

# ỨNG DỤNG GIẢI THUẬT TABU SEARCH TRONG GIẢI BÀI TOÁN ĐỊNH TUYẾN XE

## APPLICATION TABU SEARCH ALGORITHM FOR SOLVER VEHICLE ROUTING PROBLEM

Cao Ngọc Ánh, Trần Bích Thảo

*Khoa Công nghệ thông tin, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp*

Đến Tòa soạn ngày 13/05/2022, chấp nhận đăng ngày 08/06/2022

**Tóm tắt:** Mục tiêu chính của nghiên cứu này là tìm ra giải pháp cho vấn đề Định tuyến xe bằng cách sử dụng các thuật toán di truyền. Bài toán Định tuyến xe (Vehicle Routing Problem - VRP) là một bài toán tối ưu hóa tổ hợp phức tạp thuộc lớp NP - đầy đủ (nondeterministic polynomial - complete). Vehicle Routing Problem là một vấn đề toán học, và đề bài gốc của bài toán này gói gọn trong câu hỏi: “Làm thế nào để tạo ra một lộ trình tối ưu cho một đội xe giao hàng tới một lượng khách hàng có sẵn?”. Bài báo nghiên cứu thuật toán di truyền và kỹ thuật tìm kiếm Tabu để tìm ra giải pháp đúng hoặc gần đúng đến các vấn đề tối ưu hóa và tìm kiếm để giải bài toán Định tuyến xe.

**Từ khóa:** Bài toán định tuyến xe (VRP), thuật toán di truyền, thuật toán tìm kiếm Tabu.

**Abstract:** The main objective of this research is to find a solution to the Vehicle Routing Problem using genetic algorithms. Vehicle Routing Problem (VRP) is a complex combinatorial optimization problem of class NP-complete. The Vehicle Routing Problem is a math problem, and the original problem of this problem is encapsulated in the question: “How to create an optimal route for a fleet of vehicles to deliver to an existing customer?”. The article studies genetic algorithms and search techniques to find the correct or approximate solutions to optimization and search problems to solve vehicle routing problems.

**Keywords:** Vehicle Routing Problem (VRP), Genetic Algorithm, Tabu search.

### 1. GIỚI THIỆU

Có thể phát biểu bài toán VRP cơ bản một cách đơn giản như sau: Có một tập hợp  $M$  xe giống nhau cùng xuất phát tại một kho hàng đi làm nhiệm vụ giao hàng cho  $N$  khách hàng, mỗi khách hàng đòi hỏi cung cấp một lượng hàng nhất định. Yêu cầu đặt ra của bài toán là tìm đường đi ngắn nhất cho  $M$  xe đáp ứng được tất cả các đòi hỏi của khách hàng [1].

Vehicle Routing Problem bắt nguồn từ năm 1959 khi George Dantzig và John Ramser [2][3] thiết lập công thức toán học và phương

pháp tiếp cận bằng thuật toán để giải quyết vấn đề cung cấp xăng dầu cho các trạm dịch vụ. Năm 1964, Clarke và Wright đã cải tiến cách giải của Dantzig và Ramser bằng cách sử dụng một cách tiếp cận khác, được gọi là thuật toán tiết kiệm. Từ đó thì sự quan tâm đến VRP đã được mở rộng từ một nhóm các nhà toán học sang phạm vi rộng các nhà nghiên cứu và các nhà thực hành, từ các ngành khác nhau, trong nhiều lĩnh vực

Các phương pháp giải chính xác đảm bảo lời giải tối ưu sẽ được tìm thấy trong một khoảng

thời gian hữu hạn. Tuy nhiên, thời gian chạy của nó trong trường hợp tốt nhất là rất lớn, do đó lớp thuật toán này chỉ giải được những bài toán có kích thước nhỏ hoặc vừa như thuật toán nhánh cận (branch and bound) [4], thuật toán sinh cột [5],...

Các phương pháp gần đúng gồm các giải thuật cho chất lượng lời giải gần với lời giải tối ưu như nhóm các giải thuật heuristic cổ điển, nhóm các giải thuật tìm kiếm cục bộ và nhóm các giải thuật metaheuristic [6],[7]. Một trong số chúng là giải thuật tìm kiếm Tabu Search, là một giải thuật metaheuristic cung cấp một phương pháp kinh nghiệm cục bộ để khám phá không gian giải pháp ngoài mức tối ưu cục bộ.

Trong thuật toán này, từ một lời giải ban đầu, tìm kiếm Tabu sẽ lặp lại quá trình tìm kiếm nhằm cải thiện dần lời giải tốt nhất hiện có. Tại mỗi bước lặp, thuật toán sẽ duyệt trong một miền lân cận (hoặc toàn bộ lân cận của lời giải để chọn ra lời giải tốt nhất, lời giải này sẽ thay thế cho lời giải hiện tại ở bước lặp kế tiếp. Mỗi lời giải trong lân cận của lời giải hiện tại được gọi là một lân cận của lời giải hiện tại. Quá trình tác động lên lời giải hiện tại để biến nó thành một lân cận của lời giải hiện tại được gọi là một bước chuyển (move).

### **Giải thuật tìm kiếm Tabu có các thành phần chính như sau:**

#### *Tạo lời giải ban đầu:*

Lời giải ban đầu có thể được khởi tạo bằng một heuristic đơn giản hoặc phương pháp ngẫu nhiên.

#### *Chọn miền lân cận*

Tùy thuộc vào không gian tìm kiếm của bài toán mà có các cách thức chọn miền lân cận phù hợp. Thường thì hai cách thức sau được sử dụng: Thứ nhất là xét toàn bộ lân cận của

lời giải hiện tại và từ đó chọn ra lời giải tốt nhất; cách này sẽ không hiệu quả khi số lượng lân cận của lời giải là đủ lớn. Thứ hai là xét một tập con lân cận ngẫu nhiên của lời giải hiện tại (trong bài báo này chúng tôi sử dụng cách thứ hai này).

#### *Chọn lân cận*

Nếu bước chuyển tốt nhất trong miền lân cận có thể cải thiện được kỷ lục thì tất nhiên bước chuyển đó sẽ được chọn, ngược lại thì bước chuyển đó sẽ được chọn với xác suất  $p$  nào đó, nếu sau phép thử xác suất mà bước chuyển này vẫn không được chọn thì sẽ chuyển sang thực hiện bước lặp tiếp theo với lời giải hiện tại được giữ nguyên.

#### *Tiêu chuẩn mong đợi*

Một vấn đề có thể xảy ra là một bước chuyển dù đang bị cấm nhưng nó lại có khả năng cải thiện kỷ lục; do đó để tránh bỏ sót các bước chuyển tốt này, thuật toán tìm kiếm Tabu đưa ra khái niệm tiêu chuẩn mong đợi (aspiration criteria). Tiêu chuẩn mong đợi thường được áp dụng là: Nếu một bước chuyển Tabu có thể cải thiện được kỷ lục thì bước chuyển này vẫn được chọn và nó sẽ được loại khỏi **danh sách Tabu**.

#### *Chiến lược bổ sung*

Nhằm nâng cao chất lượng tìm kiếm, thuật toán tìm kiếm TABU đưa ra hai chiến lược tìm kiếm bổ sung là: chiến lược đa dạng hóa và chiến lược tăng cường hóa.

#### ▪ Đa dạng hóa lời giải (diversifying)

Mục đích của việc đa dạng hóa là hướng đến những miền không gian tìm kiếm mới. Có thể thực hiện việc đa dạng hóa lời giải bằng cách cho xáo trộn ngẫu nhiên một số phần tử của lời giải.

#### ▪ Tăng cường hóa lời giải (intensifying)

Mục đích của việc tăng cường hóa là tập trung tìm kiếm sâu hơn ở những vùng không gian tìm kiếm có triển vọng chứa lời giải tốt. Có thể thực hiện việc tăng cường hóa lời giải bằng cách là: Nếu sau một số bước lặp nhất định mà kỷ lục vẫn không được cải thiện; khi đó quá trình tìm kiếm TABU sẽ được khởi động lại với lời giải ban đầu chính là lời giải ứng với kỷ lục.

## 2. MÔ HÌNH TOÁN HỌC

Bài toán VRP được biểu diễn bởi một tập  $V$  các xe vận chuyển, một tập  $C$  các khách hàng và một đồ thị có hướng  $G$ . Đồ thị  $G$  có  $|C|+2$  đỉnh, các khách hàng là các đỉnh từ 1, 2, ..., n; đỉnh 0 biểu thị kho xuất phát và đỉnh  $n+1$  biểu diễn cho kho kết thúc. Tập  $A$  các cung đồ thị biểu diễn kết nối kho với khách hàng và giữa khách hàng với nhau. Mỗi cung  $(i, j)$  có giá trị chi phí  $c_{ij}$ . Mỗi khách hàng  $i$  có nhu cầu di. Một lộ trình bắt đầu từ kho xuất phát, đi qua một số khách hàng và kết thúc tại kho hàng kết thúc, mỗi khách hàng được một xe phục vụ. Mục tiêu của bài toán là tìm một phương án điều đội xe hợp lý để giao hàng tới tất cả các khách hàng sao cho tổng chi phí là nhỏ nhất.

Mô hình toán học của bài toán như sau, giả sử:

$n$ : tổng số khách hàng cần phục vụ;

$K$ : tổng số chuyên xe;

$N = \{0, 1, \dots, n+1\}$ : tập các đỉnh của đồ thị;

$V = \{0, 1, \dots, K\}$ : tập các xe vận chuyển;

$C = \{1, 2, \dots, n\}$ : tập các khách hàng;

$c_{ij}$ : chi phí đi từ  $i$  đến  $j$ ;

$q$ : sức chứa của một xe;

$d_i$ : nhu cầu tại nút  $i$ .

Khi đó ta có các biến trong mô hình:

$x_{ijk}$ : biểu diễn xem một xe thứ  $k$  có đi từ khách

hàng  $i$  đến khách hàng  $j$  không với

$$x_{ijk} = \begin{cases} 1 & \text{nếu xe thứ } k \text{ đi từ } i \text{ đến } j \\ 0 & \text{với các trường hợp còn lại} \end{cases}$$

trong đó  $i, j \in N, k \in V$

$y_{ik}$ : biểu diễn khách hàng  $i$  có được xe  $k$  phục vụ không với

$$y_{ik} = \begin{cases} 1 & \text{nếu xe thứ } k \text{ đi đến } i \\ 0 & \text{với các trường hợp còn lại} \end{cases}$$

trong đó  $i \in N, k \in V$

Hàm mục tiêu của bài toán:

$$\min \sum_{i \in N} \sum_{j \in N} \sum_{k \in V} c_{ij} x_{ijk}$$

Các ràng buộc của bài toán:

$$\sum_{i \in N} x_{ijk} = y_{jk} \quad \forall j \in N, k \in K \quad (1)$$

$$\sum_{i \in N} \sum_{k \in K} x_{ijk} = 1 \quad \forall j \in C \quad (2)$$

$$\sum_{i \in N} x_{ijk} - \sum_{i \in N} x_{ikj} = 0 \quad \forall j \in C, k \in K \quad (3)$$

$$\sum_{j \in N} y_{jk} * d_j \leq q \quad \forall k \in K \quad (4)$$

$$\sum_{j \in N} x_{0jk} = 1 \quad k \in K \quad (5)$$

$$\sum_{i \in N} x_{i0k} = 0 \quad \forall k \in K \quad (6)$$

$$\sum_{i \in N} x_{i(n+1)k} = 1 \quad \forall k \in K \quad (7)$$

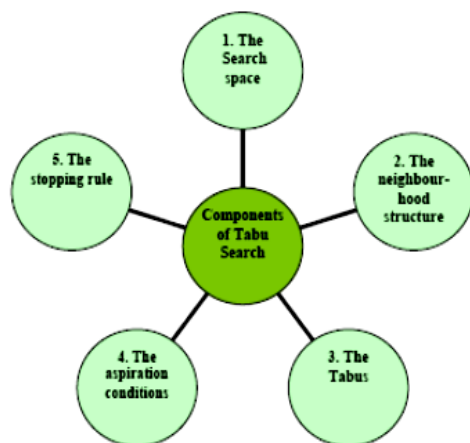
$$\sum_{j \in N} x_{(n+1)jk} = 0 \quad \forall k \in K \quad (8)$$

trong đó: (1) biểu diễn mối liên hệ giữa  $x$  và  $y$ . (2), (3): đảm bảo mỗi khách được một xe phục vụ. (4) đảm bảo sức chứa một xe. (5), (6), (7), (8) đảm bảo mọi lộ trình đều bắt đầu và kết thúc tại kho

Việc mô hình hóa bài toán VRP được nghiên cứu bởi nhiều tác giả và đã được trình bày chi tiết trong các bài báo [8], [9], [9], [10].

## 3. THUẬT TOÁN TABU SEARCH GIẢI BÀI TOÁN VRP

Các tham số chính cần thiết để chạy thuật toán Tabu Search được tóm tắt trong hình 1.



Hình 1. Thành phần của Tabu Search

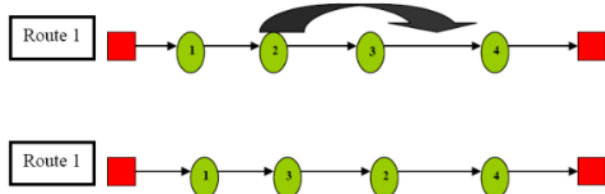
### 3.1. Không gian tìm kiếm

Trong bài toán này, không gian tìm kiếm bao gồm tất cả các tuyến khả thi thỏa mãn tất cả các ràng buộc đã cho đã nêu trước đó trong công thức của bài báo.

### 3.2. Cấu trúc của vùng lân cận

Cấu trúc vùng lân cận được tạo ra bằng cách áp dụng “di chuyển” cho giải pháp hiện tại tại mỗi lần lặp, từ đó tạo vùng lân cận  $N(x)$  trong không gian tìm kiếm.

Bài báo sử dụng cấu trúc lân cận như sau: Di chuyển một khách hàng tại mỗi lần lặp lại từ một tuyến đường và đặt khách hàng sau vào một tuyến đường khác hoặc một nơi nào đó khác trong cùng một tuyến đường. Ví dụ: trong sơ đồ bên dưới, khách hàng 2 đã được di chuyển khỏi vị trí hiện tại (giữa khách hàng 1 và khách hàng 3) và đã được chèn vào giữa khách hàng 3 và 4 (hình 2).



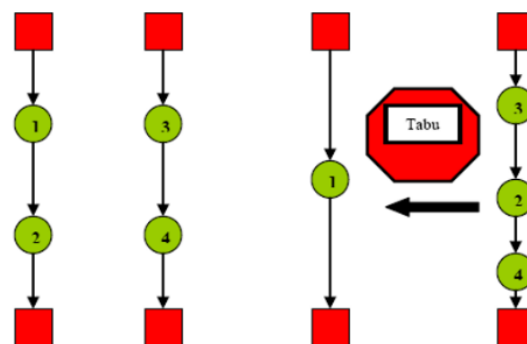
Hình 2. Cấu trúc vùng lân cận

### 3.3. Tabu Move

“Tabu Move” là tính năng cơ bản của Tabu

Search giúp phân biệt nó với Tìm kiếm cục bộ. Trên thực tế, “Tabus” được sử dụng để ngăn chặn việc di chuyển khỏi tối ưu hóa cục bộ thông qua các bước di chuyển không cải thiện (Brasssy và cộng sự 2001) [11].

Ví dụ: nếu khách hàng 2 đã được chuyển từ tuyến 1 và được đặt trong tuyến 2, việc chuyển khách hàng 2 trở lại tuyến 1 có thể được khai báo Tabu cho một số lần lặp.



Hình 3. Tabu move

Quyền sở hữu Tabu trên thực tế đề cập đến số lần lặp lại mà một nước đi được khai báo Tabu. “Tabu move” được lưu trữ trong cấu trúc bộ nhớ ngắn hạn được gọi là danh sách Tabu. Việc lưu trữ này có thể được thực hiện theo một số cách, cụ thể là:

1. Ghi giải pháp hoàn chỉnh;
2. Ghi lại một số phép biến đổi cuối cùng được thực hiện trên giải pháp hiện tại và ngăn chặn các phép biến đổi ngược lại.

### 3.4. Tiêu chuẩn mong đợi

Theo Brasssy và cộng sự (2001) [11], một tiêu chuẩn mong đợi là một thuật toán cho phép người ta hủy bỏ “Tabu move”. Tiêu chuẩn mong đợi đơn giản nhất liên quan đến việc cho phép di chuyển, mặc dù là “Tabu move”, nếu nó dẫn đến một giải pháp có giá trị mục tiêu lớn hơn so với giải pháp tốt nhất hiện tại.

### 3.5. Điều kiện dừng

Ở một số giai đoạn, quá trình tìm kiếm lặp đi

lặp lại phải kết thúc. Trong thuật toán Tìm kiếm Tabu, có một số cách để kết thúc quá trình tìm kiếm, chẳng hạn:

- Sau một số lần lặp cố định.
- Sau một số lần lặp lại mà không có sự cải thiện trong giá trị hàm mục tiêu.
- Khi vật kính đạt đến giá trị ngưỡng được chỉ định trước.

Như vậy, thuật toán Tabu Search giải bài toán VRP được thể hiện như sau:

Bước 1: Khởi tạo lời giải ban đầu  $i$  trong tập  $S$ , gán  $i^* = i$  và  $k = 0$ ;

Bước 2: Gán  $k = k + 1$  và tạo ra một tập con  $V^*$  của giải pháp  $N(i, k)$  sao cho một trong các điều kiện Tabu bị vi phạm hoặc ít nhất một trong các “tiêu chuẩn mong đợi” được thỏa mãn;

Bước 3: Chọn chỉ số  $j$  tốt nhất trong  $V^*$  và gán  $i = j$ ;

Bước 4: Nếu  $f(i) < f(i^*)$  và gán  $i^* = i$ ;

Bước 5: Cập nhật Tabu và “tiêu chuẩn mong đợi”;

Bước 6: Nếu thỏa mãn “điều kiện dừng” thì dừng lại, ngược lại thì chuyển quay lại bước 2.

Trong đó:

$i, j$ : chỉ số của giải pháp;

$k$ : chỉ số lặp lại;

$V^*$ : tập con của giải pháp;

$N(i, k)$ : lân cận của giải pháp  $i$  ở vòng lặp  $k$ ;

$f(i)$ : giá trị hàm mục tiêu cho giải pháp  $i$ .

#### 4. THỬ NGHIỆM VÀ ĐÁNH GIÁ THUẬT TOÁN

Dưới đây, thuật toán Tabu được cài đặt bằng ngôn ngữ lập trình R. Chương trình được viết và biên dịch trên Rstudio 1.4.1717 và chạy

trên máy tính cá nhân Core i7-E7440 4600U 2.7GHz 8G RAM. Sau khi khởi động RStudio, chọn project là CVRP\_NCKH.

Hàm đánh giá:

```
evaluation_ func (move_value, frequency,
Tabu_time_left)=
move_value + k1 * frequency + k2 *
Tabu_time_left
```

Trong đó:

$$k_1 = \begin{cases} c_1 \cdot \text{iter} \cdot \text{abs}(\max(\text{move\_value}, 1)), & \text{nếu } \text{freq} > \text{avgfreq} \\ c_2 \cdot \text{abs}(\text{move\_value}), & \text{nếu } \text{freq} \leq \text{avgfreq} \end{cases}$$

$$k_2 = \begin{cases} 0 & \text{nếu move là Tabu} \\ c_3 \cdot \text{abs}(\text{move\_value}), & \text{nếu move không là Tabu} \end{cases}$$

Trong đó:

$k_1, k_2$ : hệ số ảnh hưởng của việc di chuyển tần suất và thời gian còn lại của di trong trong Tabu

$c_1, c_2, c_3$ : là các hệ số thực nghiệm lần lượt tương ứng là 10; 0,01 và 0,01

Các thông số cấu hình cho thuật toán được thiết lập như sau:

- Số xe sử dụng 50 xe;
- Chi phí sử dụng một xe là 200.

Kết quả thực nghiệm cho  $I_{\max} = 10$ :

Hiệu quả của các thuật toán Tabu phụ thuộc vào khởi tạo quần thể và các chiến lược bổ sung trong Tabu. Kết quả có được khi so sánh thực nghiệm Tabu với số liệu tham khảo trên website

<https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/100-customers/> [12], chi tiết được thể hiện trong bảng 1.

Trong bảng 1, cột đầu thể hiện 4 nhóm mẫu dữ liệu với các bộ dữ liệu tiêu chuẩn số lượng khách hàng lần lượt là 100, 200, 300, 400 khách hàng. Với kết quả thực nghiệm, với dữ liệu khách hàng từ 300 khách hàng đến 400 khách hàng, thuật toán Tabu cho kết quả tốt hơn.

**Bảng 1. So sánh kết quả trung bình giữa các lần thực nghiệm**

<i>Phương pháp</i> <i>Mẫu</i>	<i>Kết quả tốt nhất</i>	<i>Chương trình TABU</i>
R1	12.00 1217.73	36.00 3070.24
R2	2.73 967.75	
C1	10.00 828.38	
C2	3.00 589.68	16.78 1985.25

## 5. KẾT LUẬN

Các thuật toán Tabu Search cung cấp một cách tiếp cận rất thú vị để giải quyết các vấn đề không thể áp dụng một phương pháp chính xác. Bài báo đã trình bày rõ thêm các kiến thức cơ bản Định tuyến xe, làm rõ thuật toán Tabu Search và tiến hành thực nghiệm với bộ dữ liệu được phân loại cụ thể.

Việc tăng các giải thuật heuristic và kết hợp các giải thuật tìm kiếm trong các quá trình là hướng nghiên cứu tiếp theo của bài báo.

## TÀI LIỆU THAM KHẢO

- [1] M. Fisher, "Vehicle routing. Handbooks of Operations Research and Management Science", Chapter 1, 8:1-31, (1995).
- [2] Sergios Theodoridis, Konstantinos Kourtoumbas, "Pattern Recognition Second Edition page 582.
- [3] Dantzing, G., Ramster, R., "The truck dispatching problem", Management Science 6, page 80-91, (1959).
- [4] G.Laporte, Y.Nobert, "A branch and bound algorithm for the capacitated vehicle routing problem", Operations Research Spektrum, pp 77-85, (1983).
- [5] F.Liberatore, G.Righini, M.Salani, "A column generation algorithm for the vehicle routing problem with soft time windows", 4OR quarterly journal of the Belgian, French and Italian Operations Research Societies 9(1), pp 49-82, (2010).
- [6] M. Gendreau, J.Y. Potvin, O. Braysy, G. Hasle, A. Lokketangen, "Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography", in B.Golden, S.Ragavan and E.Wasil (editors), *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer-Verlag, Heidelberg, Germany, pp 143-169, (2007).
- [7] S.C.H.Leung, J.Zheng, D.Zhang, X.Zhou, "Simulated annealing for the vehicle routing problem with two-dimensional loading constraints", Flexible Services and Manufacturing Journal 22(1), pp 61-82, (2010).
- [8] Thangiah S.R, "Vehicle Routing with Time Windows using Genetic Algorithms", In L. Chambers, editor, *Application handbook of genetic algorithms: New Frontiers vol. II*. Boca Raton: CRC Press, 253-277, (1995).
- [9] Tan K.C., Lee L.H., Zhu K.Q., Ou K, "Heuristic methods for vehicle routing problem with time windows", Artificial Intelligence in Engineering, Volume 15, Issue 3, Pages 281-295, (2001).
- [10] Kallehauge B., Larsen J., Madsen O.B., Solomon M.M, "Vehicle Routing Problem with Time Windows", In: Desaulniers G., Desrosiers J., Solomon M.M. (eds) *Column Generation*. Springer, Boston, MA, (2005).
- [11] Brassy, O & Gendreau M, "Tabu search heuristics for the Vehicle Routing Problem with time windows", (2001).
- [12] <https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/100-customers/>

**Thông tin liên hệ:** Cao Ngọc Ánh

Điện thoại: 0912599122 - Email: cnanh@uneti.edu.vn

Khoa Công nghệ thông tin, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp.



