá thể trong *intermediate-pop* (*PUC*)
 - 9 Lựa chọn các cá thể tốt nhất từ *intermediate-pop* (*PUC*) để hình thành *current-pop* (*P*) kế tiếp.
 - 10 End while

3.1. Khởi tạo quần thể

Giả sử rằng K tác vụ tối ưu được thực hiện đồng thời, số chiều của tác vụ thứ j^{th} là D_j . Theo đó, chúng ta định nghĩa không gian tìm kiếm hợp nhất với số chiều là $D_{multitask} = \max_j \{D_j\}$ với $j \in \{1, 2, \dots, K\}$. Vec-tơ này được gọi là *chromosome* của cá thể. Trong bước khởi tạo quần thể, mọi cá thể đều là một vectơ của $D_{multitask}$ các biến ngẫu nhiên (có giá trị trong khoảng $[0, 1]$). Về cơ bản, chiều thứ j^{th} của không gian tìm kiếm hợp nhất được biểu diễn bởi một khóa ngẫu nhiên y_i , và khoảng giá trị cố định (từ 0 đến 1) biểu diễn ràng buộc của không gian hợp nhất. Với các biểu diễn như vậy, đối với tác vụ T_j trong cá thể p_i , chúng ta sẽ sử dụng D_j khóa ngẫu nhiên đầu tiên của *chromosome* tương ứng với p_i .

3.2. Các kỹ thuật di truyền

Tương tự như các thuật toán EA cổ điển, MFEA cũng sử dụng hai toán tử di truyền đó là *lai ghép* (*crossover*) và *đột biến* (*mutation*). Điểm khác biệt chính của

MFEA là hai cá thể cha mẹ được lựa chọn ngẫu nhiên cho phép toán lai ghép phải đáp ứng một số điều kiện. Nguyên tắc tiếp theo là việc lai ghép ngẫu nhiên chỉ ra rằng các cá thể thích kết hợp với cá thể thuộc cùng một kiểu “văn hoá”: Trong MFEA, *skill factor* (τ) được xem như là một đại diện tính toán của sự thiên lệch văn hoá (*culture bias*) của một cá thể. Do vậy, hai cá thể cha mẹ được lựa chọn ngẫu nhiên chỉ được thực hiện lai ghép nếu chúng có cùng *skill factor*. Ngược lại, nếu *skill factor* là khác nhau, sự lai ghép chỉ xảy ra theo một xác suất lai ghép ngẫu nhiên được qui định (*rpm*) hoặc phép đột biến được sử dụng. Các bước tạo ra các cá thể con được mô tả trong thuật toán các thao tác di truyền như sau:

- *Đầu vào*: Hai cá thể cha mẹ;
- *Đầu ra*: Các cá thể con;
- *Chi tiết thuật toán*:
 - 1 Xét hai cá thể cha mẹ p_a và p_b được lựa chọn ngẫu nhiên trong *current-pop* (P)
 - 2 Sinh một số ngẫu nhiên *rand* trong khoảng $[0,1]$
 - 3 If ($\tau_a = \tau_b$) hoặc (*rand* < *rpm*) then
 - 4 Thực hiện lai ghép hai cá thể p_a và p_b sinh ra hai cá thể con c_a và c_b
 - 5 Else
 - 6 Đột biến p_a sinh ra cá thể con c_a
 - 7 Đột biến p_b sinh ra cá thể con c_b
 - 8 End if

Theo thuật toán này, rõ ràng sự lai ghép trong thế giới tự nhiên được sử dụng trong các mô hình kế thừa đa nhân tố để giải thích các đặc tính đã được phát triển qua nhiều thế hệ. Tham số *rpm* được sử dụng để cân bằng khai thác và tìm kiếm không gian tìm kiếm. Giá trị của *rpm* gần với 0 hàm ý rằng chỉ các cá thể có “văn hoá” giống nhau mới được lai ghép, trong khi giá trị gần 1 cho phép lai ghép hoàn toàn ngẫu nhiên.

3.3. Di truyền văn hóa theo chiều dọc thông qua sự bắt chước có chọn lọc

Rõ ràng, chi phí tính toán cho việc đánh giá từng cá thể cho mọi tác vụ đang giải quyết thường quá đắt và do đó không được mong muốn. Để làm cho MFO tính toán thực tế, MFEA phải được thiết kế hiệu quả. Điều này đạt được bằng cách giảm tổng số các đánh giá càng nhiều càng tốt. Một quan sát quan trọng mà chúng tôi đưa ra là một cá thể được tạo ra trong MFEA có thể sẽ không đạt được hiệu năng cao cho tất cả các tác vụ. Do đó, một cá thể chỉ được đánh giá cho các tác vụ được chọn mà nó có thể thực hiện tốt. Để kết hợp tính năng này vào MFEA một cách đơn giản, thuật toán mượn ý tưởng của việc lan truyền “trào lưu văn hóa”. Trong thừa kế đa nhân tố, lan truyền “trào lưu văn hóa” từ cha mẹ đến con cái (theo chiều dọc – từ trên xuống dưới) là một kiểu (*mode*) của sự kế thừa song song với sự thừa hưởng sinh học theo hướng kiểu hình của con cái bị ảnh hưởng trực tiếp bởi kiểu hình của cha mẹ.

Tương tự, MFEA cho phép con cái bắt chước nhân tố kỹ năng (đặc điểm hay trào lưu văn hoá) của bất kỳ người nào là cha mẹ. Tính năng này, được gọi là *bắt chước có chọn lọc* (*selective imitation*), đã đạt bằng cách làm theo các bước trong Thuật toán 3 đó là thay vì đánh giá cá thể con cho mọi tác vụ, nó chỉ được đánh giá cho một tác vụ. Chi tiết Thuật toán 3 như sau:

- *Đầu vào*: Hai cá thể cha mẹ;
- *Đầu ra*: Cá thể con giống cha hoặc mẹ;
- *Chi tiết thuật toán*:

Một cá thể ‘c’ hoặc có cá thể cha mẹ p_a và p_b hoặc một cha mẹ (p_a hay p_b)

```

1  If ('c' có hai cha mẹ) then
2      Sinh một số ngẫu nhiên rand giữa 0 và 1
3      If (rand < 0.5 ) then
4          'c' bắt chước  $p_a$ : Cá thể con được đánh giá chi cho tác vụ có thứ tự là  $\tau_a$ 
5      Else
6          'c' bắt chước  $p_b$ : Cá thể con được đánh giá chi cho tác vụ có thứ tự là  $\tau_b$ 
7      End If
8  Else
9      'c' bắt chước theo cá thể cha mẹ (đột biến): Cá thể con được đánh giá chi cho tác vụ có thứ tự  $\tau$  của cá thể cha mẹ đột biến tạo ra nó
10 End If
11 Factorial cost của cá thể c trên các tác vụ còn lại được đặt là  $\infty$  (một số rất lớn)
```

3.4. Sự lựa chọn

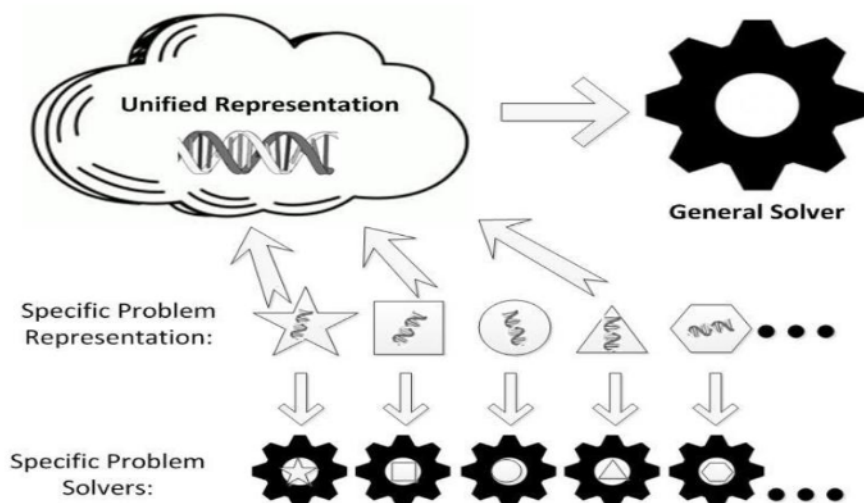
Cũng giống như EA cổ điển, MFEA cũng lựa chọn các cá thể tốt hơn cho thế hệ kế tiếp, tức là các cá thể có giá trị φ (*scalar fitness*) lớn hơn.

4. MINH HỌA THUẬT TOÁN MFEA

Như đã nói ở trên, mục tiêu của bài báo này chỉ để giới thiệu một kỹ thuật tiến hóa tối ưu mới, do vậy chúng tôi không quá đi chi tiết vào thực nghiệm áp dụng thuật toán trên một bài toán cụ thể nào. Thay vào đó, chúng tôi cố gắng *giải thích chi tiết* các khái niệm, các bước thực hiện của thuật toán. Hơn nữa, vấn đề quan trọng nhất cho người muốn thiết kế các thuật toán EA là làm sao biểu diễn mỗi giải pháp của bài toán thực tế thành một cá thể trong thuật toán EA và ngược lại. Vì vậy, phần này tập trung vào việc giới thiệu cách biểu diễn (mã hóa) các giải pháp của các bài toán cụ thể vào cùng “không gian tìm kiếm hợp nhất” và ngược lại (giải mã).

Ý tưởng mã hóa và giải mã của thuật toán MFEA có thể được minh họa như Hình 1. Theo đó, trong với EA truyền thống, mỗi tác vụ có một sự biểu diễn cụ thể (*specific problem representation*), từ đó tìm ra các lời giải cụ thể (*specific problem solvers*). Khi chuyển sang MFEA, tất cả các sự biểu diễn của các tác vụ được được

chuyển thành một biểu diễn hợp nhất (*unified representation*), và từ đó tìm ra lời giải tổng quát (*general solver*).



Hình 1. Sự biểu diễn của các tác vụ trên các không gian tìm kiếm khác nhau được chuyển về không gian tìm kiếm hợp nhất

Nguồn: Gupta và ctg. (2017).

Để làm rõ sự mã hóa và giải mã các giải pháp của các bài toán tối ưu khác nhau vào cùng một không gian tìm kiếm hợp nhất, chúng tôi sẽ sử dụng hai bài toán tối ưu là KP (*Knapsack problem*) và QAP (*Quadratic assignment problem*) do Tayarani và Bennett (2013) đề xuất.

4.1. Bài toán Knapsack

Bài toán KP phát biểu như sau: Cho trước một tập n đồ vật, một đồ vật có trọng lượng w_i và giá trị v_i với $i \in \{1, 2, \dots, n\}$, hãy chọn các đồ vật cho vào một ba lô chứa được một trọng lượng tối đa là W sao cho tổng giá trị của các đồ vật là lớn nhất.

Trong các thuật toán EA truyền thống cho bài toán KP, mỗi một cá thể được biểu diễn bằng một vectơ $x = \{x_1, x_2, \dots, x_n\}$ của các biến nhị phân, trong đó $x_i = 1$ chỉ ra rằng đồ vật thứ i^{th} được chọn để cho vào ba lô, ngược lại, không cho đồ vật thứ i^{th} vào ba lô. Một cách đơn giản để suy luận các biến nhị phân từ các khóa ngẫu nhiên là $x_i = 1$ nếu khóa ngẫu nhiên $y_i \geq 0.5$, ngược lại $x_i = 0$.

4.2. Bài toán Quadratic Assignment

Bài toán QAP phát biểu như sau: Đưa ra n phương tiện và n vị trí với khoảng cách giữa các vị trí là $d(i, j)$ và yêu cầu luồng (*flow*) từ vị trí i đến vị trí j là $f(i, j)$. Mục

tiêu của bài toán QAP là ấn định mỗi phương tiện đến một vị trí, tức là tìm ánh xạ $\varphi: n \rightarrow n$, sao cho tổng chi phí là nhỏ nhất, nghĩa là $\sum_{i=1}^n \sum_{j=1}^n f(i,j) \cdot d(\varphi(i), \varphi(j)) \rightarrow \min$.

Trong các thuật toán EA truyền thống cho bài toán QAP, mỗi một cá thể được biểu diễn bằng một vec-tơ $x = \{x_1, x_2, \dots, x_n\}$ trong đó $x_i \in \{1, 2, \dots, n\}$ và $x_i \neq x_j$ với $i \neq j$; nghĩa là phương tiện thứ i^{th} được ấn định đến vị trí thứ x_i . Một cách đơn giản để suy luận các biến x_i từ các khóa ngẫu nhiên $y = \{y_1, y_2, \dots, y_n\}$ đó là x_i là số thứ tự (*index*) của y_i trong dãy y_1, y_2, \dots, y_n được sắp xếp theo thứ tự tăng dần.

4.3. Ví dụ số

Giả sử $n = 9$ trong bài toán KP và $n = 6$ trong bài toán QAP. Như vậy, $D_{multitask} = \max\{9, 6\} = 9$. Điều này có nghĩa là, trong không gian hợp nhất, *chromosome* sẽ gồm 9 khóa ngẫu nhiên. Giả sử giá trị của *chromosome* trong không gian hợp nhất như trong Bảng 1.

Bảng 1. Biểu diễn của *Chromosome*

STT	1	2	3	4	5	6	7	8	9
y	0.79	0.31	0.53	0.17	0.60	0.26	0.65	0.69	0.75

Ghi chú: $y = (0.79, 0.31, 0.53, 0.17, 0.60, 0.26, 0.65, 0.69, 0.75)$.

Dựa vào phân tích ở trên, ta dễ dàng tính được vector biểu diễn cá thể tương ứng cho bài toán KP (độ dài 9) và QAP (độ dài 6) như sau: Cá thể p cho bài toán KP: $p_{KP} = (1, 0, 1, 0, 1, 0, 1, 1, 1)$; Cá thể p cho bài toán QAP: $p_{QAP} = (6, 3, 4, 1, 5, 2)$.

5. KẾT LUẬN

Trong bài báo này, chúng tôi đã trình bày chi tiết thuật toán tối ưu đa nhân tố cho phép đa tác vụ cùng tiến hóa trên một quần thể duy nhất. Đây là một kỹ thuật tiến hóa hoàn toàn mới dựa vào kế thừa đa nhân tố của sinh học tiến hóa, trong đó chỉ ra rằng thế hệ con cái không chỉ ảnh hưởng bởi gen di truyền từ thế hệ cha mẹ mà còn bị ảnh hưởng bởi các vấn đề văn hóa làm cho thế hệ con cái tốt hơn hay xấu đi với môi trường. Mặc dù vậy, bài báo mới chỉ tập chung giải thích chi tiết các khái niệm, các bước thực hiện của thuật toán mà chưa đi sâu vào việc giải các bài toán cụ thể trong khoa học và kỹ thuật để đánh giá hiệu quả của thuật toán. Chúng tôi hy vọng sẽ tiếp tục thực hiện công việc này trong tương lai.

TÀI LIỆU THAM KHẢO

- Back, T., Hammel, U., & Schwefel, H. P. (1997). Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(1), 3-17.
- Cloninger, C. R., Rice, J., & Reich, T. (1979). Multifactorial inheritance with cultural

- transmission and assortative mating. II. A general model of combined polygenic and cultural inheritance. *American Journal of Human Genetics*, 31(2), 176-198.
- Coello, C. A. C. (2006). Evolutionary multi-objective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1), 28-36.
- Fonseca, C. M., & Fleming, P. J. (2007). An overview of evolutionary algorithms in multi-objective optimization. *Evolution Computing*, 3(1), 1-6.
- Gupta, E., Ong, Y. S., & Feng, L. (2017). Multifactorial evolution: Towards evolutionary multitasking. *IEEE Transactions on Evolutionary Computation*, 20(3), 343-357.
- Rice, J., Cloninger, C. R., & Reich, T. (1978). Multifactorial inheritance with cultural transmission and assortative mating. I. Description and basic properties of the unitary models. *American Journal of Human Genetics*, 30(6), 618-643.
- Tayarani, N. M. H., & Bennett, P. A. (2013). On the landscape of combinatorial optimization problems. *IEEE Transactions on Evolutionary Computation*, 18(3), 420-434.