

**KẾT HỢP THUẬT TOÁN TIẾN HÓA VÀ TÌM KIẾM BIẾN ĐỔI LÂN CẬN
ĐỂ GIẢI BÀI TOÁN TÌM ĐƯỜNG ĐI LIÊN MIỀN
VỚI RÀNG BUỘC MIỀN DUY NHẤT TRÊN NÚT MẠNG**

Phạm Đình Thành, Nguyễn Duy Hiếu, Đặng Thị Vân Chi, Phan Trung Kiên
Trường Đại học Tây Bắc

Tóm tắt: Bài toán tìm đường đi liên miền với ràng buộc miền duy nhất trên nút mạng (IDPC-NDU) là một trong những bài toán tối ưu mạng mới được quan tâm nghiên cứu gần đây. Mặc dù đã có các nghiên cứu sử dụng thuật toán tiến hóa (EA) để giải bài toán IDPC-NDU, tuy nhiên hiện chưa có nhiều nghiên cứu về kết hợp giữa thuật toán EA và thuật toán tìm kiếm cục bộ. Vì vậy nghiên cứu này mô tả cách kết hợp giữa thuật toán EA và tìm kiếm biến đổi lân cận (VNS). Trong đó, VNS được sử dụng đối với cá thể tốt nhất của mỗi thế hệ. Để áp dụng được VNS vào giải bài toán IDPC-NDU, nghiên cứu cũng mô tả phương pháp biểu diễn lời giải bài toán IDPC-NDU dưới dạng biểu diễn hoán vị. Thuật toán đề xuất được đánh giá trên các tập dữ liệu với 53 bộ dữ liệu khác nhau. Kết quả thực nghiệm cho thấy thuật toán đề xuất tốt hơn thuật toán so sánh trên 39 bộ dữ liệu.

Từ khóa: Thuật toán tiến hóa, thuật toán tìm kiếm biến đổi lân cận, tìm đường đi liên miền với ràng buộc miền duy nhất.

1. MỞ ĐẦU

Ngày nay, sự phát triển nhanh chóng của mạng Internet và nhu cầu kết nối dẫn tới sự hình thành các hệ thống mạng ngày càng lớn hơn. Các hệ thống mạng lớn thường được chia thành nhiều miền nhỏ, khi đó các mạng được gọi là mạng đa miền [1]. Trong lịch sử hình thành, các mạng đa miền ban đầu được phát triển để đảm bảo yêu cầu về khả năng mở rộng và tính bảo mật [1]. Tuy nhiên hiện nay, bên cạnh các yêu cầu trên, việc xác định được các thuật toán định tuyến hiệu quả trong mạng đa miền là một trong các thách thức nhận được nhiều sự quan tâm nghiên cứu. Gần đây, tác giả L. Maggi và các đồng nghiệp [5] đã phát biểu mô hình hóa một trong các thách thức trên thành bài toán tìm đường đi ngắn nhất giữa hai nút trong mạng liên miền (Inter-Domain Path Computation under Domain Uniqueness constraint - IDPC-DU) [2]. Tùy theo ràng buộc về miền duy nhất được xác định trên tập cạnh hay trên tập đỉnh mà bài toán IDPC-DU có hai biến thể là: Inter-Domain Path Computation under Edge-defined Domain Uniqueness Constraint (ký hiệu là

IDPC-EDU) - với ràng buộc trên tập cạnh; Inter-Domain Path Computation under Node-defined Domain Uniqueness Constraint (ký hiệu là IDPC-NDU) - với ràng buộc trên tập đỉnh. Mặc dù các tác giả đã đề xuất thuật toán lập trình động để tìm lời giải bài toán IDPC-DU [2], nhưng do IDPC-DU là bài toán NP-Khó nên hướng tiếp cận này không phù hợp khi các bộ dữ liệu đầu vào có kích thước lớn.

Do đó, các nghiên cứu gần đây đã sử dụng các thuật toán xấp xỉ để giải bài toán IDPC-DU như: Trong nghiên cứu [3], tác giả đã đề xuất thuật toán tiến hóa thích nghi đa nhân tố với phương pháp mã hóa lời giải bằng biểu diễn hoán vị. Nghiên cứu đã phát huy được các thế mạnh của thuật toán tiến hóa đa nhân tố và phương pháp mã hóa hoán vị để nâng cao chất lượng lời giải tìm được. Trong nghiên cứu [4], các tác giả cũng mô tả việc áp dụng thuật toán di truyền (Two-level Genetic Algorithm, ký hiệu là PGA) dựa trên phương pháp mã hóa thứ tự ưu tiên. Một trong các ưu điểm của thuật toán PGA là có thể sử dụng các toán tử tiến hóa dùng cho mã hóa hoán vị nên dễ dàng cài đặt. Tuy nhiên, lời giải tìm được bởi thuật toán PGA

chưa tốt do phương pháp mã hóa lời giải trong PGA chú trọng vào việc đáp ứng các ràng buộc của bài toán IDPC-NDU. Tác giả Đỗ Tuấn Anh và các đồng nghiệp [5] đã mô tả cách áp dụng thuật toán di truyền (two-level strategy based on evolutionary algorithm, ký hiệu là TLGA) để giải bài toán IDPC-NDU. Trong thuật toán TLGA, các tác giả đã đề xuất phương pháp mã hóa lời giải hai mức cũng như các toán tử tiến hóa dựa trên phương pháp mã hóa này. Tuy nhiên, do thông tin được lưu trữ ở hai mức nên yêu cầu bộ nhớ lưu trữ nhiều và các toán tử lai ghép, đột biến cũng phức tạp hơn. Mặc dù việc áp dụng các thuật toán EA đã giúp cải thiện chất lượng lời giải bài toán IDPC-NDU tìm được nhưng do các thuật toán này vẫn sử dụng phương pháp khởi tạo quần thể ngẫu nhiên nên chất lượng quần thể ban đầu chưa cao, từ đó ảnh hưởng tới chất lượng kết quả cuối cùng.

Mặc dù đã có các nghiên cứu sử dụng thuật toán tiến hóa giải bài toán IDPC-NDU, tuy nhiên, theo hiểu biết của nhóm tác giả, hiện chưa có nghiên cứu nào kết hợp giữa thuật toán tiến hóa và thuật toán tìm kiếm cục bộ trên một quần thể. Cho nên nghiên cứu này thử nghiệm hướng kết giữa thuật toán EA và thuật toán VNS. Đóng góp chính của nghiên cứu gồm:

- Xây dựng cách kết hợp giữa thuật toán EA và thuật toán VNS sử dụng một quần thể [6].

- Xây dựng hướng sử dụng thuật toán VNS vào giải bài toán IDPC-NDU.

Các phần tiếp theo của nghiên cứu gồm: Phần 2 trình bày về định nghĩa bài toán IDPC-NDU và các khái niệm liên quan; Chi tiết về thuật toán đề xuất được mô tả trong Phần 3; Các kết quả thực nghiệm cũng như các so sánh, phân tích về hiệu quả thuật toán đề xuất được trình bày trong Phần 4; Cuối cùng, Phần 5 trình bày các kết quả chính của nghiên cứu cũng như hướng phát triển tiếp theo của nghiên cứu trong thời gian tới.

2. PHÁT BIỂU BÀI TOÁN

Gọi $G = (V, E, w, D)$ là đa đồ thị có hướng, có trọng số. Trong đó, các tập các đỉnh (nút), tập các cạnh và ma trận trọng số cạnh của đồ thị G được lần lượt ký hiệu là V, E và w . Tập đỉnh V được phân hoạch thành k tập con (mỗi tập con gọi là một miền) đôi một không giao nhau $D = \{D^1, D^2, \dots, D^k\}$.

Định nghĩa 1 (miền duy nhất trên tập đỉnh [2]). Đường đi $p = (p_1, p_2, \dots, p_l)$ trên đồ thị G được gọi là thỏa mãn ràng buộc miền duy nhất trên tập đỉnh nếu đường đi p đi ra một miền thì không đi vào lại miền đó nữa. Tức là, nếu $p_i \in D^d$ và $p_{i+1} \notin D^d$ thì $p_{i+j} \notin D^d, \forall j \geq 2, d \in \{1, \dots, k\}$.

Định nghĩa 2 (bài toán IDPC-NDU [2], [7]). Cho trước hai đỉnh $s, t \in V$ (lần lượt gọi là đỉnh nguồn và đỉnh đích), mục tiêu của bài toán IDPC-NDU là tìm đường đi p từ s tới t trên đồ thị G có chi phí nhỏ nhất, sao cho đường đi p thỏa mãn điều kiện ràng buộc về miền duy nhất trên tập đỉnh. Hay nói cách khác, bài toán IDPC-NDU được định nghĩa như sau:

Đầu vào: Đa đồ thị có hướng, có trọng số $G = (V, E, w, D)$.
 - Tập đỉnh V được phân hoạch thành k miền $D = \{D^1, D^2, \dots, D^k\}$, $D^i \cap D^j = \emptyset \forall i, j \in \{1, \dots, k\}$.
 - Đỉnh nguồn $s \in V$, đỉnh đích $t \in V$.

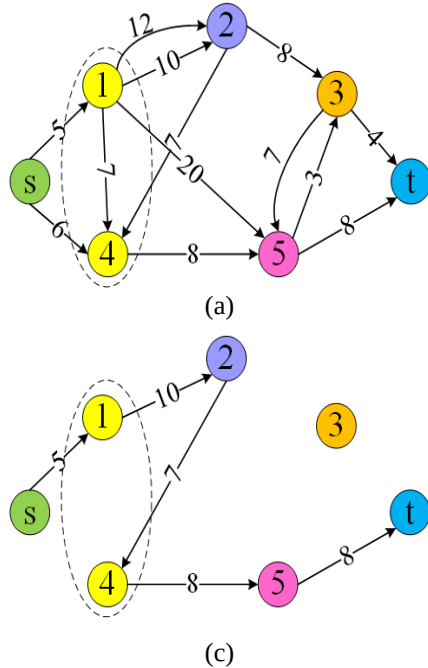
Đầu ra: Đường đi $p = (p_1, p_2, \dots, p_l)$ với $p_1 = s$ và $p_l = t$ thỏa mãn điều kiện miền duy nhất trên tập đỉnh.

Hàm mục tiêu:
$$f(p) = \sum_{i=1}^{l-1} w(p_i, p_{i+1}) \rightarrow \min$$

Hình 1 minh họa định nghĩa bài toán IDPC-NDU với đồ thị đầu vào gồm 6 miền (mỗi màu của đỉnh tương ứng với một miền) và 7 đỉnh. Hình 1(a) minh họa đồ thị đầu vào với các số ghi trên mỗi cạnh là trọng số của cạnh đó; đỉnh s là đỉnh nguồn; đỉnh t là đỉnh đích. Hình 1(b) minh họa đường đi $d_1 = (s, 1, 5, t)$ là một lời giải hợp lệ với chi phí $f(d_1)=33$. Đường đi $d_2 = (s, 1, 2, 4, 5, t)$ trong Hình 1(c) là lời giải không hợp lệ do vi phạm ràng buộc về miền duy nhất: khi đi

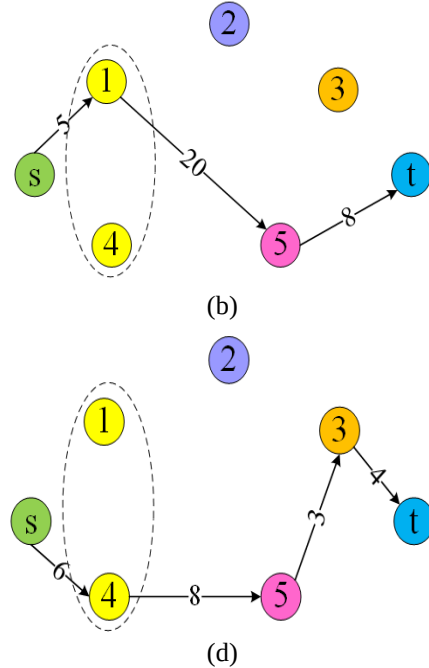
ra miền màu vàng tại nút 1 nhưng vào lại miền này tại nút 4. Lời giải tối ưu của đồ thị đầu vào như trong Hình 1(a) là đường đi $d_3 = (s, 4, 5, t)$ với chi phí $f(d_3)=21$ (xem Hình 1(d)).

3. THUẬT TOÁN ĐỀ XUẤT



dựng lời giải bài toán *IDPC-NDU* từ hoán vị của cá thể đó, sau đó mới tính chi phí của lời giải và giá trị thích nghi của cá thể (dòng lệnh 4–7 và 14–17).

Thuật toán VNS được áp dụng chỉ cho một cá thể tốt nhất của thế hệ hiện tại.



Hình 1. Ví dụ về định nghĩa bài toán *IDPC-NDU* là đồ thị có 6 miền và 7 đỉnh

Phần này mô tả thuật toán đề xuất (ký hiệu *P-MEM*) dựa trên việc áp dụng thuật toán *EA* và thuật toán *VNS* để giải bài toán *IDPC-NDU*.

3.1. Lược đồ thuật toán

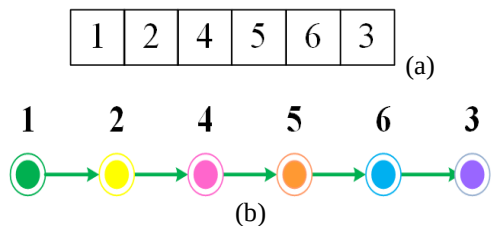
Do lời giải của bài toán *IDPC-NDU* có thể được tạo ra từ thứ tự của các miền nên nghiên cứu này mã hóa mỗi lời giải của bài toán *IDPC-NDU* bằng một hoán vị của các miền. Do đó, để đánh giá mỗi cá thể, thuật toán

P-MEM sẽ phải tạo đồ thị G' (gọi là đồ thị toàn cục) dựa trên thông tin từ hoán vị của các miền mà cá thể đó biểu diễn và đồ thị đầu vào G , sau đó áp dụng thuật toán tìm đường đi ngắn nhất từ đỉnh nguồn tới đỉnh đích để tính chi phí của lời giải.

Các bước chính của thuật toán *P-MEM* được mô tả bằng mã giả trong 0. Do mỗi lời giải của bài toán *IDPC-NDU* được mã hóa bằng hoán vị của các miền nên các toán tử của thuật toán *P-MEM* chỉ xử lý trên các hoán vị này và để tính giá trị thích nghi của mỗi cá thể, thuật toán *P-MEM* phải xây

3.2. Phương pháp mã hóa lời giải

Nếu đồ thị đầu vào có số miền N_d thì mỗi cá thể là một hoán vị của tập $\{1, \dots, N_d\}$.



Hình 2. Ví dụ về phương pháp mã hóa lời giải

0 minh họa ví dụ về phương pháp mã hóa lời giải được sử dụng trong thuật toán *P-MEM* với đồ thị đầu vào như trong Hình 1(a). Giả sử các miền có màu “Xanh lá cây, Vàng, Tím, Hồng, Cam, Xanh nước biển” được gán nhãn lần lượt là: 1, 2, 3, 4, 5 và 6. Hình 2(a) minh họa một cá thể với độ ưu tiên các miền là 2–5–3–1–4–6. Hình 2(b) minh họa đường đi qua các miền tương ứng

với thứ tự các miền của cá thể trong Hình 2(a).

0 và 0 lần lượt minh họa toán tử lai ghép OX và toán tử đột biến SM cho cá thể có độ dài bằng 6.

3.3. Phương pháp khởi tạo quần thể

Do mỗi lời giải của bài toán IDPC-NDU được mã hóa bởi một hoán vị của các cụm nên thuật toán *P-MEM* sử dụng phương pháp tạo ngẫu nhiên một hoán vị của một tập hợp để tạo cá thể trong quần thể ban đầu.

Input: Đa đồ thị có hướng, có trọng số $G = (V, E, w, D)$;
Tập đỉnh V được phân hoạch thành k miền $D = \{D_1, D_2, \dots, D_k\}$;
Đỉnh nguồn $s \in V$; Đỉnh đích $t \in V$;
Output: Lời giải của bài toán IDPC-NDU;

```

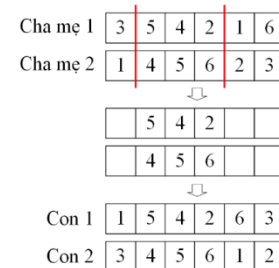
1 begin
2    $P_t \leftarrow$  Khởi tạo ngẫu nhiên  $N$  cá thể;
3   foreach cá thể  $ind_j \in P_0$  do
4     Tạo đồ thị  $G'_{ind_j}$  từ các thể  $ind_j$ .
5     Tạo lời giải  $s_j$  của bài toán IDPC-NDU bằng cách áp dụng thuật toán Dijkstra cho đồ
      thị  $G'_{ind_j}$  để tìm được đi ngắn nhất từ đỉnh nguồn  $s$  tới đỉnh đích  $t$ .;
6     Tính chi phí của lời giải  $s_j$ ;
7     Tính giá trị thích nghi của cá thể  $ind_j$  dựa trên chi phí lời giải  $s_j$ ;
8    $t \leftarrow 0$ ;
9   while điều kiện dừng chưa thỏa mãn do
10     $O_t \leftarrow \emptyset$ ;
11     $P'_t \leftarrow$  Tournament Selection( $P_t$ ) ▷ Chọn các cá thể cha mẹ;
12     $O_t \leftarrow$  Thực hiện lai ghép và đột biến ( $P'_t$ );
13    foreach cá thể  $c_j \in O_t$  do
14      Tạo đồ thị  $G'_{c_j}$  từ các thể  $c_j$ .
15      Tạo lời giải  $s'_j$  của bài toán IDPC-NDU bằng cách áp dụng thuật toán Dijkstra cho
        đồ thị  $G'_{c_j}$  để tìm được đi ngắn nhất từ đỉnh nguồn  $s$  tới đỉnh đích  $t$ .;
16      Tính chi phí của lời giải  $s'_j$ ;
17      Tính giá trị thích nghi của  $c_j$  dựa trên chi phí của  $s'_j$  ▷ Xem phần 3.5
18     $b \leftarrow$  Cá thể tốt nhất trong tập cá thể con  $O_t$ ;
19     $b' \leftarrow$  VNS( $b$ );
20    Thay thế  $b$  bằng cá thể  $b'$  trong tập  $O_t$ ;
21     $R_t \leftarrow O_t \cup P_t$ ;
22     $P_{t+1} \leftarrow$  Chọn  $N$  cá thể tốt nhất trong tập  $R_t$ ;
23     $t \leftarrow t + 1$ ;
24  return Lời giải của tốt nhất của bài toán IDPC-NDU;

```

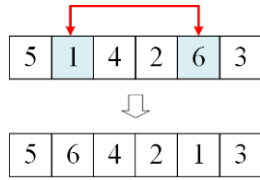
Hình 3. Lược đồ các bước chính của thuật toán *P-MEM* giải bài toán IDPC-NDU

3.4. Toán tử lai ghép và đột biến

Do sử dụng phương pháp mã hóa nên thuật toán *P-MEM* có thể sử dụng các toán tử tiến hóa cho biểu diễn hoán vị. Thuật toán *P-MEM* sử dụng toán tử lai ghép thứ tự (Order Crossover - OX) [8] và toán tử đột biến đổi vị trí (Swapping Mutation - SM) [8].



Hình 4. Minh họa phép lai ghép OX

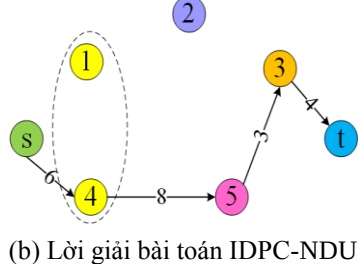
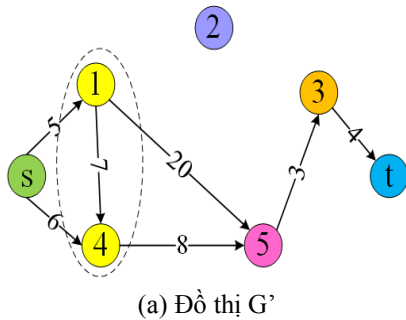


Hình 5. Minh họa phép đột biến SM

3.5. Đánh giá cá thể

Do mỗi cá thể trong thuật toán *P-MEM* là một hoán vị của tập các miền nên để đánh giá cá thể *ind* cần thực hiện:

- **Bước 1:** Xác định thứ tự của các miền dựa theo thứ tự của các gen trên cá thể *ind*.
- **Bước 2:** Tạo đồ thị có hướng G' dựa trên đồ thị đầu vào G và thứ tự các miền trong Bước 1 như sau:
 - + Tập đỉnh của G' trùng với tập đỉnh của G .
 - + Cạnh liên miền bao gồm các cạnh nối miền ind_i tới miền ind_{i+1} hoặc từ miền ind_k tới miền ind_1 .
 - + Cạnh trong mỗi miền của G' trùng với cạnh trong mỗi miền của G .
- **Bước 3:** Xây dựng lời giải *sol* của bài toán *IDPC-NDU* từ cá thể *ind*.
- **Bước 4:** Áp dụng thuật toán Dijkstra [9], [10] để tìm được đi ngắn nhất từ đỉnh nguồn tới đỉnh đích. Độ dài đường đi ngắn nhất tìm được chính là chi phí của cá thể *ind*.



Hình 6. Minh họa phương pháp đánh giá cá thể trong thuật toán *P-MEM*

Hình 6 minh họa phương pháp đánh giá cá thể sử dụng trong thuật toán *P-MEM*. Hình 6(a) minh họa đồ thị G' được tạo với đồ thị đầu vào như trong Hình 1 và cá thể *ind* như trong Hình 6(b) minh họa lời giải bài toán *IDPC-NDU* tìm được sau khi áp dụng thuật toán Dijkstra tìm đường đi ngắn nhất từ đỉnh *s* tới đỉnh *t* trên đồ thị trong Hình 6(a).

3.6. Thuật toán tìm kiếm biến đổi lân cận

Nghiên cứu này sử dụng thuật toán tìm kiếm cục bộ là thuật toán biến đổi lân cận [6] với cấu trúc lân cận là 2-opt [6], [11], [12]. Các bước chính của thuật toán tìm kiếm biến đổi lân cận được trình bày trong 0

4. KẾT QUẢ THỰC NGHIỆM

4.1. Dữ liệu thực nghiệm và tiêu chí đánh giá

Nghiên cứu chọn hai tập dữ liệu (có tên là Type 1, Type 2) được sử dụng trong các nghiên cứu trước, để đánh giá hiệu quả của thuật toán *P-MEM*. Mỗi tập Type 1 và Type 2 lại được chia thành các tập dữ liệu nhỏ hơn Type 1 Small, Type 1 Large, Type 2 Small và Type 2 Large). Bảng mô tả thông tin về tập dữ liệu sử dụng để đánh giá thuật toán *P-MEM*.

Bảng 1. Bảng mô tả thông tin về tập dữ liệu sử dụng để đánh giá thuật toán *P-MEM*

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

Thông tin tóm tắt về các tập dữ liệu này được mô tả trong Bảng 1. Các bộ dữ liệu thực nghiệm được lưu trữ tại [13].

Chất lượng lời giải tìm được của các thuật toán được đánh giá theo các tiêu chí: giá trị tốt nhất tìm được trong các lần thực hiện (*BF*), giá trị trung bình cộng (*Avg*) và độ lệch tiêu chuẩn (*Std*).

Input:	- Đa đồ thị có hướng, có trọng số $G = (V, E, w, D)$;
	- Đỉnh nguồn $s \in V$; Đỉnh đích $t \in V$;
	- Cá thể x ;
	- Các mức của một lân cận $N_l, l = 1, 2, \dots, l_{max} (l_{max} > 1)$
Output:	Cá thể x^* ;

```

1 begin
2    $t \leftarrow 0$ ;
3   repeat
4      $l \leftarrow 1$ ;
5     repeat
6       Tạo điểm  $x'$  ngẫu nhiên từ lân cận mức  $k$  của  $x$ ;
7        $x'' \leftarrow$  Áp dụng tìm kiếm cục bộ với  $x'$  là điểm bắt đầu;
8       if  $x''$  tốt hơn  $x$  then
9          $x \leftarrow x''$ ;
10         $l \leftarrow 1$ ;
11      else
12         $l \leftarrow l + 1$ ;
13    until  $l = l_{max}$ ;
14  until điều kiện dừng được thỏa mãn;
15  return  $x$ ;

```

Hình 7. Các bước chính của thuật toán tìm kiếm biến đổi lân cận VNS

4.2. Môi trường tiến hành thực nghiệm và tham số thực nghiệm

Để đánh giá thuật toán $P-MEM$, các tác giả so sánh lời giải tìm được bằng thuật toán $P-MEM$ với lời giải tìm được bởi một trong các thuật toán tiến hóa mới được nghiên cứu trong công bố [3] (ký hiệu $dmFEA$) và kết quả tối ưu (ký hiệu Opt).

Các tham số thực nghiệm của thuật toán $P-MEM$ gồm: kích thước quần thể: 100; tỉ lệ đột biến: 0,05; tỉ lệ lai ghép: 0,9; thuật toán sẽ dừng khi đạt số lần đánh giá là 50.000.

Tác giả sử dụng ngôn ngữ lập trình C# để cài đặt thuật toán $P-MEM$. Thực nghiệm được tiến hành trên máy tính có cấu hình là CPU: Intel Xeon E5620, RAM: 8GB. Máy tính sử dụng hệ điều hành MS Windows 10 Home (64 bit). Thuật toán $P-MEM$ được thực nghiệm 30 lần trên mỗi bộ dữ liệu.

4.3. Kết quả thực nghiệm

Ovà trình bày kết quả so sánh của thuật toán $P-MEM$ với các thuật toán $dmFEA$ và kết quả tối ưu. Trong đó, tên các bộ dữ liệu chỉ hiển thị phần cuối, phần đầu “idpc_ndu_” được bỏ đi để tiết kiệm không gian bảng.

Bảng 2. Bảng so sánh lời giải của P-MEM, dmFEA và kết quả tối ưu trên tập dữ liệu Type 1.

	Bộ dữ liệu	dmFEA		P-MEM		Opt
		BF	Avg	BF	Avg	
Kích thước nhỏ (small)	1002_12_82252	14,0	14,0	8	13,8	8
	1002_22_36564	35,0	38,2	30	37,0	18
	1192_19_37744	40,0	53,4	40	51,0	18
	1202_22_65521	22,0	22,0	22	22,1	16
	1256_21_44446	43,0	54,5	42	48,5	26
	1322_22_48821	44,0	45,7	44	44,7	26
	1506_9_130556	11,0	16,6	11	12,6	11
	1514_16_78292	33,0	33,4	33	33,1	21
	1514_30_78351	30,0	34,8	30	31,7	21
	1602_22_60574	46,0	51,7	46	48,9	24
	1682_23_67612	46,0	74,2	46	61,1	24
	1730_20_88509	39,0	41,9	39	41,4	24
	2002_22_178026	24,0	24,0	24	24,0	16
	427_7_14927	8,0	8,2	8	8,0	8
	704_15_16990	31,0	33,2	31	31,3	21
	842_23_31617	22,0	22,0	22	22,1	16
Kích thước lớn (large)	2102_23_164811	27,0	27,0	27	27,1	18
	2402_22_260967	24,0	24,1	24	24,0	16
	2494_12_276620	23,0	23,0	23	23,0	16
	2502_22_229601	29,0	30,1	29	29,5	18
	2522_20_171940	41,0	41,3	41	41,6	18
	2715_22_592246	13,0	13,0	13	13,0	8
	2918_29_293109	30,0	30,2	30	30,6	21
	3161_15_339881	30,0	30,0	30	30,0	21
	3602_32_406192	35,0	49,8	35	45,6	21

Bảng 3. Bảng so sánh lời giải của P-MEM, dMFEA và kết quả tối ưu trên tập dữ liệu Type 2.

	Bộ dữ liệu	dMFEA		P-MEM		Opt
		BF	Avg	BF	Avg	
Kích thước nhỏ (small)	1002_32_60942	19,0	22,8	19	24,8	13
	102_10_834	7,0	8,0	7	7,0	7
	1102_17_56280	25,0	27,0	25	26,2	14
	1202_18_68002	24,0	24,7	24	25,7	15
	1302_22_78953	25,0	27,8	26	26,4	17
	1402_24_92365	24,0	28,8	24	27,0	16
	1502_27_104878	27,0	33,2	27	30,7	16
	152_14_1869	16,0	16,0	8	15,5	8
	1602_14_96765	32,0	32,9	17	30,7	17
	1702_18_110993	20,0	24,4	20	29,8	20
	1802_21_123666	33,0	43,7	33	36,3	21
	1902_27_137407	30,0	47,9	30	35,8	21
	202_22_2341	20,0	32,4	20	26,6	9
	252_11_3513	11,0	11,0	11	17,5	11
	302_12_4930	11,0	24,6	11	16,3	11
	352_17_6667	26,0	30,4	13	29,5	13
	402_22_8220	26,0	32,3	26	29,4	13
	452_32_10406	22,0	30,6	22	29,0	13
	502_12_10949	11,0	27,2	11	18,2	7
	52_6_204	6,0	6,0	6	6,0	6
Kích thước lớn (large)	2002_17_128021	37,0	43,0	37	41,4	16
	2102_19_141155	36,0	37,5	36	37,0	20
	2202_22_153764	36,0	38,3	36	37,4	21
	2302_27_169859	42,0	62,8	42	56,5	22
	2402_29_186438	37,0	55,7	38	51,9	23
	2502_32_201852	34,0	75,7	34	59,2	25
	2602_19_185818	41,0	54,8	42	48,4	21
	2802_30_215589	59,0	83,4	50	80,6	25

Bên cạnh đó, trong hai bảng này, tại mỗi dòng của thuật toán dMFEA và P-MEM, lời giải có giá trị tốt hơn thuật toán còn lại được in nghiêng; lời giải có giá trị bằng giá trị tối ưu được in đậm ở cột giá trị tối ưu.

Kết quả của các thuật toán tại các bảng trên cho thấy thuật toán P-MEM tốt hơn thuật toán dMFEA trên 39 trong tổng số 53 trường hợp. Đặc biệt, thuật toán P-MEM tốt hơn dMFEA trên tất cả bộ dữ liệu thuộc tập Type 2 large. Thuật toán dMFEA chỉ tốt hơn thuật toán P-MEM trên 9 bộ dữ liệu. Cụ thể hơn:

- Thuật toán P-MEM có kết quả trội hơn thuật toán dMFEA trên 14 trong tổng số 16 bộ dữ liệu thuộc Type 1 small. Trong khi chỉ tốt hơn 4 trên tổng số 9 bộ dữ liệu thuộc Type 2 large.

- Thuật toán P-MEM có kết quả trội hơn thuật toán dMFEA trên 14 trong tổng số 20 bộ dữ liệu thuộc Type 2 small.

Thuật toán P-MEM có 11 trường hợp tìm được lời giải có giá trị tối ưu. Các lời giải có giá trị tối ưu thuộc các bộ dữ liệu có số miền nhỏ và đều thuộc các bộ dữ liệu thuộc tập Type 1 small và Type 2 small. Điều đó có nghĩa là P-MEM hiệu quả hơn với các bộ dữ liệu có số miền bé. Nguyên nhân của kết quả này là do P-MEM mã hóa mỗi cá thể là một hoán vị của số miền.

5. KẾT LUẬN

Bài toán IDPC-NDU là một trong những bài toán mới được nghiên cứu trong tối ưu mạng đa miền. Bài toán IDPC-NDU có nhiều ứng dụng trong cả nghiên cứu lý thuyết và ứng dụng thực tiễn. Nghiên cứu này mô tả cách kết hợp giữa thuật toán EA và thuật toán VNS để giải bài toán IDPC-NDU, trong đó sử dụng thuật toán VNS vào giải bài toán IDPC-NDU. Kết quả thực nghiệm đã bước đầu chỉ ra tính hiệu quả của hướng kết hợp giữa hai thuật toán.

Trong thời gian tới, các tác giả sẽ tiếp tục cải thiện hiệu quả của việc kết hợp giữa thuật toán EA và thuật toán tìm kiếm cục bộ khác.

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

- [1] F. Paolucci, F. Cugini, A. Giorgetti, N. Sambo, and P. Castoldi, "A survey on the path computation element (PCE) architecture," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1819–1841, 2013.
- [2] 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.
- [3] P. D. Thanh, "An adaptive multifactorial evolutionary algorithm for Inter-domain path computation under node-defined domain uniqueness constraint," *TNU Journal of Science and Technology*, vol. 227, no. 08, pp. 114–122, 2022.
- [3] P. D. Thanh, "An adaptive multifactorial evolutionary algorithm for Inter-domain path computation under node-defined domain uniqueness constraint," *TNU Journal of Science and Technology*, vol. 227, no. 08, pp. 114–122, 2022.

- [4] H. T. T. Binh, N. H. Long, T. B. Thang, 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.
- [5] A. Do Tuan, L. N. Hoang, T. T. Bao, H. T. T. Binh, 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, pp. 687–700.
- [6] P. Hansen and N. Mladenović, "Variable neighborhood search methods," in *Encyclopedia of Optimization*, C. A. Floudas and P. M. Pardalos, Eds. Springer US, 2009, pp. 3975–3989. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-74759-0_694
- [7] 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.
- [8] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*, vol. 53. Springer, 2003.
- [9] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [10] J.-C. Chen, "Dijkstra's shortest path algorithm," *Journal of Formalized Mathematics*, vol. 15, no. 9, pp. 237–247, 2003.
- [11] P. Hansen and N. Mladenović, "Variable neighborhood search: Principles and applications," *European Journal of Operational Research*, vol. 130, no. 3, pp. 449–467, May 2001, doi: 10.1016/S0377-2217(00)00100-4.
- [12] P. Hansen, N. Mladenović, and José A. Moreno Pérez, "Variable neighbourhood search: methods and applications," *4OR*, vol. 6, no. 4, pp. 319–360, Dec. 2008, doi: 10.1007/s10288-008-0089-1.
- [13] T. Pham Dinh, T. Ta Bao, H. Ngo Viet, and A. Do Tuan, "Inter-Domain Path Computation under Node-defined Domain Uniqueness Constraint Instances," *Mendeley Data*, vol. 2, 2021, doi: 10.17632/tpg2nbcsc5.2.

A COMBINATION OF EVOLUTIONARY ALGORITHM AND VARIABLE NEIGHBOURHOOD SEARCH FOR INTER-DOMAIN PATH COMPUTATION WITH A NODE-DEFINED DOMAIN UNIQUENESS CONSTRAINT

Pham Dinh Thanh, Nguyen Duy Hieu, Dang Thi Van Chi, Phan Trung Kien
Tay Bac University

Abstract: *There has recently been an increasing interest in conducting a research on the inter-domain path computation under node-defined domain uniqueness constraint (IDPC-NDU). Although there have been some studies using evolutionary algorithm (EA) to solve the IDPC-NDU problem, there are a few studies on the combination of EA and a local search algorithm. Therefore, this study was carried out to describes a combination of EA and variable neighbourhood search (VNS). In order to apply the VNS to solve the IDPC-NDU, the study also provided a description of a solution encoding method for the IDPC-NDU based on the permutation representation. The proposed algorithm was evaluated on the datasets with 53 different instances. The findings showed that the proposed algorithm is better than the counterparts in 39 cases.*

Keywords: *Evolutionary algorithm, variable neighbourhood search, inter-domain path computation under node-defined domain uniqueness constraints.*

Ngày nhận bài: 03/02/2023. Ngày nhận đăng: 05/03/2023
Liên lạc: Phạm Đình Thành, e-mail: thanhpd@utb.edu.vn