

AN ADAPTIVE MULTIFACTORIAL EVOLUTIONARY ALGORITHM FOR INTER-DOMAIN PATH COMPUTATION UNDER NODEDEFINED DOMAIN UNIQUENESS CONSTRAINT

Pham Dinh Thanh

Tay Bac University

ARTICLE INFO	ABSTRACT
Received: 22/02/2022 Revised: 20/4/2022 Published: 21/4/2022	Nowadays, the rapid development of networks in size and complexity in architecture leads to the optimization of network routing becoming more and more important. The Inter-Domain Path Computation under Node defined Domain Uniqueness Constraint (IDPC-DU) has much attention from communication research. IDPC-DU is NP-Hard so approximation approaches are suitable to solve this problem for instances having large dimensionality. Multifactorial evolutionary algorithm (MFEA) is an effective approach to deal with the various types of problems. This paper proposed an approach based on an algorithm based on an Adaptive Multifactorial Evolutionary Algorithm (dMFEA-II) for solving IDPC-DU under node defined domain uniqueness constraint. The encoding and evaluating methods based on the permutation representation are also introduced. The proposed algorithm is evaluated on the two types of instances. The experimental results point out the effectiveness of the proposed algorithm in comparing with existing algorithms.
KEYWORDS Evolutinary Algorithm Transfer Optimization Multifactorial Optimization Inter-Domain Path Computation Evolutionary Multitasking	

THUẬT TOÁN TIẾN HÓA ĐA NHÂN TỔ THÍCH NGHI GIẢI BÀI TOÁN TÌM ĐƯỜNG ĐI LIÊN MIỀN VỚI RÀNG BUỘC MIỀN DUY NHẤT TRÊN NÚT MẠNG

Phạm Đình Thành

Trường Đại học Tây Bắc

THÔNG TIN BÀI BÁO	TÓM TẮT
Ngày nhận bài: 22/02/2022 Ngày hoàn thiện: 20/4/2022 Ngày đăng: 21/4/2022	Ngày nay, cùng với sự phát triển nhanh chóng của các mạng thông tin về cả kích thước và độ phức tạp, vấn đề tối ưu chi phí định tuyến trong mạng ngày càng trở nên cấp thiết. Bài toán tìm đường đi liên miền với ràng buộc miền duy nhất (IDPC-DU) là một trong các bài toán tối ưu chi phí định tuyến nhận được nhiều sự quan tâm của các nhà nghiên cứu. Do IDPC-DU thuộc lớp bài toán NP-Khó nên hướng tiếp cận gần đúng được đánh giá là phù hợp khi kích thước dữ liệu đầu vào lớn. Trong các thuật toán gần đúng, thuật toán tiến hóa đa nhân tổ (MFEA) là một trong những thuật toán hiệu quả để giải nhiều lớp bài toán khác nhau. Nghiên cứu này đề xuất áp dụng thuật toán tiến hóa đa nhân tổ thích nghi (dMFEA-II) vào giải bài toán IDPC-DU với ràng buộc được xét trên các nút mạng. Nghiên cứu cũng đề xuất phương pháp mã hóa và đánh giá cá thể dựa trên biểu diễn hóa vị. Thuật toán đề xuất được đánh giá trên hai tập dữ liệu khác nhau. Kết quả thực nghiệm đã cho thấy tính hiệu quả của thuật toán đề xuất so với các thuật toán đã có.
TỪ KHÓA Thuật toán tiến hóa Tối ưu hóa chuyển giao Tối ưu hóa đa nhân tổ Tối ưu đường liên miền Tiến hóa đa tác vụ	

DOI: <https://doi.org/10.34238/tnu-jst.5579>

Email: thanhpd@utb.edu.vn

<http://jst.tnu.edu.vn>

114

Email: jst@tnu.edu.vn

1. Giới thiệu

Trong kỷ nguyên công nghệ phát triển nhanh chóng như hiện nay, rất nhiều thiết bị cần được kết nối với các dịch vụ hoặc các thiết bị khác thông qua các hệ thống mạng. Do đó sẽ dẫn tới hình thành các hệ thống mạng vô cùng lớn, cũng như các thách thức trong việc xây dựng phương thức kết nối hiệu quả giữa các thiết bị.

Trước các thách thức đó, các hệ thống mạng lớn thường được chia nhỏ thành các miền (multi-domains - mạng đa miền) [1]. Việc chia nhỏ này nhằm giải quyết các vấn đề liên quan tới khả năng mở rộng và đảm bảo tính bảo mật [2]. Đối với mạng đa miền, việc xác định liên kết giữa các nút mạng trong mỗi miền được thực hiện bởi một phần tử xác định đường liên kết (Path Computation Element - PCE) [3]. Để trao đổi thông tin giữa các miền, các phần tử PCE phải liên kết với nhau theo một kiến trúc nhất định. Trong nghiên cứu [4], tác giả Lorenzo Maggi và các cộng sự đã giới thiệu bài toán tìm được đi ngắn nhất giữa hai nút trong mạng liên miền (gọi là bài toán *Inter-Domain Path Computation under Domain Uniqueness constraint - IDPC-DU*). Trong bài toán IDPC-DU mạng liên miền cần thỏa mãn ràng buộc về miền (mỗi miền được đi qua nhiều nhất là một lần). Có hai biến thể của bài toán IDPC-DU: biến thể thứ nhất ràng buộc miền trên tập cạnh (gọi là bài toán *Inter-Domain Path Computation under Edge-defined Domain Uniqueness Constraint - IDPC-EDU*); biến thể thứ hai là ràng buộc miền trên đỉnh (gọi là bài toán *Inter-Domain Path Computation under Node-defined Domain Uniqueness Constraint - IDPC-NDU*). Cả hai biến thể của bài toán IDPC-DU đều được chứng minh là thuộc lớp bài toán NP-Khó [4]. Cũng trong nghiên cứu này, các tác giả cũng giới thiệu thuật toán lập trình động để giải bài toán IDPC-DU với độ phức tạp tính toán là $O(|V|^2 2^{|D|} |D|^2)$ với $|V|$ là số nút và $|D|$ là số miền. Do độ phức tạp cao nên thuật toán lập trình động khó hiệu quả trong thực tế với các mạng có số miền lớn. Do đó, việc lựa chọn hướng tiếp cận giải xấp xỉ để tìm lời giải “đủ tốt” là cần thiết.

Trong những năm gần đây, các thuật toán tiến hóa là một trong các nhóm thuật toán nhận được nhiều sự quan tâm nghiên cứu [5]. Thuật toán tiến hóa được đánh giá là phù hợp để giải nhiều bài toán phức tạp hoặc phi tuyến ngay cả đối với nhiều bài toán được đánh giá là khó giải khi sử dụng một số thuật toán tối ưu khác [6].

Thuật toán tiến hóa đa nhân tố (MFEA) đề xuất lần đầu năm 2016 và là lược đồ mới trong tính toán tiến hóa [7]. So với thuật toán tiến hóa cơ bản, thuật toán tiến hóa đa nhiệm ưu điểm về khả năng có thể giải đồng thời nhiều tác vụ và quá trình tìm kiếm lời giải được cải thiện về cả tốc độ và chất lượng [7]. Năm 2019, tác giả Kavitesh Kumar Bali và cộng sự [7] đề xuất cải tiến của thuật toán MFEA (ký hiệu MFEA-II) dựa trên khai thác điểm tương đồng giữa các tác vụ. Nghiên cứu cũng đã chỉ ra hiệu quả của thuật toán MFEA-II đối với các hàm benchmark liên tục. Năm 2020, tác giả Eneko Osaba [8] đề xuất thuật toán MFEA-II áp dụng cho các bài toán tối ưu tổ hợp (ký hiệu dMFEA-II). Kết quả thực nghiệm đã chỉ ra hiệu quả của thuật toán dMFEA-II khi giải bài toán người du lịch và bài toán định tuyến xe với ràng buộc sức chứa.

Với các ưu điểm như trên, đã có một số tác giả nghiên cứu sử dụng các thuật toán tiến hóa và thuật toán tiến hóa đa nhân tố vào giải bài toán IDPC-NDU. Trong nghiên cứu [9], tác giả đề xuất thuật toán hai mức dựa trên thuật toán tiến hóa (ký hiệu là TLGA). Do cần phải lưu trữ và giải mã thông tin của cả hai mức nên mã hóa được sử dụng trong thuật toán TLGA tương đối phức tạp và tốn bộ nhớ để lưu trữ, kéo theo các toán tử tiến được đề xuất cũng tương đối phức tạp, ảnh hưởng tới hiệu quả của thuật toán. Tác giả Đỗ Tuấn Anh và các cộng sự [10] cũng đề xuất thuật toán áp dụng thuật toán di truyền (ký hiệu là PGA) dựa trên việc chia bài toán ban đầu thành hai bài toán con. Mặc dù chất lượng lời giải đã được cải thiện so với thuật toán TLGA nhưng do sử dụng mã hóa dựa trên độ ưu tiên và tập trung vào tạo ra lời giải hợp lệ nên có thể dẫn tới chất lượng lời giải có thể không được cải thiện.

Để sử dụng các ưu điểm của thuật toán dMFEA-II [8] và khắc phục một số hạn chế trên, nghiên cứu này đề xuất áp dụng thuật toán dMFEA-II để giải bài toán IDPC-NDU. Đóng góp chính của thuật toán đề xuất là:

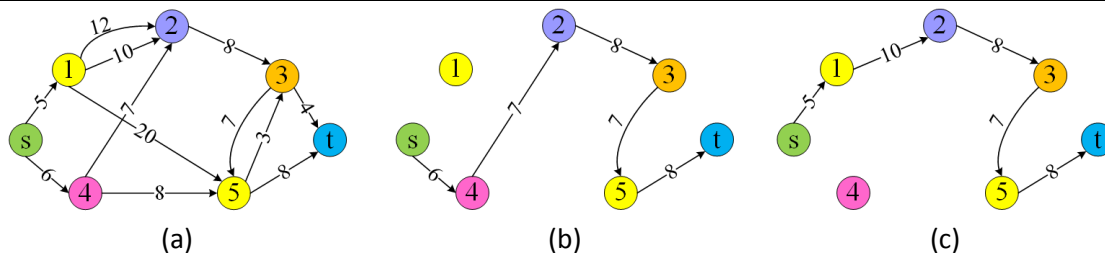
- Đề xuất cách mã hóa lời giải bài toán IDPC-DU dựa trên biểu diễn hoán vị.
- Đề xuất thuật toán dựa trên sự kết hợp giữa thuật toán dMFEA-II và thuật toán Dijkstra.
- Thuật toán đề xuất sử dụng được các toán tử tiến hóa có sẵn nên dễ cài đặt.
- Phân tích hiệu quả của thuật toán đề xuất trên các tập dữ liệu khác nhau.

Các phần còn lại của nghiên cứu được tổ chức như sau: phần 2 trình bày phát biểu bài toán IDPC-NDU và các khái niệm liên quan; phần 3 trình bày về thuật toán đề xuất; phần 4 trình bày các kết quả thực nghiệm và đánh giá thuật toán; phần kết luận của nghiên cứu được trình bày trong phần 5.

2. Phát biểu bài toán

Xét đa đồ thị có hướng, có trọng số $G = (V, E, w, C)$, với V, E và w lần lượt là tập các nút, tập các cạnh và ma trận trọng số cạnh của đồ thị G ; tập các đỉnh V được phân hoạch thành K cụm (cluster) đôi một không giao nhau (mỗi cụm tương đương với một miền của mạng) $C = \{C^1, C^2, \dots, C^H\}$. Gọi hai nút $s, t \in V$ lần lượt gọi là các nút nguồn và đích; Mục tiêu của bài toán IDPC-NDU là tìm đường đi có chi phí nhỏ nhất p từ nút nguồn s tới nút đích t sao cho p thỏa mãn ràng buộc về miền duy nhất đối với đỉnh (đường đi p đi qua mỗi miền nhiều nhất một lần). Bài toán IDPC-NDU được phát biểu dưới chi tiết như sau:

Đầu vào:	- Đa đồ thị có hướng, có trọng số $G = (V, E, w, C)$.
	- Tập đỉnh V được phân hoạch thành H miền $C = \{C^1, C^2, \dots, C^H\}$,
	$C^i \cap C^j = \emptyset \forall i, j \in \{1, \dots, H\}$.
	- Một nút nguồn $s \in V$.
	- Một nút đích $t \in V$.
Đầu ra:	Đường đi $p = (p_1, p_2, \dots, p_l)$ với $p_1 = s$ và $p_l = t$.
Ràng buộc:	Đường đi p đi ra khỏi một miền thì không được đi vào lại miền đó nữa. Nghĩa là nếu $p_i \in C^d$ và $p_{i+1} \notin C^d$ thì $p_{i+j} \notin C^d, \forall j \geq 2, d \in \{1, \dots, H\}$.
Hàm mục tiêu:	$f(p) = \sum_{i=1}^{l-1} w(p_i, p_{i+1}) \rightarrow \min$



Hình 1. Minh họa về định nghĩa bài toán IDPC-NDU với (a) đồ thị đầu vào, (b) lời giải hợp lệ, (c) lời giải không hợp lệ

Hình 1 **Error! Reference source not found.** minh họa phát biểu bài toán IDPC-NDU, trong đó Hình 1(a) minh họa đồ thị đầu vào với 7 đỉnh và 6 miền (mỗi màu của một đỉnh tương ứng với một miền), các số ghi bên cạnh mỗi cạnh là trọng số của cạnh đó. Hình 1(b) minh họa đường đi $p_1 = (s, 4, 2, 3, 5, t)$ là một lời giải hợp lệ của bài toán IDPC-NDU với chi phí $f(p_1) = 36$. Hình 1(c) minh họa đường đi $p_2 = (s, 1, 2, 3, 5, t)$ không phải là lời giải hợp lệ của bài toán IDPC-NDU do vi phạm ràng buộc về miền duy nhất khi đi ra miền màu vàng tại nút 1 nhưng vào lại miền này tại nút 5.

3. Thuật toán đề xuất

Phần này trình bày về áp dụng thuật toán đề xuất (ký hiệu P-dMFEA) vào giải IDPC-NDU.

3.1. Sơ đồ thuật toán

Do lời giải của bài toán IDPC-NDU có thể được xây dựng thông qua thứ tự của các miền nên mỗi lời giải bài toán IDPC-NDU chỉ cần mã hóa thông qua thứ tự của các miền. Do đó, khi cần đánh giá cá thể, thuật toán IDPC-NDU phải thực hiện giải mã cá thể sau đó mới tiến hành xây dựng đồ thị G' và áp dụng thuật toán tìm đường đi ngắn nhất từ đỉnh nguồn tới đỉnh đích. Các bước chính của thuật toán P-dMFEA được minh họa dưới dạng mã giả trong Hình 2.

Input: Đồ thị $G_i = (V_i, E_i, w_i, C_i)$, $i \in \{1, \dots, K\}$; Đỉnh nguồn $s_i \in V_i$; Đỉnh đích $t_i \in V_i$;
Output: Các lời giải của bài toán IDPC-NDU;

```

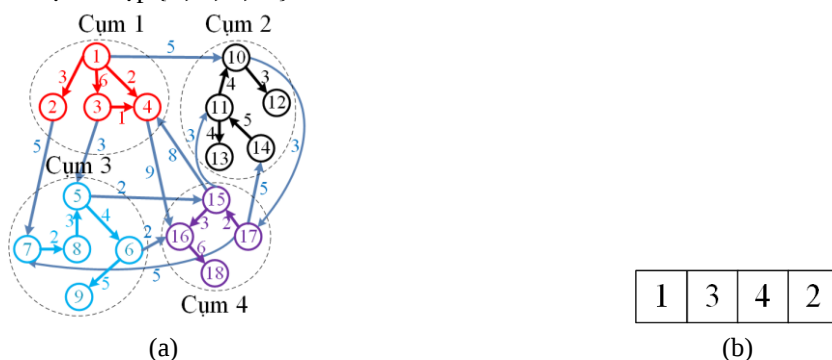
1 begin
2    $t \leftarrow 0$ ;
3    $P_t \leftarrow$  Khởi tạo ngẫu nhiên  $N$  cá thể trong không gian USS;
4   foreach cá thể  $ind_j \in P_t$  do
5     Gán ngẫu nhiên chỉ số kỹ năng phù hợp nhất (skill factor)  $\tau_j$  cho cá thể;
6     Xây dựng lời giải  $s_j$  của bài toán IDPC-NDU dựa trên  $ind_j$  và  $\tau_j$   $\triangleright$  Xem mục 3.4 và 3.5;
7     Tính chi phí đối với mỗi tác vụ (factorial cost) của lời giải  $s_j$   $\triangleright$  Xem mục 3.5;
8     Tính xếp hạng đối với mỗi tác vụ (factorial rank), giá trị thích nghi vô hướng (scalar fitness) của cá thể  $ind_j$ ;
9   end
10  while điều kiện dừng chưa thỏa mãn do
11     $P'_t \leftarrow$  Tournament Selection( $P_t$ )  $\triangleright$  Chọn các cá thể cha mẹ;
12     $O_t \leftarrow$  Thực hiện lai ghép và đột biến ( $P'_t$ )  $\triangleright$  Xem mục 3.3;
13    foreach cá thể  $c_j \in O_t$  do
14      Xây dựng lời giải  $s'_j$  của bài toán IDPC-NDU dựa trên  $c_j$  và chỉ số kỹ năng phù hợp nhất của  $c_j$ ;
15      Tính chi phí đối với mỗi tác vụ của  $c_j$  dựa trên chi phí của  $s'_j$   $\triangleright$  Xem mục 3.5
16    end
17     $R_t \leftarrow O_t \cup P_t$ ;
18    Cập nhật xếp hạng đối với mỗi tác vụ, giá trị thích nghi vô hướng và chỉ số kỹ năng phù hợp nhất của các cá thể trong tập  $R_t$ ;
19     $P_{t+1} \leftarrow$  Chọn  $N$  cá thể tốt nhất trong tập  $R_t$ ;
20     $t \leftarrow t + 1$ ;
21  end
22  return Các lời giải của tốt nhất của các tác vụ;
23 end

```

Hình 2. Lược đồ thuật toán đề xuất P-dMFEA

3.2. Mã hóa lời giải

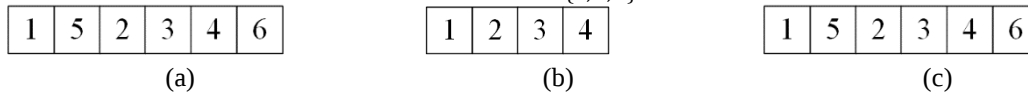
Trong thuật toán P-dMFEA, mỗi lời giải của bài toán IDPC-NDU được mã hóa bởi thứ tự của các cụm. Do đó, nếu đồ thị đầu vào có m cụm thì cá thể của một tác vụ trong thuật toán P-MFEA là một hoán vị của tập $\{1, 2, \dots, m\}$.



Hình 3. Minh họa về mã hóa cá thể trong P-dMFEA với (a) Đồ thị đầu vào, (b) Cá thể

Hình 3 minh họa cách mã hóa cá thể của bài toán IDPC-NDU của thuật toán P-dMFEA, trong đó Hình 3(a) minh họa đồ thị đầu vào, giả sử thứ tự các cụm là: cụm 1 – cụm 2 – cụm 4 – cụm 3 thì thuật toán P-dMFE sẽ mã hóa thành cá thể có dạng tương tự như Hình 3(b).

Do sử dụng mã hóa hoán vị [11], [12] để mã hóa thứ tự các cụm nên thuật toán P-dMFEA sử dụng cách mã hóa hoán vị để mã hóa cá thể trong không gian tìm kiếm chung (*Unified Search Space - USS*). Tức là, giả sử D_k là số cụm của đồ thị đầu vào của tác vụ thứ k , khi đó chiều dài của cá thể trong không gian USS là $D_{USS} = \max_{k \in \{1, \dots, K\}} D_k$.



Hình 4. Minh họa về mã hóa cá thể trong không gian USS với (a) cá thể trong không gian USS , (b) cá thể của tác vụ thứ nhất T_1 , (c) cá thể của tác vụ thứ hai T_2

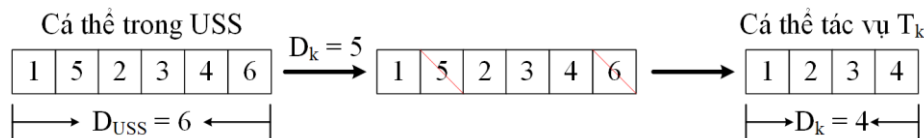
Hình 4 minh họa về biểu diễn cá thể trong không gian USS sử dụng trong thuật toán P-dMFEA giải hai tác vụ có số cụm lần lượt là 4 (tác vụ T_1) và 6 (tác vụ T_2). Hình 4(a) minh họa một cá thể trong không gian USS , Hình 4(b) và Hình 4(c) lần lượt minh họa cá thể của hai tác vụ được xây dựng từ cá thể trong Hình 4(a).

3.3. Toán tử lai ghép và đột biến

Thuật toán P-dMFEA sử dụng toán tử lai ghép liên tác vụ (inter-task crossover) là Dynamic Order Crossover (dOX) [8]; toán tử lai ghép trong tác vụ (intra-task crossover) là Order Crossover (OX) [6]. Toán tử đột biến là đột biến 2-opt [8].

3.4. Toán tử giải mã

Để tìm cá thể ind_k của tác vụ T_k từ cá thể ind_{USS} trong không gian USS , thuật toán P-dMFEA sẽ duyệt và giữ lại các gen của cá thể trong không gian USS có giá trị nhỏ hơn bằng số chiều D_k của tác vụ T_k , sau đó sắp xếp các gen được giữ lại theo đúng thứ tự để được cá thể của tác vụ T_k .



Hình 5. Minh họa về toán tử giải mã trong thuật toán P-dMFEA

Hình 5 minh họa toán tử giải mã trong thuật toán P-dMFEA với cá thể trong không gian USS có số chiều là 6, cá thể của tác vụ cần giải mã có số chiều là 4. Do hai gen có giá trị 5 và 6 của cá thể trong không gian USS lớn hơn 4 nên các gen này không được chọn. Sau khi sắp xếp lại vị trí các gen, ta được cá thể của tác vụ cần tìm ứng với cá thể trong không gian USS như Hình 5.

3.5. Đánh giá cá thể

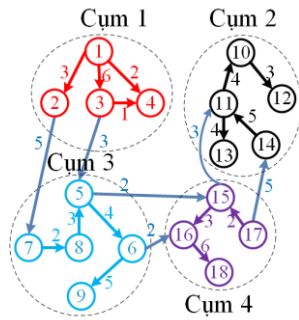
Do mỗi cá thể ind_i trong tác vụ T_k trong thuật toán P-dMFEA là một thứ tự của các cụm cho nên để tìm lời giải của tác vụ T_k ứng với cá thể $ind_i = (ind_i^1, ind_i^2, \dots, ind_i^{D_{USS}})$, ta cần phải tìm đường đi ngắn nhất từ đỉnh nguồn tới đỉnh đích theo đúng thứ tự cụm trên ind_i . Với mỗi đồ thị đầu vào G_k và cá thể ind_i , thuật toán P-dMFEA thực hiện đánh giá cá thể thông qua ba bước:

- Bước 1: Xác định thứ tự các miền từ thông tin của cá thể ind_i .
- Bước 2: Xây dựng đồ thị có hướng, có trọng số G'_k từ đồ thị đầu vào G_k và thứ tự các miền được xác định ở bước 1 bằng cách:

+ Đối với cạnh liên cụm: Với chỉ giữ lại cạnh có hướng từ miền ind_i^j sang miền ind_i^{j+1} ($j = 1, \dots, D_{USS} - 1$) và từ miền $ind_i^{D_{USS}}$ sang miền ind_i^1 .

+ Đối với cạnh trong mỗi cụm: Giữ lại tất cả các cạnh trong cụm.

- Bước 3: Áp dụng thuật toán Dijkstra để tìm đường đi từ nút nguồn tới nút đích trên đồ thị G'_k .



Hình 6. Minh họa về cách xây dựng đồ thị G'_k

Bảng 1. Tóm tắt thông tin về các bộ dữ liệu thực nghiệm

	Type	Số nút	Số miền	Số cạnh	Số bộ dữ liệu
Type 1 Small	Minimum	427	7	14927	16
	Maximum	2002	30	178026	
Type 1 Large	Minimum	2102	12	164811	9
	Maximum	7352	42	1461349	
Type 2 Small	Minimum	52	6	204	20
	Maximum	1902	32	137407	
Type 2 Large	Minimum	2002	17	128021	8
	Maximum	2902	32	229375	

Hình 6 minh họa đồ thị G'_k được xây dựng từ đồ thị đầu vào trong Hình 3(a) và cá thể trong Hình 3(b). Do thứ tự các cụm là cụm 3 $\rightarrow$ cụm 4 nên cạnh (17, 7) ngược hướng với thứ tự này sẽ bị xóa bỏ; cạnh (5, 15) và (6, 16) cùng hướng với thứ tự sẽ được giữ lại.

4. Kết quả thực nghiệm

4.1. Dữ liệu thực nghiệm và tiêu chí đánh giá

Để đánh giá hiệu quả của thuật toán đề xuất, chúng tôi chọn các bộ dữ liệu thuộc 2 tập (có tên là Type 1, Type 2) khác nhau, mỗi tập được phân làm hai nhóm: nhóm các bộ dữ liệu nhỏ (bao gồm các đồ thị có số nút trong khoảng từ 50 tới 2000 và nhóm các bộ dữ liệu lớn (các đồ thị có số nút lớn hơn 2000). Bảng tóm tắt thông tin về các bộ dữ liệu thực nghiệm sử dụng trong nghiên cứu này được trình bày trong Bảng 1 và được lưu trữ tại [13].

Nghiên cứu tập trung phân tích các tiêu chí chất lượng lời giải tìm được của các thuật toán theo giá trị trung bình cộng (Avg) chi phí của hàm mục tiêu, giá trị tốt nhất tìm được trong các lần thực hiện (BF) và độ lệch tiêu chuẩn (Std).

4.2. Môi trường và tham số thực nghiệm

Để đánh giá hiệu quả của thuật toán đề xuất, nghiên cứu tiến hành phân tích hiệu quả của các thuật toán P-dMFEA so sánh với thuật toán TLGA [9] và kết quả tối ưu.

Thuật toán P-dMFEA sử dụng một số tham số thực nghiệm tương tự thiết lập của thuật toán dMFEA-II [8], chi tiết các tham số được tóm tắt trong Bảng 2.

Bảng 2. Bảng tóm tắt các tham số thực nghiệm thuật toán P-dMFEA

Tham số	Giá trị	Tham số	Giá trị
Số lần đánh giá	50.000	$\Delta_{inc}/\Delta_{dec}$	0,99/0,99
Kích thước quần thể	100	Phép lai ghép liên tác vụ	dOX
Tỉ lệ lai ghép	0,95	Phép lai ghép trong tác vụ	OX
Tỉ lệ đột biến	0,05	Phép đột biến	2-opt

Với mỗi bộ dữ liệu, thuật toán được thực nghiệm 30 lần trên máy tính cài đặt hệ điều hành Microsoft Windows 10 với cấu hình: CPU - Intel Xeon E5620, RAM - 8GB. Mã nguồn của thuật toán đề xuất được cài đặt bằng ngôn ngữ lập trình C#.

Thuật toán tìm đường đi ngắn nhất được sử dụng trong P-dMFEA là thuật toán Dijkstra [14].

4.3. Kết quả thực nghiệm

Kết quả so sánh giữa hai thuật toán P-dMFEA và TLGA được trình bày trong Bảng 3 và Bảng 4. Trong hai bảng này, tại mỗi dòng, giá trị tương ứng với lời giải tốt hơn được in nghiêng và có màu đỏ; giá trị bằng giá trị tối ưu được in đậm. Bảng 3 và Bảng 4 cho thấy thuật toán P-dMFEA

có kết quả tốt hơn thuật toán kết quả của thuật toán TLGA trong đa số các trường hợp. Đặc biệt, thuật toán P-dMFEA tìm được lời giải tối ưu tại các bộ dữ liệu *idpc_ndu_252_11_3513* và *idpc_ndu_52_6_204*. Chi tiết kết quả so sánh hai thuật toán như sau:

- Tập dữ liệu Type 1: Thuật toán P-dMFEA có kết quả tốt hơn thuật toán TLGA đối với 22/24 bộ dữ liệu. Thuật toán P-dMFEA kém hơn thuật toán TLGA trên bộ dữ liệu *idpc_ndu_1514_30_78351*. Hai thuật toán có kết quả bằng nhau trên hai bộ dữ liệu *idpc_ndu_842_23_31617* và *idpc_ndu_2102_23_164811*.

- Tập dữ liệu Type 2: Thuật toán P-dMFEA có kết quả tốt hơn thuật toán TLGA trên tất cả các bộ dữ liệu.

5. Kết luận

Thuật toán tiến hóa đa nhân tố là một trong những biến thể mới nhất của thuật toán tiến hóa. Nghiên cứu này đề xuất áp dụng thuật toán tiến hóa đa nhân tố thích nghi vào giải bài toán Tìm đường đi liên miền với ràng buộc miền duy nhất trên nút mạng. Nghiên cứu cũng trình bày các đề xuất về phương pháp mã hóa và đánh giá cá thể. Để đánh giá hiệu quả của thuật toán đề xuất, nghiên cứu so sánh thuật toán đề xuất với thuật toán di truyền hai mức (TLGA) và kết quả tối ưu trên 52 bộ dữ liệu thuộc hai tập khác nhau. Kết quả thực nghiệm cho thấy thuật toán đề xuất cho kết quả tốt hơn thuật toán TLGA trong đa số các trường hợp và tìm được kết quả tối ưu trên một số bộ dữ liệu. Để cải thiện chất lượng lời giải, trong thời gian tới, tác giả sẽ nghiên cứu áp dụng thuật toán biến đổi lân cận vào mã hóa cá thể được đề xuất trong nghiên cứu này.

Bảng 3. Kết quả thực nghiệm của các thuật toán trên bộ dữ liệu thuộc Type 1.

Instances		TLGA			P-dMFEA			Opt
		BF	Avg	Std	BF	Avg	Std	
Kích thước nhỏ	idpc_ndu_1002_12_82252	14	14,2	0,5	14	14,0	0,0	8
	idpc_ndu_1002_22_36564	42	42,0	0,0	35	38,2	3,3	18
	idpc_ndu_1192_19_37744	63	63,0	0,0	40	53,4	7,1	18
	idpc_ndu_1202_22_65521	22	22,5	0,9	22	22,0	0,0	16
	idpc_ndu_1256_21_44446	77	97,1	4,0	43	54,5	10,4	26
	idpc_ndu_1322_22_48821	48	48,0	0,0	44	45,7	2,0	26
	idpc_ndu_1506_9_130556	19	20,0	1,5	11	16,6	3,7	11
	idpc_ndu_1514_16_78292	37	50,3	2,6	33	33,4	0,5	21
	idpc_ndu_1514_30_78351	30	30,0	0,0	30	34,8	9,3	21
	idpc_ndu_1602_22_60574	52	101,8	13,9	46	51,7	11,5	24
	idpc_ndu_1682_23_67612	78	78,0	0,0	46	74,2	9,7	24
	idpc_ndu_1730_20_88509	42	42,0	0,0	39	41,9	0,5	24
	idpc_ndu_2002_22_178026	24	24,7	4,0	24	24,0	0,0	16
	idpc_ndu_427_7_14927	13	13,0	0,0	8	8,2	0,9	8
	idpc_ndu_704_15_16990	34	72,6	13,1	31	33,2	1,5	21
	idpc_ndu_842_23_31617	22	22,0	0,0	22	22,0	0,0	16
Kích thước lớn	idpc_ndu_2102_23_164811	27	27,0	0,2	27	27,0	0,2	18
	idpc_ndu_2402_22_260967	24	25,4	5,0	24	24,1	0,3	16
	idpc_ndu_2494_12_276620	26	30,8	5,9	23	23,0	0,0	16
	idpc_ndu_2502_22_229601	35	36,2	2,7	29	30,1	1,6	18
	idpc_ndu_2522_20_171940	77	77,3	1,5	41	41,3	0,8	18
	idpc_ndu_2715_22_592246	13	13,9	0,8	13	13,0	0,0	8
	idpc_ndu_2918_29_293109	30	31,5	2,0	30	30,2	0,8	21
	idpc_ndu_3161_15_339881	30	41,7	11,9	30	30,0	0,0	21
	idpc_ndu_3602_32_406192	71	78,3	5,4	35	49,8	11,3	21

Bảng 4. Kết quả thực nghiệm của các thuật toán trên bộ dữ liệu thuộc Type 2

	Instances	TLGA			P-dMFEA			Opt
		BF	Avg	Std	BF	Avg	Std	
Kích thước nhỏ	idpc_ndu_1002_32_60942	19	37,1	3,5	19	22,8	6,3	13
	idpc_ndu_102_10_834	19	26,9	5,9	7	8,0	3,1	7
	idpc_ndu_1102_17_56280	49	57,9	2,7	25	27,0	1,7	14
	idpc_ndu_1202_18_68002	31	34,9	0,7	24	24,7	1,5	15
	idpc_ndu_1302_22_78953	45	63,2	6,8	25	27,8	5,1	17
	idpc_ndu_1402_24_92365	41	80,3	9,5	24	28,8	6,1	16
	idpc_ndu_1502_27_104878	30	57,0	9,4	27	33,2	6,8	16
	idpc_ndu_152_14_1869	16	34,8	12,3	16	16,0	0,0	8
	idpc_ndu_1602_14_96765	46	80,9	10,0	32	32,9	0,5	17
	idpc_ndu_1702_18_110993	40	40,0	0,0	20	24,4	6,1	20
	idpc_ndu_1802_21_123666	68	70,9	0,6	33	43,7	11,8	21
	idpc_ndu_1902_27_137407	73	101,1	8,1	30	47,9	14,0	21
	idpc_ndu_202_22_2341	27	31,3	10,3	20	32,4	9,9	9
	idpc_ndu_252_11_3513	24	41,4	9,9	11	11,0	0,0	11
	idpc_ndu_302_12_4930	42	45,9	0,7	11	24,6	5,8	11
	idpc_ndu_352_17_6667	36	46,9	11,3	26	30,4	1,3	13
	idpc_ndu_402_22_8220	32	51,1	24,3	26	32,3	8,4	13
	idpc_ndu_452_32_10406	22	86,7	32,9	22	30,6	19,7	13
	idpc_ndu_502_12_10949	29	54,5	12,0	11	27,2	5,5	7
	idpc_ndu_52_6_204	6	16,3	11,2	6	6,0	0,0	6
Kích thước lớn	idpc_ndu_2002_17_128021	78	78,0	0,0	37	43,0	6,3	16
	idpc_ndu_2102_19_141155	96	107,6	3,5	36	37,5	3,1	20
	idpc_ndu_2202_22_153764	44	88,8	14,1	36	38,3	1,7	21
	idpc_ndu_2302_27_169859	76	105,3	6,7	42	62,8	1,5	22
	idpc_ndu_2402_29_186438	124	136,1	3,0	37	55,7	5,1	23
	idpc_ndu_2502_32_201852	67	124,7	13,2	34	75,7	6,1	25
	idpc_ndu_2602_19_185818	93	93,0	0,0	41	54,8	6,8	21
	idpc_ndu_2802_30_215589	79	154,4	22,2	59	83,4	0,0	25

Lời cảm ơn

Nghiên cứu này được tài trợ bởi Bộ Giáo dục và Đào tạo trong đề tài mã số: B2021-TTB-01.

TÀI LIỆU THAM KHẢO/ REFERENCES

- [1] T. T. B. Huynh, B. T. Ta, B. L. Nguyen, V. H. Nguyen, and D. T. Pham, "Multifactorial Evolutionary Algorithm for Inter-Domain Path Computation under Domain Uniqueness Constraint," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1-8.
- [2] F. Paolucci, F. Cugini, A. Giorgetti, N. Sambo, and P. Castoldi, "A survey on the path computation element (PCE) architecture," *IEEE Commun. Surv. Tutor.*, vol. 15, no. 4, pp. 1819-1841, 2013.
- [3] D. King and A. Farrel, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS," IETF RFC 6805, 2012.
- [4] L. Maggi, J. Leguay, J. Cohen, and P. Medagliani, "Domain clustering for inter-domain path computation speed-up," *Networks*, vol. 71, no. 3, pp. 252-270, 2018.
- [5] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Evolutionary computation 1: Basic algorithms and operators*. CRC press, 2018.
- [6] E. Agoston and Eiben, *Introduction to Evolutionary Computing*. Berlin, Springer-Verlag, 2003.

-
- [7] K. K. Bali, Y. -S. Ong, A. Gupta, and P. S. Tan, "Multifactorial Evolutionary Algorithm with Online Transfer Parameter Estimation: MFEA-II," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 69-83, 2019.
 - [8] E. Osaba, A. D. Martinez, A. Galvez, A. Iglesias, and J. D. Ser, "dMFEA-II: An adaptive multifactorial evolutionary algorithm for permutation-based discrete optimization problems," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, 2020, pp. 1690-1696.
 - [9] T. A. Do, H. L. Nguyen, B. T. Ta, T. T. B. Huynh, and S. Su, "A two-level strategy based on evolutionary algorithm to solve the inter-domain path computation under node-defined domain uniqueness constraint," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*, 2021, vol. 11746, p. 1174620.
 - [10] T. T. B. Huynh, H. L. Nguyen, B. T. Ta, and S. Simon, "A Two-level Genetic Algorithm for Inter-domain Path Computation under Node-defined Domain Uniqueness Constraints," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, 2021, pp. 87-94.
 - [11] Y. Yuan, Y. -S. Ong, A. Gupta, P. S. Tan, and H. Xu, "Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with TSP, QAP, LOP, and JSP," in *Region 10 Conference (TENCON)*, 2016 IEEE, 2016, pp. 3157-3164.
 - [12] L. Zhou, L. Feng, J. Zhong, Y.-S. Ong, Z. Zhu, and E. Sha, "Evolutionary Multitasking in Combinatorial Search Spaces: A Case Study in Capacitated Vehicle Routing Problem", In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, pp. 1-8.
 - [13] D. T. Pham, B. T. Ta, V. H. Ngo, and T. A. Do, "Inter-Domain Path Computation under Node-defined Domain Uniqueness Constraint Instances," *Mendeley Data*, 2022, doi: 10.17632/tpg2nbcsc5.2.
 - [14] J. -C. Chen, "Dijkstra's shortest path algorithm," *J. Formaliz. Math.*, vol. 15, no. 9, pp. 237-247, 2003.