

PHÁT TRIỂN GIẢI THUẬT LAI CÓ SỬ DỤNG HỌC MÁY ĐỂ GIẢI BÀI TOÁN ĐỊNH TUYẾN XE VEHICLE ROUTING PROBLEMS (VRP)

● Nguyễn Minh Đế* ● Lê Văn Hạnh
Trường Đại học Quốc Tế Hồng Bàng

TÓM TẮT

Sự phát triển trong lĩnh vực Trí tuệ nhân tạo Artificial Intelligence đã cung cấp các kỹ thuật mạnh mẽ để giải quyết bài toán Vehicle Routing Problems (VRP) và các biến thể của nó. Bài báo này đề xuất kết hợp kỹ thuật Học máy Machine Learning ML với một giải thuật lai để giải quyết bài toán VRP. Giải thuật lai này là sự phối hợp giữa giải thuật Tối ưu bầy đàn PSO và giải thuật Di truyền Genetic Algorithm GA. Việc tận dụng Học máy được thực hiện theo hai bước, và hai bước này được thực thi với danh sách các khách hàng như sau: Một mô hình phân lớp để dự đoán số lượng xe cần thiết cho từng tập khách hàng được xác định bằng kỹ thuật Cây phân lớp tối ưu Optimal Classification Trees OCT; một giải thuật phân cụm mà có thể tận dụng được các tri thức có được từ sự phân lớp để có thể tối thiểu hóa số lượng xe vận tải cần để phục vụ học tập các khách hàng. Các kết quả thực nghiệm cho thấy rằng giải thuật lai đã thực thi với tập dữ liệu khách hàng đầu vào là nhỏ hơn từ 1% đến 5% và đã giảm tải xử lý cho giải thuật lai.

Từ khóa: bài toán lập lộ trình/định tuyến xe, giải thuật di truyền, Tối ưu bầy đàn; Học máy, giải thuật tiến hóa

DEVELOPING A HYBRID ALGORITHM USING MACHINE LEARNING TO SOLVE VEHICLE ROUTING PROBLEMS

● Nguyen Minh De ● Le Van Hanh

ABSTRACT

The development of Artificial Intelligence AI field has provided powerful techniques to solve the Vehicle Routing Problems VRP problems and its variants. In this paper, a combination of Machine Learning ML technique with a hybrid algorithm is proposed to solve the VRP problem, which this hybrid algorithm is obtained from the combination of the Particle Swarm Optimization PSO and the Genetic Algorithm GA. Leveraging Machine Learning is done in two steps, and these two steps are executed with the customer list as follows: A classification model predicts the required number of vehicle for each set of customers determined by the Optimal Classification Trees OCT; A clustering algorithm which can take advantage of the knowledge gained from classification to minimize the number of vehicle required to serve a set of customers. Experimental results show that the hybrid algorithm implemented with the input customer data set is 1% to 5% smaller and has reduced the processing load for the hybrid algorithm.

Keywords: vehicle Routing Problems VRP, Genetic Algorithm GA, Particle Swarm Optimization PSO, Machine Learning ML, Heuristic

* Tác giả liên hệ: ThS. Nguyễn Minh Đế, Email: denml@hiu.vn

(Ngày nhận bài: 10/10/2022; Ngày nhận bản sửa: 11/11/2022; Ngày duyệt đăng: 16/11/2022)

1. TỔNG QUAN

Bài toán định tuyến xe VRP là bài toán được chính thức giới thiệu vào năm 1950 bởi G. B. Dantzig và J. H. Ramser [1]. Kể từ đó đã có rất nhiều công trình mở rộng cho dạng bài toán VRP và nhiều công trình cố gắng giải quyết chúng [2],[3]. Bài toán định tuyến xe VRP là bài toán thuộc nhóm bài toán tổ hợp Combinatorial Optimization (CO), và còn là bài toán dạng Nondeterministic Polynomial time (NP) khó [2], do đó việc giải quyết bài toán là vấn đề thách thức rất lớn. Về tổng quát thì bài toán VRP nhằm đến việc xây dựng một tập lộ trình (LT) giao hàng có chi phí thấp nhất từ kho đến một nhóm khách hàng phân tán theo địa lý do một đội xe thực hiện, và chịu các ràng buộc xác định từ trước. Việc giải bài toán VRP thông thường được giải tổng quát bằng các nhóm giải thuật chính xác và giải thuật heuristic (gồm mở rộng metaheuristic), và khoảng thời gian vài thập niên gần đây là nhóm giải thuật được đề xuất là nhóm giải thuật lai [4]. Các công trình nghiên cứu gần đây cũng phân loại các nhóm phương pháp giải dạng bài toán VRP theo 3 nhóm phương pháp: Chính xác (exact); Heuristic (gồm metaheuristic); Lai (hybrid).

Kỹ thuật của ba nhóm phương pháp ở trên để giải bài toán VRP có hiệu quả khác nhau. Nhóm phương pháp chính xác có hiệu quả cho đến khách hàng thứ 50, và đến kho thứ 50, mà có thể phân vào 3 loại: Tìm kiếm trên cây trực tiếp; Quy hoạch động; Quy hoạch nguyên và tuyến tính. Nhóm phương pháp Heuristic (cũ) cung cấp các thủ tục để tìm các giải pháp chấp nhận được thông qua một tập đã phát hiện thấy có giới hạn của không gian tìm kiếm. Các kỹ thuật của Nhóm phương pháp Heuristic (cũ) gồm: Chèn Insertion; Ngoại trừ Savings; Quét dây Sweep; Hai bước Two Phase. Nhóm phương pháp Heuristic mới (metaheuristic) gồm: Tabu Search Tìm kiếm Tabu; Genetic Di truyền; Iterated Local Search Tìm địa phương lặp lại; Simulated Annealing Mô phỏng luyện kim; Variable Neighborhood Search Tìm lân cận tùy biến; Ant Colony Cột đàn kiến; Neural network Mạng nơ-ron; Artificial Bee Colony Bầy ong nhân tạo; Particle Swarm Optimization Tối ưu bầy đàn. Nhóm này đã cho ra các kết quả tốt hơn so với các heuristic cũ. Nhóm phương pháp giải thuật lai được phát triển vài thập niên gần đây đã cho thấy sự đơn giản với tốc độ tối ưu hóa nhanh và tính toán ít do nhóm giải thuật này đã kết hợp giải thuật chính xác vào các heuristic. Một trong số các heuristic được quan tâm gần đây để giải quyết bài toán VRP giải thuật PSO, hơn nữa PSO thường sẽ kết hợp với một số giải thuật meta-heuristic khác (như GA) để tạo nên một giải thuật lai có hiệu quả cao hoặc là đưa vào chính bản thân các giải thuật chính xác để tạo thành giải thuật lai [5 - 11].

Bài báo này trình bày các bước phân tích và thiết kế để xây dựng giải thuật lai GWPSO từ việc kết hợp giải thuật di truyền GA vào giải thuật PSO, sau đó ước tính số lượng xe tối thiểu cho từng tập dữ liệu khách hàng theo giải thuật CFA có từ phương pháp Học máy ML. Phần còn lại của bài báo như sau: Phần 2, phần kế tiếp trình bày các dạng mở rộng của bài toán VRP; Phần 3, trình bày Giải thuật lai đề nghị để giải quyết bài toán VRP; Phần 4, trình bày kỹ thuật máy học và việc tích hợp nó vào giải thuật lai nói trên để giải quyết bài toán VRP; Phần 5, trình bày cài đặt, kết quả và thảo luận.

2. BÀI TOÁN VRP VÀ MỘT SỐ DẠNG MỞ RỘNG

Khai báo bài toán [12]: Cho $G(V, E)$ là một đồ thị vô hướng có trọng số, V đại diện cho các địa điểm (đỉnh) và các cạnh trong E đại diện cho đường đi giữa một cặp địa điểm. Đi kèm với từng đường đi $e \in E$ là một trọng số không âm $a(e)$ thể hiện khoảng cách giữa hai đỉnh.

Cho $O \subseteq V$ đại diện cho tập kho hàng, mỗi kho $x \in O$ lúc đầu chứa $m(x)$ xe. Cho $C \subseteq V$ là tập khách hàng, mỗi $x \in C$ có nhu cầu $d(x) \geq 0$, nhu cầu là số đơn vị hàng hóa mà khách hàng có yêu cầu. Mục đích bài toán là thiết kế một tập lộ trình tối ưu và/hoặc lịch trình cho từng xe theo thứ tự để thỏa mãn tất cả các yêu cầu của khách hàng. Nhóm bài toán VRP có nhiều dạng khác nhau, trong khuôn khổ bài báo sẽ trình bày các dạng bài toán VRP như sau:

a. *Bài toán VRP (góc)*: có nhiều xe bắt đầu ở một kho trung tâm để phục vụ khách hàng. Mỗi xe có tốc độ khác nhau. Mục đích bài toán là tối thiểu tổng thời gian để đến từng khách hàng có yêu cầu. Nếu thay thế yêu cầu về thời gian thành tổng chi phí (khoảng cách) ít nhất thì ta có bài toán người bán hàng đi du lịch - Traveling Salesman Problem (TSP).

b. *Bài toán CVRP*: là bài toán VRP, có nhiều xe được dán nhãn từ: 1, 2, ..., m, khởi hành từ một kho trung tâm để phục vụ khách hàng. Xe $i \in \{1, 2, \dots, m\}$ có sức tải $w(i)$, từng khách hàng $x \in C$ có nhu cầu $d(x)$.

Các xe được đặt ở một kho trung tâm và sẽ đi đến các khách hàng nằm rải rác khắp nơi trên các vị trí và sẽ đi đến các vị trí theo trình tự trước sau để phục vụ từng khách hàng. Bài toán cần xây dựng một tập lộ trình hợp lý, chi phí thấp – dành cho từng xe một. Một lộ trình là thứ tự của các vị trí mà một xe phải đến với số lượng hàng được yêu cầu nó phải cung cấp. Xe phải bắt đầu và kết thúc lộ trình của nó tại kho trung tâm.

Mục đích bài toán là tìm ra lộ trình xe ngắn nhất (tối thiểu tổng độ dài của tất cả các lộ trình xe) mà tất cả khách hàng có yêu cầu phục vụ phải được thỏa mãn; hoặc tìm ra lộ trình nhanh nhất (tối thiểu lộ trình dài nhất cho tất cả các lộ trình xe).

Như vậy, phải xây dựng một tập có tới m lộ trình mà: (i) từng lộ trình bắt đầu và kết thúc tại kho; (ii) tất cả nhu cầu phải được thỏa mãn; (iii) sức tải của các xe không bị vượt tải; (iv) một khách hàng chỉ được duy nhất một xe đi đến; (v) tổng chi phí phải tối thiểu hóa.

c. *Bài toán VRPTW*: là bài toán VRP, có cấu hình tương tự CVRP với thêm ràng buộc là từng yêu cầu khách hàng phải được phục vụ trong một khung thời gian (từng khách hàng phải được đi đến trong một khoảng thời gian xác định, được gọi là khung thời gian).

Mục đích bài toán là thiết kế một tập lộ trình tối ưu và/hoặc lịch trình cho từng xe theo thứ tự để thỏa mãn tất cả các yêu cầu của khách hàng.

3. XÂY DỰNG GIẢI THUẬT LAI GWPSO

3.1. Xây dựng và phát triển giải thuật di truyền GA

a. *Giải thuật GA cơ bản*: Xét giải thuật GA cơ bản có các quá trình sau [16]:

i. *Mã hóa dữ liệu*: hay còn gọi là biểu diễn di truyền cho lời giải của bài toán

ii. *Khởi tạo quần thể* (xây dựng tập hợp nghiệm ban đầu): có thể ngẫu nhiên hoặc không ngẫu nhiên

iii. *Xác định hàm thích nghi*: hay hàm lượng giá cho mỗi nhiễm sắc thể hay chính là cho các phương án nghiệm trong tập nghiệm. Hàm này dùng để đánh giá độ thích nghi của các nhiễm sắc thể.

iv. *Quá trình lai ghép*: đây là quá trình nhiễm sắc thể mới được hình thành dựa trên nhiễm sắc thể cha mẹ bằng cách lai ghép một hay nhiều đoạn nhiễm sắc thể cha mẹ với nhau. Phép lai ghép xảy ra với xác suất là p1 có thể được mô phỏng như sau: i) Chọn hai (hay nhiều) cá thể bất kỳ trong quần thể. Quần thể ở đây bao gồm các nhiễm sắc thể (cha mẹ) có độ dài bằng nhau. ii) Chọn điểm lai là một điểm có vị trí bất kỳ (như nhau) trên nhiễm sắc thể cha-mẹ và thực hiện hoán đổi các đoạn gen của nhiễm sắc thể cha mẹ tại điểm lai này. iii) Đưa hai cá thể này vào quần thể để thực hiện vào các quá trình tiến hóa tiếp theo.

Một số phép lai cải tiến như sau [16]: i) Lai ghép có xét tới các đặc tính trội và lặn trong tự nhiên (Các đặc tính này được quy định trước trong khi biểu diễn cấu trúc nhiễm sắc thể); ii) Lai ghép từng phần (Việc giữ lại những đoạn mã đã “tối ưu” trong nhiễm sắc thể cũng là một cách để quá trình lai ghép trở nên hiệu quả hơn); Lai ghép có trật tự; Lai ghép dựa trên vị trí; Lai ghép chu trình; Lai

ghép thứ tự tuyến tính; Lai ghép đa điểm (có thể cho 2 cá thể lai ghép ở 2 hay nhiều điểm lai ghép).

v. *Quá trình đột biến*: là quá trình cá thể con mang một hay một số tính trạng không có trong mã di truyền của cha mẹ. Quá trình này xảy ra với xác suất p_2 (nhỏ hơn nhiều so với p_1) có thể được mô tả như sau: i) Chọn ngẫu nhiên một cá thể bất kỳ trong quần thể; ii) Chọn một gen bất kỳ của cá thể vừa chọn; iii) Thay đổi giá trị gen đó, rồi trả về quần thể để thực hiện các quá trình tiếp theo

Nhiều cách thức [18] để thực hiện quá trình gây đột biến ngày càng hiệu quả hơn: i) Đột biến đảo ngược (Inversion Mutation); ii) Đột biến chèn (Insertion Mutation); iii) Đột biến thay thế (Replacement Mutation); iv) Đột biến tương hỗ (Reciprocal Exchange Mutation); v) Đột biến dịch chuyển (Shift Mutation).

vi. *Quá trình chọn lọc*: các cá thể mới sinh ra được giữ lại hay bị loại bỏ khỏi quần thể dựa vào độ thích nghi của chúng. Độ thích nghi ở đây thường là một hàm gán một giá trị thực cho các cá thể trong quần thể.

b. Giải thuật GA có chỉnh sửa

Trong 6 quá trình của GA trên thì có 3 quá trình (2) Khởi tạo quần thể, (4) Quá trình lai ghép, và (5) Quá trình đột biến là có thể thiết kế lại theo công trình [18] để giải quyết các bài toán VRP, cụ thể như sau:

- Đối với quá trình (2) Khởi tạo quần thể thì có một thủ tục chuyên biệt *Khởi_Tạo_Quần_Thể*(.), thủ tục này tạo các cá thể không phải hoàn toàn ngẫu nhiên mà có đưa vào một số kỹ thuật (mạng nơ-ron, nguyên lý tham lam, leo đồi,...) và sử dụng thêm toán tử Lựa chọn TS.

- Đối với quá trình (4) Quá trình lai ghép thì sử dụng 2 toán tử lai ghép là Sinusoidal Motion Crossover và *Hai_Con_Lech* (nghiên cứu này đề xuất thêm một toán tử lai ghép riêng). Toán tử *Hai_Con_Lech* có đầu vào là 2 cá thể cha mẹ và cho ra 2 cá thể con (đặc thù là có sự chênh lệch về sự thích nghi của chúng). Ngoài ra quá trình này có sử dụng thêm toán tử Lai ghép từ công trình [18] đề nghị là toán tử OX để thực thi.

- Đối với (5) Quá trình đột biến thì đã có nhiều cách thức để thực hiện quá trình gây đột biến ngày càng hiệu quả hơn: i) Đột biến đảo ngược (Inversion Mutation); ii) Đột biến chèn (Insertion Mutation); iii) Đột biến thay thế (Replacement Mutation); iv) Đột biến tương hỗ (Reciprocal Exchange Mutation); v) Đột biến dịch chuyển (Shift Mutation). Trong bài báo này thì đề nghị sử dụng 2 toán tử đột biến là: Toán tử EM do [18] đề nghị; và Toán tử *Phoi_Hop*. Toán tử *Phoi_Hop* là kết hợp của các toán tử (Inversion Mutation, Insertion Mutation, Replacement Mutation, Reciprocal Exchange Mutation, Shift Mutation).

Kết quả của quá trình này là tạo ra một giải thuật lai là sự kết hợp giải thuật di truyền đã chỉnh lại và một số kỹ thuật chính xác như đã trình bày ở trên, và có tên là Mix Genetic Algorithm MGA.

Giải thuật MGA

Giải thuật MGA

Procedure MGA

Input: tập tham số vào

Output: tập kết quả tìm được

1. Khởi tạo tham số, thủ tục và thành phần cần thiết
2. $PM \leftarrow Tao_ThamSo()$
3. $Q \leftarrow KhởiTạo_QuầnThe().$
4. $Q \leftarrow Q^{if} + Q^{fe} // Q^{if} : Q \text{ dưới ngưỡng} + Q^{fe} : Q \text{ trên ngưỡng}$

```

5. While <DieuKien_Lap()> // hết lặp (MaxIter) hoặc hết thời gian maxTime
    3.1. P ← chọn_ChaMe(.)
    3.2. CH ← Lai_Ghep(P)
    3.3. ILS_RVND_SP(CH)
    3.4. If ~HieuQua(CH) then Chen(CH, Qij); CaiTao(CH);
        Else Chen(CH, Qib);
    3.5. if SoLuong(Q) >= MaxQ then Loc(Q)
    3.6. CauHinh_ThamSO(PM);
    3.7. if ~KiemTra_TotNhat(Qi) then CaiTao_QuanThe(Q)
6. End While
7. Return Lay_TotNhat(Q)
8. end MGA.
    
```

3.2. Xây dựng và phát triển giải thuật PSO

a. Giải thuật PSO cơ bản:

Xét giải thuật PSO cải tiến gồm các bước được mô tả như sau:

i) *Khởi tạo*: Tạo một quần thể và đánh giá hàm mục tiêu (hàm thích nghi).

ii) *Cập nhật tốt nhất cục bộ (personal best) và tốt nhất toàn cục (global best)*: Xét mỗi phân tử để xác định vị trí tốt nhất cục bộ mới. Nếu vị trí hiện tại tốt hơn tốt nhất cục bộ, tốt nhất cục bộ sẽ là vị trí hiện tại. Nếu không, tốt nhất cục bộ vẫn được giữ nguyên. Nếu bất kỳ phân tử nào trong bầy đàn có vị trí tốt nhất cục bộ tốt hơn vị trí tốt nhất toàn cục, cá thể đó sẽ trở thành phân tử đầu đàn và vị trí tốt nhất cục bộ của nó sẽ trở thành tốt nhất toàn cục.

iii) *Cập nhật vận tốc và vị trí của tất cả các phân tử*: Vị trí và vận tốc ở thế hệ thứ t được cập nhật bởi các phương trình:

$$\begin{aligned}
 v_i(t) &= wv_i(t-1) + a_1u_d(p_i(t-1)) - x_i(t-1) + a_2U_d(g(t-1)) - x_i(t-1) \\
 x_i(t) &= x_i(t-1) + v_i(t) \Delta t
 \end{aligned}$$

Trong đó: v_i là vận tốc của phân tử thứ i; x_i là vị trí của phân tử trong không gian tìm kiếm; p_i là vị trí tốt nhất cục bộ mà phân tử đó chiếm giữ; g là vị trí tốt nhất toàn cục của một cá thể nào đó trong bầy đàn; u_d và U_d có giá trị ngẫu nhiên trong khoảng $[0, 1]$; w , a_1 , a_2 lần lượt là các tham số gia tốc, ảnh hưởng cá nhân và ảnh hưởng xã hội (tập thể).

iv) *Đột biến thích nghi (Adaptive mutation)*: Nhằm giúp các cá thể không bị dừng khi gặp cực trị cục bộ, vị trí của x_i sẽ bị đột biến một cách ngẫu nhiên và tăng dần tỷ lệ theo số vòng lặp như sau: Chọn ngẫu nhiên cá thể i, và số chiều j tại vòng lặp t và thực hiện:

$$\begin{aligned}
 x_{ij}^*(t) &= m * x_{ij}(t) * \text{rand}() \\
 x_{ij}^*(t) &= \begin{cases} x_{ij}^*(t) \text{ nếu } f(x_{ij}^*(t)) < f(x_{ij}(t)) \\ x_{ij}(t) \text{ nếu khác} \end{cases}
 \end{aligned}$$

Trong đó, $x_{ij}^*(t)$ là cá thể sau đột biến, m là hệ số thích nghi tăng dần theo số vòng lặp, rand() là hàm ngẫu nhiên trong dải % (cho trước hoặc xác định) của vùng tìm kiếm.

Chấm dứt quá trình tìm kiếm hoặc tiếp tục tìm kiếm: Quá trình tìm kiếm được dừng lại nếu: i) bước hiện tại tương đương với bước gần nhất hoặc ii) bầy đàn đã hội tụ (bán kính của bầy đàn nhỏ hơn % (cho trước hoặc xác định) của không gian tìm kiếm). Nếu không, quay trở lại bước 2.

b. Giải thuật PSO chỉnh sửa:

Trong bài báo này đề nghị thêm vào một số kỹ thuật để gia tăng khả năng tìm kiếm của PSO và duy trì được sự đa dạng của quần thể, cách làm này tương tự như các công trình [8 - 11]. Các kỹ thuật được đưa vào giải thuật lai này thuộc loại phương pháp tìm kiếm địa phương để thoát khỏi các tối ưu địa phương mà vẫn duy trì tính đa dạng của quần thể giải pháp.

Bài báo này cũng sử dụng lại kỹ thuật loại bỏ và chèn thông minh để tạo ra các giải pháp lân cận do [12] giới thiệu: Worse-Distance Removal (WDR); Worse-Time Removal (WTR); Greedy Insertion (GI); Best Time Insertion (BTI). Ngoài ra, nghiên cứu này cũng đề nghị thêm một kỹ thuật để loại bỏ và chèn thông minh là Time-Distance Removal (TDR) và Balance Insertion (BI).

Giải thuật 6RIPSO

Giải thuật 6RIPSO

Procedure 6RIPSO

Input: tập tham số vào

Output: tập kết quả tìm được

1. Khởi tạo tham số, thủ tục và thành phần cần thiết
 2. $Q \leftarrow \text{Khoi_Tao_Quan_The}()$
 3. $\text{Tinh_Gia_Tri_Thich_Ngh}(\text{Qi})$
 4. $\text{Luu_GP_Thich_Ngh}(\text{Qi})$
 5. $\text{Luu_Ca_The_Thich_Ngh}(\text{Q})$
 6. **While** <Điều kiện dừng> // đạt số lần lặp tối đa
 - 6.1. $\text{Toc_Do}(\text{Qi})$
 - 6.2. $\text{Toc_Do_Max}(\text{Qi})$
 - 6.3. $\text{Vi_Tri}(\text{Qi})$
 - 6.4. $\text{Vi_Tri_Gioi_Han}(\text{Qi})$
 - 6.5. $\text{Tinh_GT_Thich_Ngh}(\text{Qi})$
 - 6.6. $\text{CapNhat_GiaiPhap_XacSuat}(\text{Qi}, e^{E^T})$
 - 6.7. $\text{CapNhat_CaThe_TotNhat}(\text{Qi})$
 - 6.8. $\text{Chon_LoaiBo_ChenVao}()$
 - 6.9. $\text{Tinh_GT_Thich_Ngh}(\text{Qi})$
 - 6.10. $\text{CapNhat_GiaiPhap_XacSuat}(\text{Qi}, e^{E^T})$
 - 6.11. $\text{CapNhat_CaThe_TotNhat}(\text{Qi})$
 - 6.12. $\text{CapNhat_TrongLuong_QuanTinh}(\text{Qi})$
 7. **End while**
 8. **Return** s*
 9. **end** 6RIPSO.
-

Kết quả của các quá trình kết hợp ở trên là tạo ra một giải thuật lai là sự kết hợp giải thuật PSO đã chỉnh lại như ở trên và mang tên là 6RIPSO, như sau:

Xét PSO kết hợp với GA ở trên thì có một số bước đã tăng cường:

- Ngay tại 1 thì dùng kết quả của phần trên
- Sau bước 3 có thể thêm vào các phương pháp chính xác
- Sau bước 7.1 có thể thêm vào 1 bước để thực thi quá trình Đột biến
- Các bước 5, 6, 7 thì vận dụng giải thuật 6RIPSO

● Kết hợp giải thuật di truyền MGA và giải thuật tối ưu bầy đàn có trọng số 6RIPSO để tạo ra giải thuật có tên là GWPSO như sau:

Giải thuật GWPSO**Giải thuật GWPSO****Procedure GWPSO**

Input: tập tham số vào

Output: tập kết quả tìm được

1. Khởi tạo tham số, thủ tục và thành phần cần thiết
2. ThietLap_ThamSo()
3. $Q \leftarrow Khoi_Tao_Quan_The(.)$
4. **Do**
 - 4.1. MaHoa_Thuc_CaThe(Qi)
 - 4.2. Tinh_GT_Thich_Nghi(Qi)
 - 4.3. Tìm P_{best} và G_{best}
 - 4.4. Chon_LoaiBo_ChenVao()
 - 4.5. CapNhat_CaThe_TotNhat(Qi) // dùng hàm tuyến tính giảm
 - 4.6. **If** $n > n_{max}$ **or** GiaiPhap_Tot()
 - $CH \leftarrow T$
 - Else** Lai_Ghep(); Dot_Bien(); $CH \leftarrow F$
5. **Until** CH
6. **Return** s^*
7. **End** GWPSO.

4. KỸ THUẬT HỌC MÁY VÀ GIẢI THUẬT LAI**4.1. Kỹ thuật học máy**

Một số phương pháp được xây dựng từ việc kết hợp các kỹ thuật ML vào trong các heuristic đã được phát triển để giải quyết VRP và một số dạng liên quan. Các phương pháp này có thể được chia thành hai nhóm như sau:

- *Nhóm 1:* Giải bài toán VRP dựa trên các tập dữ liệu về khách hàng. Phương pháp trong nhóm này sử dụng dữ liệu quá khứ để dự đoán cho: Yêu cầu cấp hàng; Quá tải. Trong bài toán VRP có yêu cầu phân phối hàng cho khách hàng định kỳ theo thời gian cố định như 30 ngày. Các kết quả dự đoán sẽ dùng để tối ưu lộ trình tổng quát. Công trình Bosman and La Poutre (2006) [13] đã khai thác việc dự đoán nhu cầu để cải thiện hiệu suất cho giải thuật tiến hóa. Dự đoán yêu cầu được tính toán bằng mô hình xác suất để giải thuật tiến hóa xử lý các yêu cầu hiện tại và tương lai. Công trình [13] cho thấy các ích lợi của việc tích hợp các kết quả dự đoán vào các heuristic giải quyết VRP.

- *Nhóm 2:* Sử dụng các kỹ thuật của học Máy ML điều chỉnh và cải thiện cho các meta-heuristic. Phương pháp trong nhóm này đã sử dụng việc phân cụm để xác định các kết cấu giống nhau trong bài toán VRP với lý thuyết thông tin. Tiếp đến, các phương pháp này sẽ dự đoán các cụm với giải thuật di truyền GA, sau đó sẽ điều chỉnh việc lựa chọn các tác vụ tìm kiếm và tham số của bài toán VRP. Công trình Ventresca, Ombuki-Berman, and Runka (2013) [14] cũng đã sử dụng chiến lược ở trên để giải quyết bài toán VRPTW mà cho ra kết quả có hiệu suất xử lý tăng lên một phần nhỏ.

4.2. Áp dụng ML để phân lớp và dự đoán số xe

Bài toán học có quan sát là để tìm một hàm phân lớp y , mà gán cho từng điểm dữ liệu x giá trị là nhãn l thuộc một tập nhãn L giới hạn. Chất lượng của hàm phân lớp thường được đo bởi độ chính

xác từ tập kiểm thử, tập kiểm thử là tập các điểm dữ liệu mà thuộc về lớp dán nhãn đã xác định từ trước đó. Bài toán học có quan sát có các nhóm phương pháp xử lý như sau [15]: Phân lớp tuyến tính; Lân cận gần nhất; Mạng nơ-ron; Cây phân lớp; Xây dựng tập dữ liệu.

Nhiệm vụ học quan sát trong bài toán VRP ở đây là dự đoán được số xe cần thiết để có thể phục vụ được tập khách hàng có yêu cầu. Dữ liệu tập khách hàng ở đây được xem là tập dữ liệu thô nên cần được làm mịn lại theo khuôn mẫu riêng để thực hiện cho các tác vụ kế tiếp, các bước thực hiện:

a. Xây dựng tập dữ liệu mịn của khách hàng, từng dữ liệu của tập này cần phải có các thông tin như: Ô vuông bản đồ (có thể kích thước là 400x400); Kho ở trung tâm bản đồ; Khách hàng tại các vị trí khác nhau trên bản đồ; Tốc độ của các xe đều là 1 trong cùng một đơn vị khoảng cách trên bản đồ; Khung thời gian khách hàng theo đơn vị thời gian; Khoảng thời gian định kỳ; Nhu cầu của khách hàng; Thời gian dịch vụ cung cấp.

b. Tạo các nhãn, mỗi điểm dữ liệu trong tập dữ liệu đều tương ứng với mỗi tập khách hàng được dán nhãn cùng với số xe phục vụ. Việc thực hiện tạo các nhãn này là nhờ vào thuật toán được xây dựng riêng.

c. Tăng cường dữ liệu, để phát triển kích thước của tập dữ liệu mà với chi phí tính toán thấp thì giải thuật phải tính đến từng tổ hợp các xe trong một tập khách hàng cụ thể. Cho S là tập khách hàng, có 4 xe theo yêu cầu là a, b, c, d để phục vụ. Việc lập lộ trình cho tập S này với 4 xe được dán nhãn 4 sẽ bùng nổ việc tính toán cho một điểm dữ liệu, do đó để giảm việc tính toán thì sẽ xét đến việc dán nhãn như sau cho: Các xe a, b, c, d đều được gán nhãn 1; Xe a, b gán nhãn 2; Xe a, c gán nhãn 2; Xe a, d gán nhãn 2; Xe b, c gán nhãn 2; Xe b, d gán nhãn 2; Xe c, d gán nhãn 2; ...

Kỹ thuật OCT theo công trình [19] áp dụng phân lớp cho tập khách hàng có các đặc trưng như sau:

- Tại từng nút mới thì dùng hàm giá trị để xác định phân nhánh hay ngừng lại
- Nếu một nút được xác định là dừng thì sẽ gán nhãn cho nó là nút lá
- Nếu một nút được xác định là phân nhánh thì sẽ gán giá trị biến cho nó
- Trong quá trình phân lớp cho các điểm dữ liệu huấn luyện dựa trên cây đang được xây dựng thì phải chọn ra một nút lá mà điểm dữ liệu đó phải được bảo toàn được cấu trúc cây (theo ràng buộc xác định từ trước).

Để phân lớp và dự đoán số xe cho tập khách hàng thì cây OCT được xây dựng như sau:

- Xác định độ sâu đối đa của cây là D.
- Có được chiều sâu thì xác định được cây tối đa.
- Cây có $T = 2(D+1) - 1$ nút, theo chỉ số $t = 1, \dots, T$.
- $p(t)$ để chỉ nút cha của nút t và $A(t)$ để biểu thị tập các tổ tiên của nút t .
- $AL(t)$ là tập hợp các tổ tiên của t có nhánh trái được theo sau trên đường dẫn từ nút gốc đến t , và tương tự $AR(t)$ là tập hợp các tổ tiên nhánh phải, sao cho $A(t) = AL(t) \cup AR(t)$.
- Chia các nút trên cây theo hai nhóm:
 - + Các nút nhánh: Các nút $t \in T_B = \{1, \dots, [T/2]\}$ áp dụng phân chia có dạng $a^T x < b$. Các điểm thỏa mãn sự phân chia này theo nhánh bên trái trong cây và những điểm không theo đúng nhánh.
 - + Các nút lá: Các nút $t \in T_L = \{[T/2] + 1, \dots, T\}$ đưa ra dự đoán phân lớp cho mỗi điểm rơi vào nút lá.
- + Tạo sự lẩn vết của việc phân chia các nút trên theo ràng buộc xác định.
 - Thực hiện tổ chức cây lại theo phân cấp xác định.

- Xây dựng danh sách các biến để mô hình cấu trúc cây trong tập dữ liệu khách hàng.
- Thực hiện áp dụng hàm tối thiểu hóa các điểm dữ liệu bị phân lớp sai lầm
- Tổng chi phí phân loại sai là $\sum_{t \in T_1} L_t$ và độ phức tạp của cây là số lần tách trong cây, được cho bởi $\sum_{t \in T_B} d_t$.

- Xác định cây OCT cho tập dữ liệu khách hàng.
- Dự đoán số xe cần thiết trên tập dữ liệu khách hàng đã làm mịn.

4.3. Áp dụng ML để phân cụm và xác định số xe tối thiểu cần thiết cho tập khách hàng xác định

Giải thuật Clustering Framework Algorithm

Sử dụng giải thuật phân cụm Clustering Framework Algorithm:

Giải thuật Clustering Framework Algorithm

Procedure CFA

Input: tập tham số vào

Output: tập kết quả tìm được X

1. Khởi tạo tham số, thủ tục và thành phần cần thiết
 2. $X \leftarrow$ rỗng
 3. $C \leftarrow$ Tao_Cum(.) //phát sinh các cụm với $t = 0.1$
 4. $C_L \leftarrow$ Phan_Cum_Lai(.) //tạo các cụm trong C mà số xe dự đoán < ngưỡng
 5. $S \leftarrow$ Quy_Hoach_Cum(.) //xây dựng các cụm lại với ràng buộc theo các chỉ số ngưỡng
 - 5.1. $R \leftarrow$ Xu_Ly_R(.) //tính toán lại R theo điều kiện
 - 5.2. $R_x \leftarrow$ Xu_Ly_R_x(.)//tính toán lại R_x theo điều kiện
 6. $X \leftarrow$ To_Hop(.) //tổ hợp danh sách R và R_x
 7. $S \leftarrow$ Quy_Hoach_Cum_Phu(.) //xây dựng các cụm lại với ràng buộc theo các chỉ số ngưỡng theo C_L và có giải pháp A
 8. $X \leftarrow$ To_Hop(.)//tổ hợp danh sách X và A
 9. Return X
 10. end CFA.
-

5. CÀI ĐẶT, KẾT QUẢ VÀ BÀN LUẬN

5.1. Cài đặt và kết quả thực nghiệm

Các giải thuật (MGA, 6RIPSO, GWPSO, CFA) sẽ được triển khai với lập trình với Python và trên hệ điều hành Window 10 (64bit), trên máy Intel (R) Core (TM) i7 3.4 GHz CPU và Ram DDR4 16G/3600. Dữ liệu của bài toán VRP cụ thể sẽ được lấy từ 2 địa chỉ web sau:

- Trang web <http://web.cba.neu.edu> cung cấp chi tiết cơ sở dữ liệu của bài toán VRPTW. Trang web này còn cung cấp chi tiết dữ liệu của 6 tập dữ liệu bài toán khác nhau: R1-type; C1-type; RC1-type; R2-type; C2-type; RC2-type. Trang web này còn cung cấp các BKS cho các tập dữ liệu trên theo bảng thống kê, mỗi bảng đều có các cột: Problem (tên tập dữ liệu bài toán); NV (giá trị NV); Distance (khoảng cách/chi phí); Authors (tác giả). Chi tiết của các dữ liệu của bảng trên được trình bày tại <http://web.cba.neu.edu/~msolomon/heuristi.htm>.

- Trang web <http://neo.lcc.uma.es/vrp/vrp-instances>. được dành riêng cho việc nghiên cứu VRP, web này đã tổng hợp ở trong rất nhiều thông tin về chính bản thân web để truy cập công khai, gồm: Báo cáo kỹ thuật; Nhiều biến thể khác nhau của VRP; Thuật toán và kỹ thuật thay thế để giải

quyết nó; Một số trường hợp nổi tiếng của bài toán VRP; giải pháp tốt nhất BKS cho đến nay cho các trường hợp của bài toán VRP; Thư mục; Liên kết liên quan.

5.2. Đánh giá kết quả

Kết quả thực nghiệm của bài báo có sử dụng 4 chỉ số để đánh giá các giải thuật chuyên giải quyết nhóm bài toán VRP là: Tổng khoảng cách di chuyển Total Distance TD; Tổng khoảng cách di chuyển trung bình Average Total Distance ATD; Thời gian chạy CPU; và Chất lượng của các giải pháp. TD và ATD chính là các tiêu chí so sánh chính cho VRP và xác định giá trị của giải thuật. Thời gian chạy CPU mô tả hiệu quả giải thuật. Chất lượng của các giải pháp là chỉ số toàn diện, biểu thị tỷ lệ phần trăm sai lệch so với giải pháp BKS. Công thức tính tỷ lệ phần trăm sai lệch như sau:

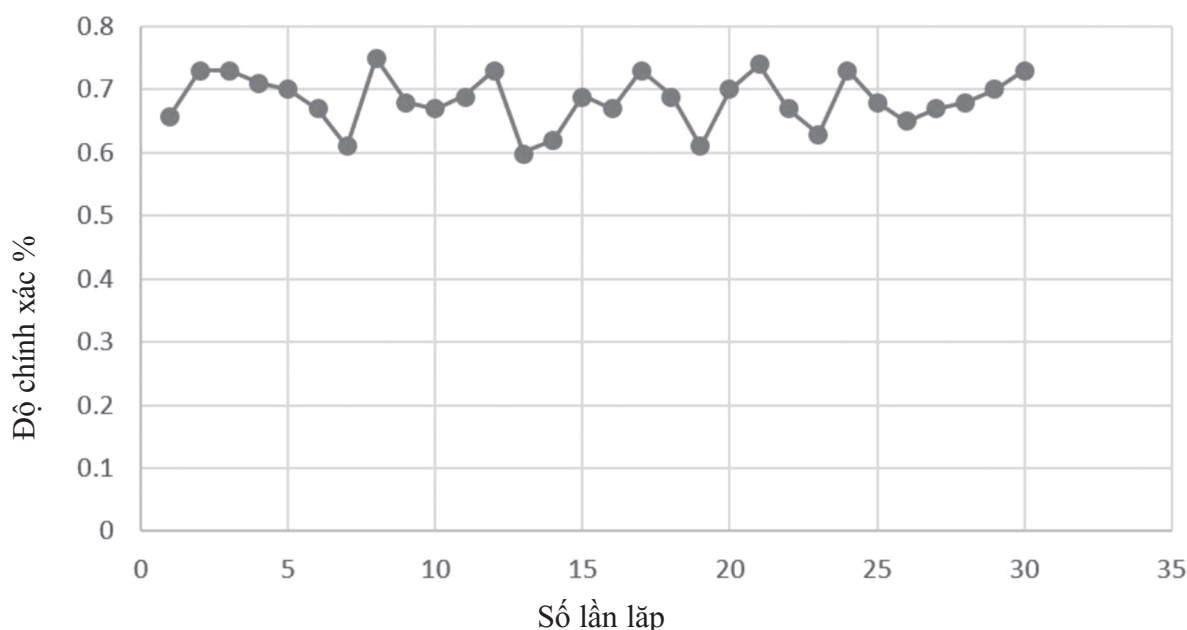
$$Z_{dev} = \frac{Z - Z_{best}}{Z_{best}} \times 100\%$$

Trong đó, Z_{dev} là tỷ lệ phần trăm sai lệch so với Best Known Solutions BKS, z là chi phí của giải pháp hiện tại, Z_{best} là chi phí của BKS.

Sau đây là một số kết quả thống kê thực nghiệm:

Bảng 1. Kết quả việc học với số khách hàng khác nhau

| Number customers | OCT accuracy (%) | Prediction error rate (%) |
|------------------|------------------|---------------------------|
| 500 | 75.2% | 5.1% |
| 1200 | 74.9% | 7.2% |
| 1900 | 76.2% | 9.4% |
| 3500 | 76.4% | 10.7% |
| 9000 | 76.6% | 8.9% |



Hình 1. Độ chính xác của OCT theo số lần lặp

5.3. Thảo luận

Trong bài báo này đã đề xuất một giải thuật lai GWPSO từ việc điều chỉnh giải thuật di truyền GA và kết hợp vào giải thuật PSO, sau đó xây dựng giải thuật CFA để phân lớp và tối thiểu hóa số lượng xe vận tải cần có. Việc sử dụng kỹ thuật Học máy vào việc giải quyết bài toán VRP theo hai bước: Mô hình phân lớp, áp dụng Máy học để dự đoán số lượng xe cần thiết cho từng tập khách hàng xác định; Giải thuật phân cụm mà có thể tận dụng được các tri thức có được từ sự phân lớp để có thể tối thiểu hóa số lượng xe vận tải cần có. Kết quả thực nghiệm cho thấy giải thuật lai đã thực thi với tập dữ liệu khách hàng đầu vào là nhỏ hơn từ 1% đến 5% và đã giảm tải xử lý cho giải thuật lai đã xây dựng. Các nghiên cứu trong tương lai có thể triển khai các hướng sau:

+ Xây dựng toán tử trong giải thuật Di truyền có hiệu quả tốt hơn cho bài toán VRP.

+ Gia tăng độ chính xác của phân cụm sẽ làm cho việc các giải thuật lai giải quyết bài toán VRP có được hiệu suất tốt hơn. Thực hiện việc này sẽ gặp hai vấn đề: Các cụm được tạo trong trường hợp xấu và xấu nhất; Tập dữ liệu khách hàng không được đồng nhất và theo các khung làm việc khác nhau.

LỜI CẢM ƠN

Nghiên cứu này được Trường Đại học Quốc tế Hồng Bàng cấp kinh phí thực hiện dưới mã số đề tài GVTC 15.15.

TÀI LIỆU THAM KHẢO

- [1]. G. B. Dantzig và J. H. Ramser, “*The truck dispatching problem*”. *Management Science*, 6(1), 80–91, 1959.
- [2]. Grigorios D. Konstantakopoulos, Sotiris P. Gayialis, Evripidis P. Kechagias, “*Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification*”. *Operational Research*, SpringerLink, 2020.
- [3]. Tomislav Erdelic, Tand Tonli Caric, (2019). “*A Survey on the Electric Vehicle Routing Problem: Variants and Solution Approaches*”. *Journal of Advanced Transportation* Volume 2019, Article ID 5075671, 48 pages.
- [4]. Rajeev Kr. Goela and Sandhya Rani Bansal, “*Hybrid algorithms for rich vehicle routing problems: a survey*”. *Computers and Operations Research*, Eslevier, 2019.
- [5]. Binbin Pan^b, Zhenzhen Zhang^a, Andrew Lim^b, “*A hybrid algorithm for time-dependent vehicle routing problem with time windows*”. *Computers and Operations Research*, Eslevier, 2021.
- [6]. Samuel Reong, Hui-Ming Wee, and Yu-Lin Hsiao, (2022). “*20 Years of Particle Swarm Optimization Strategies for the Vehicle Routing Problem: A Bibliometric Analysis*”. *MDPI, Basel, Switzerland*, 2020.
- [7]. Aldair #Alvarez; Pedro Munari An, “*Exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen*”. *Computers & Operations Research*, 2017.
- [8]. Sheng-Hua Xu, Ji-Ping Liu; Fu-Hao Zhang; Liang Wang; Li-Jian Sun, “*A Combination of Genetic Algorithm and Particle Swarm Optimization for Vehicle Routing Problem with Time Windows*”. *Research Center of Government GIS, Chinese Academy of Surveying and Mapping*, 2015.
- [9]. Yannis Marinakis; Magdalene Marinaki; Athanasios Migdalas, “*Particle Swarm Optimization for the Vehicle Routing Problem: A Survey and a Comparative Analysis*”. *Springer Nature Switzerland AG*, 2017.
- [10]. Hamed Alinezhad, Saeed Yaghubi, Seyyed-Mehdi Hoseini-Motlagh, Somayeh Allahyari,

- MojtabaSaghafi Nia, “An Improved Particle Swarm Optimization for a Class of Capacitated Vehicle Routing Problems”. *International Journal of Transportation Engineering*, 5(4), 2018.
- [11]. Mu-ChenChen; Yu-Hsiang Hsiao; Reddivari Himadeep Reddy, “The Self-Learning Particle Swarm Optimization approach for routing pickup and delivery of multiple products with material handling in multiple cross-docks”. *Transportation Research Part E: Logistics and Transportation Review*, Volume 91, Pages 208-226, 2016.
- [12]. Nguyễn Minh Đê, “Tổng quan một số dạng của bài toán lập lộ trình xe và giải thuật metaheuristic iterated local search có cải tiến để giải quyết một số dạng của bài toán lập lộ trình xe”. *Tạp chí Kinh tế-Công nghiệp Long An*, 3, 50-61, 2014.
- [13]. Bosman, P. A., & La Poutré, H., “Computationally intelligent online dynamic vehicle routing by explicit load prediction in an evolutionary algorithm”. In *Parallel problem solving from nature-ppsn ix* (pp. 312–321). Springer, 2006.
- [14]. Ventresca, M., Ombuki-Berman, B., & Runka, A., “Predicting genetic algorithm performance on the vehicle routing problem using information theoretic landscape measures”. In *European conference on evolutionary computation in combinatorial optimization* (pp. 214–225), 2013.
- [15]. Singh, A., Thakur, N., & Sharma, A., “A review of supervised machine learning algorithms”. In *2016 3rd international conference on computing for sustainable global development (indiacom)* (pp. 1310–1315), 2016.
- [16]. Varun Kumar SG1, Dr. R. Panneerselvam, “A Study of Crossover Operators for Genetic Algorithms to Solve VRP and its Variants and New Sinusoidal Motion Crossover Operator”. *International Journal of Computational Intelligence Research*, ISSN 0973-1873 Volume 13, Number 7, pp. 1717-1733, 2017.
- [17]. Aldair #Alvarez; Pedro Munari An, “Exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen”. *Computers & Operations Research*, February, 2017.
- [18]. Borja Vicario Medrano; Iván Colodro Sabell; Javier López Ruiz; Antonio Moratilla Ocana; Eugenio Fernández Vicente, “Optimal combination of operators in Genetic Algorithms for VRP Problem”. *International Journal of Modern Research In Engineering & Technology (IJMRET)*, 2018.
- [19]. Bertsimas, D., & Dunn, J., “Optimal classification trees”. *Machine Learning*, 106(7), 1039–1082, 2017.