

LUẬN VĂN THẠC SĨ

Phương pháp tìm kiếm lân cận rộng thích ứng
cho bài toán định tuyến xe

Đại học Quốc gia Hà Nội
Trường Đại học Khoa học Tự nhiên
Khoa Toán-Cơ-Tin học

Giảng viên hướng dẫn: TS. Hoàng Nam Dũng

Học viên: Nguyễn Mạnh Linh

Mục lục

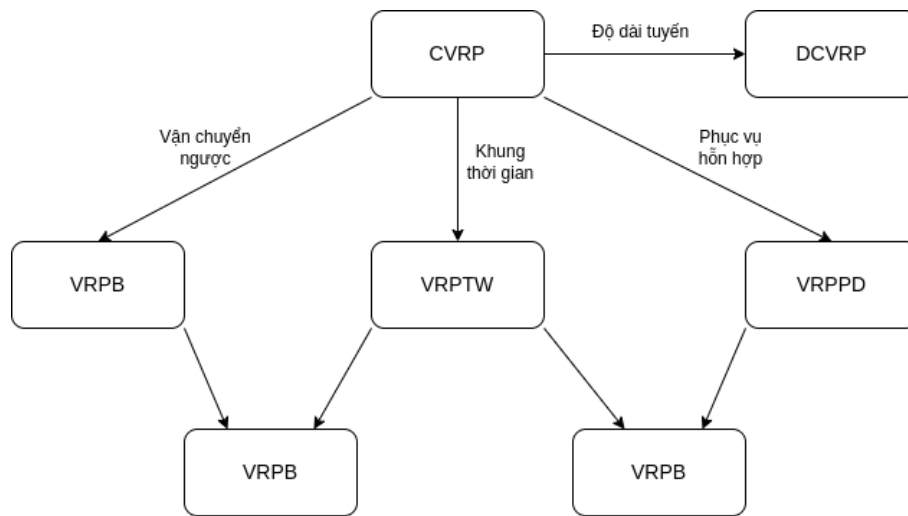
1	Mở đầu	1
2	Định nghĩa và một số kí hiệu	2
2.1	Định nghĩa bài toán	2
2.1.1	CVRP	2
2.1.2	CVRPTW	4
2.1.3	VRPPD	5
2.2	Mô hình toán học	5
3	Một số phương pháp cho VRP	9
3.1	Thuật toán chính xác	9
3.1.1	Nhánh và cận	9
3.1.2	Quy hoạch động	9
3.1.3	Công thức dòng xe và thuật toán	9
3.1.4	Công thức dòng hàng và thuật toán	10
3.1.5	Công thức phân hoạch tập hợp và thuật toán	11
3.2	Heuristics cổ điển	12
3.2.1	Thuật toán tiết kiệm	12
3.2.2	Heristics phân hoạch tập hợp	12
3.2.3	Phân cụm trước, heuristics tuyển sau	12
3.2.4	Heuristics cải tiến	12
3.3	Metaheuristics	12
3.3.1	Tìm kiếm cục bộ	12
3.3.2	Tìm kiếm phổ biến	12
3.3.3	Cơ chế học	12
4	Phương pháp tìm kiếm lân cận	13
4.1	Tìm kiếm lân cận	13
4.2	Tìm kiếm lân cận rộng	13
4.3	Tìm kiếm lân cận rộng thích ứng	13
5	Ứng dụng ALNS vào CVRPTW	14
5.1	Thuật toán hủy	14
5.2	Thuật toán sửa	14
5.3	Tiêu chí chấp nhận nghiệm	14
6	Thực nghiệm và kết quả	15
7	Kết luận	16
	Tài liệu tham khảo	II

1 Mở đầu

2 Định nghĩa và một số kí hiệu

Trong chương này, chúng ta sẽ đưa ra định nghĩa chính tắc cho lớp các bài toán định tuyến xe. Các định nghĩa này được xây dựng theo ngôn ngữ của lý thuyết đồ thị được đưa ra bởi Toth (2002) [10].

Các bài toán được mô tả thuộc lớp VRP (*vehicle routing problem*) bao gồm *định tuyến xe với ràng buộc tải trọng* - CVRP (*capacited VRP*), *định tuyến xe với ràng buộc khung thời gian* - VRPTW (*VRP with time windows*), *định tuyến xe với lấy và giao hàng* - VRPPD (*VRP with pickup and delivery*).



Hình 2.1: Các bài toán, biến thể của VRP

2.1 Định nghĩa bài toán

2.1.1 CVRP

Trước hết ta xem xét mô hình cho bài toán *nguyên bản*: *bài toán định tuyến xe với ràng buộc tải trọng*. Một cách tự nhiên, tại sao không phải là VRP (không ràng buộc)? Bạn sẽ thấy rằng nếu không có bất kì ràng buộc nào thì một xe có thể phục vụ tất cả các yêu cầu và bài toán VRP sẽ suy biến về TSP (*travelling salesman problem*). Ít nhất ràng buộc về tải trọng là thực tế và giữ cho mỗi xe chỉ phục vụ được một số yêu cầu nhất định (trong trường hợp số yêu cầu không quá nhỏ cũng như tải trọng của xe là quá lớn).

Gọi $G = (V, A)$ là một đồ thị đầy đủ với $V = \{0, \dots, n\}$ là tập nút và A là tập các cung. Các nút $i = 1, \dots, n$ đại diện cho các yêu cầu hay khách hàng cần phục vụ, nút 0 là kho hàng. Đôi khi, kho hàng cũng được biểu diễn bằng nút $n + 1$.

Một số không âm được gọi là chi phí c_{ij} đại diện cho mỗi cung $(i, j) \in A$. Nói cách khác c_{ij} là chi phí cần bỏ ra để di chuyển từ nút i tới nút j . Trong bài toán này và hầu hết các bài toán định tuyến ta không định nghĩa cạnh (i, i) nên có thể gán $c_{ii} = \infty$ với $i \in V$.

Nếu đồ thị là có hướng thì ma trận chi phí c là bất đối xứng, khi đó ta có bài toán CVRP bất đối xứng ACVRP (*asymmetric CVRP*). Ngược lại nếu $c_{ij} = c_{ji}$ với mọi $(i, j) \in A$ ta có bài toán CVRP đối xứng SCVRP (*symmetric CVRP*) và các cung của A được thay thế bằng tập cách cạnh vô hướng E . Với một cạnh $e \in E$, ta định nghĩa $\alpha(e)$ và $\beta(e)$ là nút bắt đầu và kết thúc của cạnh.

Đồ thị G phải là đồ thị kết nối mạnh và nhìn chung ta giả thiết đồ thị G là đầy đủ. Với một nút i , gọi $\Delta^+(i)$ là tập ra của i (*forward star*), được định nghĩa là tập các nút j mà cung $(i, j) \in A$, nói cách khác đây là tập các nút có thể tiếp cận trực tiếp từ nút i . Tương tự như vậy, $\Delta^-(i)$ là tập vào của i (*backward star*), được định nghĩa là tập các nút j mà cung $(j, i) \in A$ hay là tập các nút tiếp cận trực tiếp tới nút i . Với một tập nút con $S \subseteq V$, gọi $\delta(S)$ là tập các cạnh $e \in E$ chỉ có một hoặc cả hai đầu nút thuộc S . Để thuận tiện, khi xét một nút $i \in V$, ta viết $\delta(i)$ thay cho $\delta(\{i\})$.

Trong hầu hết các bài toán thực tế, ma trận chi phí thỏa mãn bất đẳng thức tam giác

$$c_{ij} + c_{jk} \geq c_{ik} \quad \forall i, j, k \in V \quad (2.1)$$

Nói cách khác việc đi trực tiếp từ nút i tới nút j luôn tốn ít chi phí hơn là đi gián tiếp. Với nhiều thuật toán, bất đẳng thức tam giác là điều kiện cần, điều này có thể được đảm bảo bằng cách thêm một đại lượng dương lớn (hợp lý) vào chi phí của mỗi cung. Ta chú ý thêm rằng nếu chi phí của mỗi cung thuộc đồ thị bằng với chi phí của đường đi ngắn nhất giữa hai đầu nút của cung thì ma trận chi phí thỏa mãn bất đẳng thức tam giác.

Trong nhiều trường hợp, tập các nút nằm trên một mặt phẳng, vị trí của chúng được cho bởi tọa độ và chi phí c_{ij} của mỗi cung $(i, j) \in A$ là khoảng cách Euclide giữa hai điểm ứng với nút i và j . Khi đó, ma trận chi phí là đối xứng và thỏa mãn bất đẳng thức tam giác. Bài toán này được gọi là *Euclidian SCVRP*.

Mỗi khách hàng i có một nhu cầu (về tải trọng) là d_i và nhu cầu của kho $d_0 = 0$. Với một tập nút $S \subseteq V$, ta kí hiệu $d(S) = \sum_{i \in S} d_i$ là tổng nhu cầu của tập.

Một tập hợp K đại diện cho các xe, mỗi xe có tải trọng C và sẵn sàng ở kho. Ta giả thiết $d_i \leq C$ với mỗi $i = 1, \dots, n$. Giả thiết này là cần thiết để mỗi khách hàng đều được phục vụ. Mỗi xe phục vụ nhiều nhất một tuyến và ta giả thiết K không nhỏ hơn K_{min} với K_{min} là số xe ít nhất cần để phục vụ toàn bộ khách hàng.

Với một tập $S \subseteq V \setminus \{0\}$, ta gọi $r(S)$ là số xe ít nhất để phục vụ toàn bộ khách hàng thuộc tập S . Chú ý rằng $r(V \setminus \{0\}) = K_{min}$.

CVRP yêu cầu tìm một tập chính xác K các chu trình đơn (mỗi chu trình ứng với một tuyến đường) với tổng chi phí của tất cả các cung thuộc các chu trình này là nhỏ nhất. Lời giải phải thỏa mãn các ràng buộc sau:

- (i) Mỗi chu trình đều đi qua nút ứng với kho hàng
- (ii) Mỗi nút ứng với một khách hàng được đi qua bởi đúng một chu trình
- (iii) Tổng nhu cầu của các khách hàng trong mỗi chu trình không được vượt quá tải trọng của xe.

2.1.2 CVRPTW

Bài toán định tuyến xe với ràng buộc thời gian - VRPTW (*VRP with time windows*) là một mở rộng của CVRP. Trong đó ngoài ràng buộc về tải trọng cho mỗi xe, mỗi khách hàng i bị ràng buộc bởi một khoảng thời gian $[a_i, b_i]$ được gọi là khung thời gian hay cửa sổ thời gian (*time window*). Thời gian phục vụ khách hàng i là s_i . Thời gian di chuyển từ nút i tới nút j là t_{ij} với mỗi cung $(i, j) \in A$ hay t_e với $e \in E$. Ngoài ra nếu xe đến nút i sớm thì phải chờ đến thời gian a_i mới được phục vụ. Nếu xe đến nút i muộn hơn thì khách hàng sẽ không được phục vụ.

Thường thì ma trận chi phí và ma trận thời gian di chuyển là như nhau, hơn nữa các xe được giả thiết đều xuất phát từ kho tại thời điểm 0. Ràng buộc thời gian dẫn tới mỗi tuyến đường là có hướng (có thứ tự đi đến các nút) ngay cả khi ma trận chi phí là đối xứng. Chính vì thế, VRPTW thường được mô tả như một bài toán bất đối xứng.

VRPTW yêu cầu tìm một tập chính xác K chu trình đơn với tổng chi phí là nhỏ nhất, thỏa mãn các ràng buộc sau đây:

- (i) Mỗi chu trình đều đi qua nút ứng với kho hàng
- (ii) Mỗi nút ứng với một khách hàng được đi qua bởi đúng một chu trình
- (iii) Tổng nhu cầu của các khách hàng trong mỗi chu trình không được vượt quá tải trọng của xe
- (iv) Với mỗi khách hàng i , thời gian bắt đầu phục vụ phải nằm trong khung thời gian $[a_i, b_i]$ và xe ngừng phục vụ sau khoảng thời gian s_i

VRPTW là bài toán NP-khó, nó là trường hợp tổng quát của CVRP. Nếu ta đặt $a_i = 0$ và $b_i = \infty$ với $i \in V \setminus \{0\}$ thì VRPTW suy biến về CVRP. Ngoài ra ta cũng thu được biến thể TSP với ràng buộc thời gian (TSPTW) nếu $C \geq d(V)$ và $K = 1$.

2.1.3 VRPPD

Một biến thể khác nữa của CVRP là bài toán định tuyến xe với lấy và giao hàng (*VRP with pickup and delivery - VRPPD*). Trong đó, mỗi khách hàng i có thêm hai đại lượng đặc trưng nữa là d_i và p_i lần lượt là nhu cầu lấy và giao tại khách hàng i . Đôi khi chỉ một đại lượng $d_i = d_i - p_i$ được sử dụng cho mỗi khách hàng i để chỉ lượng nhu cầu chênh lệch giữa việc lấy và giao hàng (có thể là số âm). Với mỗi khách hàng i , gọi O_i là nút đại diện cho việc giao hàng và D_i là nút đại diện cho điểm lấy hàng.

Giả thiết rằng, tại mỗi điểm khách hàng, điểm giao được phục vụ trước điểm lấy. Do đó, tải hiện tại của một xe trước khi tới điểm đã cho là tải ban đầu trừ đi tổng nhu cầu đã giao cộng với tổng nhu cầu đã lấy.

VRPPD yêu cầu tìm chính xác một tập K các chu trình đơn với tổng chi phí là nhỏ nhất, thỏa mãn các ràng buộc sau đây:

- (i) Mỗi chu trình đều đi qua nút ứng với kho hàng
- (ii) Mỗi nút ứng với một khách hàng được đi qua bởi đúng một chu trình
- (iii) Tải hiện tại của xe trong suốt quá trình phục vụ không âm và không được vượt quá tải trọng của xe
- (iv) Với mỗi khách hàng i , khách hàng O_i khác với kho phải được phục vụ trong cùng một tuyến và trước khách hàng i
- (v) Với mỗi khách hàng i , khách hàng D_i khác với kho phải được phục vụ trong cùng một tuyến và sau khách hàng i .

VRPPD là trường hợp tổng quát của CVRP. Nếu ta đặt $O_i = D_i = 0$ và $p_i = 0$ cho mọi $i \in V$ thì VRPPD suy biến về CVRP. Hơn nữa nếu đặt $K = 1$ thì ta thu được TSP với lấy và giao hàng (*TSP with pickup and delivery - TSPPD*).

2.2 Mô hình toán học

Chương này trình bày biểu diễn toán học cho bài toán VRPTW. Trong luận văn này, tác giả tập trung giải quyết VRPTW, từ đó ta cũng có thể giản ước về CVRP

cũng như tổng quát với VRPPD (*VRP with pickup and delivery*) hoặc PDPTW (*pickup and delivery with time window*). Như đã trình bày ở chương trước VRPTW là một mở rộng của CVRP với ràng buộc khung thời gian. Trong đó mỗi khách hàng i được ràng buộc bởi một khung thời gian $[a_i, b_i]$. Xe không được đến i tại thời điểm $t_i > b_i$, ngoài ra nếu đến sớm hơn thời điểm a_i hay $t_i < a_i$ thì xe cần phải chờ tới thời điểm a_i để phục vụ khách hàng. Thời gian phục vụ của khách hàng i là s_i .

VRPTW là bài toán NP-khó, việc tìm lời giải hay nghiệm tối ưu (chính xác) gần như là bất khả thi. Để dễ hình dung, xét bài toán VRP, với số lượng khách hàng $n = 100$, và chỉ một xe, số lượng lời giải là $n! \approx 10^{158}$. Nếu ta có số CPU ước tính bằng toàn bộ số nguyên tử trong vũ trụ $n_{CPU} \approx 10^{80}$, thời gian nhỏ nhất là thời gian Plank $t_p \approx 5.39 \times 10^{-44}$. Để kiểm tra toàn bộ lời giải có phải nghiệm tối ưu ta cần thời gian $T \approx 10^{158} \times 5.39 \times 10^{-44} / 10^{80} \approx 5.39 \times 10^{34}$. Để so sánh, tuổi của vũ trụ được ước tính khoảng 4.33×10^{17} . Nghĩa là ta sẽ mất thời gian lớn gấp cỡ *một trăm triệu tỉ* lần tuổi của cả vũ trụ! ¹

Như đã trình bày ở chương trước, VRPTW được định nghĩa trên đồ thị $G = (V, A)$, kho hàng được biểu diễn bởi nút 0 và $n + 1$. Một tuyến thỏa mãn là một đường đi trên đồ thị G bắt đầu từ 0 và kết thúc ở $n + 1$. Nếu kho hàng được biểu diễn chỉ bởi nút 0 thì tuyến thỏa mãn là một đơn chu trình trên đồ thị G chứ nút 0. Khung thời gian của nút 0 và $n + 1$ là $[a_0, b_0] = [a_{n+1}, b_{n+1}] = [E, L]$, trong đó E và L lần lượt là thời gian sớm nhất rời kho và thời gian muộn nhất trở về kho. Ngoài ra, thời gian phục vụ và nhu cầu của kho đều được đặt bằng 0, hay $s_0 = s_{n+1} = 0$ và $d_0 = d_{n+1} = 0$. Lời giải chấp nhận được chỉ tồn tại nếu $a_0 = E \leq \min_{i \in V \setminus \{0\}} \{b_i - t_{0i}\}$ và $b_{n+1} = L \geq \min_{i \in V \setminus \{0\}} \{a_i + s_i + t_{0i}\}$. Chú ý rằng, cung $(i, j) \in A$ có thể được bỏ đi nếu không thỏa mãn ràng buộc thời gian $a_i + s_i + t_{ij} > b_j$ hoặc vi phạm ràng buộc về tải trọng $d_i + d_j > C$. Cuối cùng nếu mục tiêu chính là giảm thiểu số lượng xe thì cung $(0, n + 1)$ với $c_{0,n+1} = t_{0,n+1} = 0$ phải được thêm vào A .

Tiếp theo, chúng ta trình bày một mô hình toán cho VRPTW với hai biến: biến x_{ijk} (*flow variable*) với $(i, j) \in A, k \in K$ nhận giá trị 1 nếu xe k đi trực tiếp từ nút i tới nút j và 0 nếu ngược lại. Biến w_{ik} với $i \in V, k \in K$ là thời gian bắt đầu phục vụ khách hàng i bởi xe k . VRPTW được mô hình một cách chính tắc như sau theo Toth (2002) [10]:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \quad (2.2)$$

¹ Slides của Thibaut Vidal (SOICT, Nha Trang 2017)

Với ràng buộc:

$$\sum_{k \in K} \sum_{j \in \Delta^+(i)} x_{ijk} = 1 \quad \forall i \in N, \quad (2.3)$$

$$\sum_{j \in \Delta^+(0)} x_{0jk} = 1 \quad \forall k \in K, \quad (2.4)$$

$$\sum_{i \in \Delta^-(j)} x_{ijk} - \sum_{i \in \Delta^+(j)} x_{jik} = 0 \quad \forall k \in K, j \in N, \quad (2.5)$$

$$\sum_{i \in \Delta^-(n+1)} x_{i,n+1,k} = 1 \quad \forall k \in K, \quad (2.6)$$

$$x_{ijk}(w_{ik} + s_i + t_{ij} - w_{jk}) \leq 0 \quad \forall k \in K, (i, j) \in A, \quad (2.7)$$

$$a_i \sum_{j \in \Delta^+(i)} x_{ijk} \leq w_{ik} \leq b_i \sum_{j \in \Delta^+(i)} x_{ijk} \quad \forall k \in K, i \in N, \quad (2.8)$$

$$E \leq w_{ik} \leq L \quad \forall k \in K, i \in \{0, n+1\}, \quad (2.9)$$

$$\sum_{i \in N} d_i \sum_{j \in \Delta^+(i)} x_{ijk} \leq C \quad \forall k \in K, \quad (2.10)$$

$$x_{ijk} \geq 0 \quad \forall k \in K, (i, j) \in A, \quad (2.11)$$

$$x_{ijk} \in \{0, 1\} \quad \forall k \in K, (i, j) \in A. \quad (2.12)$$

Hàm mục tiêu trong phương trình (2.2) biểu diễn tổng chi phí của tất cả các tuyến đường. Tập $N = V \setminus \{0\}$ biểu diễn cho tập khách hàng.

- Ràng buộc (2.3) đảm bảo rằng mỗi khách hàng chỉ được phục vụ bởi một xe.
- Ràng buộc (2.4) đảm bảo rằng mỗi xe phải xuất phát từ kho hàng.
- Ràng buộc (2.5) đảm bảo rằng trên một tuyến, nếu khách hàng i được phục vụ thì trước và sau đó đều có một khách hàng khác được phục vụ hoặc trước và sau đó là kho hàng. Nói cách khác, khách hàng i phải ở giữa tuyến.
- Ràng buộc (2.6) đảm bảo rằng mỗi xe phải trở về kho hàng.
- Ràng buộc (2.7) đảm bảo về khung thời gian khi xe đi từ khách hàng i tới khách hàng j . Nếu xe k đi từ khách hàng i tới khách hàng j thì thời gian bắt đầu phục vụ khách hàng i cộng với thời gian phục vụ khách hàng i cộng với thời gian di chuyển từ khách hàng i tới khách hàng j phải nhỏ hơn hoặc bằng thời gian bắt đầu phục vụ khách hàng j . Dấu bằng xảy ra khi xe đến j sau thời điểm a_j (khách hàng j được phục vụ luôn), nếu đến sớm hơn thì xe phải chờ để phục vụ khách hàng.
- Ràng buộc (2.8) đảm bảo rằng thời gian bắt đầu phục vụ khách hàng i bởi xe k nằm trong khung thời gian $[a_i, b_i]$.

- Ràng buộc (2.9) đảm bảo rằng thời gian bắt đầu phục vụ khách hàng i bất kì phải nằm trong khoảng thời gian từ sớm nhất xuất phát từ kho và muộn nhất về kho.
- Ràng buộc (2.10) đảm bảo rằng tổng tải của mỗi xe không được vượt quá tải trọng tối đa C .
- Ràng buộc (2.11) và (2.12) đảm bảo điều kiện nhị phân của *flow variable* x_{ijk} .

Ta có thể nhận thấy rằng, ràng buộc (2.8) ép $w_{ik} = 0$ nếu như khách hàng i không được phục vụ bởi xe k . Điều kiện nhị phân trong ràng buộc (2.12) cho phép ràng buộc (2.7) được thay thế bởi

$$w_{ik} + s_i + t_{ij} - w_{jk} \leq (1 - x_{ijk})M_{ij} \quad \forall k \in K, (i, j) \in A, \quad (2.13)$$

với M_{ij} là các hằng số rất lớn. Hơn nữa M_{ij} có thể thay bằng $\max\{b_i + s_i + t_{ij} - a_j, 0\}$ với $(i, j) \in A$ và như vậy ta chỉ cần kiểm tra ràng buộc (2.7) và (2.13) cho các cung $(i, j) \in A$ thỏa mãn $M_{ij} > 0$. Mặt khác, khi $\max\{b_i + s_i + t_{ij} - a_j, 0\} = 0$ các điều kiện này được thỏa mãn với mọi w_{ik} , w_{jk} và x_{ijk} .

Chúng ta không cần đưa ra mô hình cho CVRP nữa, bởi ta có thể bỏ qua các ràng buộc về thời gian ở điều kiện từ (2.7) đến (2.9). Khi đó VRPTW suy biến về CVRP như đã trình bày ở những phần trước đó. Tác giả cũng không đưa ra mô hình cho VRPPD hay PDPTW để tránh sự phức tạp. VRPTW vừa đủ để ta có một mô hình đẹp và thực tế.

3 Một số phương pháp cho VRP

Trong chương này, chúng ta sẽ xem xét một số khái niệm cần thiết và phương pháp để giải (lớp) bài toán định tuyến xe. Trong suốt chặng đường hơn 50 năm của bài toán VRP, có rất nhiều phương pháp được nghiên cứu và thực nghiệm từ các thuật toán giải chính xác đến các thuật toán xấp xỉ. Ba lớp thuật toán được trình bày bao gồm *thuật toán chính xác*, *heuristics cổ điển* và *metaheuristics*. Lớp các thuật toán được trình bày một cách khái quát theo Laporte, Gilbert (2009) [7]. Cuối cùng tác giả đưa ra lựa chọn và đi sâu vào thuật toán tìm kiếm lân cận rộng thích ứng - ALNS (*Adaptive Large Neighborhood Search*) để giải quyết bài toán VRPTW trong luận văn này.

3.1 Thuật toán chính xác

3.1.1 Nhánh và cận

Một trong những thuật toán chính xác được nghiên cứu sớm nhất là *nhánh và cận*, lần đầu xuất hiện trong bài báo "*An Algorithm for the Vehicle Dispatching Problem*" của N. Christofides và S. Eilon năm 1969 [3]

3.1.2 Quy hoạch động

Eilon, Watson-Gandy và Christofides (1971) [3] đưa ra lời giải cho bài toán VRP bằng phương pháp quy hoạch động. Gọi $c(S)$ là chi phí tối ưu của một tuyến ứng với tập nút $S \subseteq V \setminus \{0\}$. Mục tiêu là cực tiểu hóa $\sum_{r=1}^m c(S_r)$ trên tất cả các quy hoạch khả dĩ $\{S_1, \dots, S_m\}$ của $V \setminus \{0\}$. Gọi $f_k(U)$ là chi phí nhỏ nhất có thể đạt được khi sử dụng k xe cho một tập con U của $V \setminus \{0\}$. Ta có:

$$f_k(U) = \begin{cases} c(U) & \text{nếu } k = 1, \\ \min_{U^* \subseteq U \subseteq V \setminus \{0\}} \{f_{k-1}(U \setminus U^*) + c(U^*)\} & \text{nếu } k > 1. \end{cases} \quad (3.1)$$

Chi phí tối ưu là $f_m(V \setminus \{0\})$ và các tuyến là các phân hoạch của $V \setminus \{0\}$ theo phương trình (3.1).

3.1.3 Công thức dòng xe và thuật toán

Công thức 2-chỉ số cho bài toán VRP được nghiên cứu đầu tiên bởi Laporte, Nobert (1983) [8] và Laporte, Nobert, Desrochers (1985) [9] và mở rộng công thức TSP cổ

điền của Dantzig, Fulkerson, Johnson (1954) [4]. Gọi x_{ij} là biến 0-1-2 bằng số lần một xe đi qua cung (i, j) . Bài toán được mô hình hóa như sau:

$$\text{cực tiểu } \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (3.2)$$

với ràng buộc:

$$\sum_{j=1}^n x_{0j} = 2m, \quad (3.3)$$

$$\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2 \quad (k \in V \setminus \{0\}), \quad (3.4)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - \nu(S) \quad (S \subseteq V \setminus \{0\}), \quad (3.5)$$

$$x_{0j} = 0, 1, 2 \quad (j \in V \setminus \{0\}), \quad (3.6)$$

$$x_{ij} = 0, 1 \quad (i, j \in V \setminus \{0\}), \quad (3.7)$$

trong đó $\nu(S)$ là cận dưới của số lượng xe cần thiết để phục vụ tập S .

3.1.4 Công thức dòng hàng và thuật toán

Trong công thức dòng hàng, biến y_{ij} (hoặc y_{ijk}) định nghĩa tải (lượng hàng) của xe mang theo trên cung (i, j) . Ví dụ được trình bày bởi Gavish, Graves (1979) [6], tuy nhiên các tác giả không đưa ra kết quả tính toán. Các ví dụ gần đây hơn được nghiên cứu bởi Baldacci, Hadjiconstantinou, Mingozzi (2004) [1] dựa trên mô hình TSP của Finke, Claus, Gunn (1984) [5]. Công thức cho một đồ thị mở rộng $\bar{G} = (\bar{V}, \bar{E})$, với $\bar{V} = V \cup \{(i, n+1) : i \in V\}$. Một tuyến được định nghĩa là một đường đi có hướng từ 0 đến $n+1$. Biến nhị phân x_{ij} bằng 1 khi và chỉ khi cạnh (i, j) được chọn vào tuyến. Biến y_{ij} định nghĩa tải của xe trên cung (i, j) và $y_{ji} = Q - y_{ij}$ biểu diễn xe rỗng trên cung (j, i) mỗi khi $x_{ij} = 1$. Công thức dòng hàng được mô hình hóa như sau:

$$\text{cực tiểu } \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (3.8)$$

với ràng buộc:

$$\sum_{j \in \bar{V}} (y_{ji} - y_{ij}) = 2q_i \quad (i \in V \setminus \{0\}), \quad (3.9)$$

$$\sum_{j \in V \setminus \{0\}} y_{0j} = \sum_{i \in V \setminus \{0\}} q_i, \quad (3.10)$$

$$\sum_{j \in V \setminus \{0\}} y_{j0} = mQ - \sum_{i \in V \setminus \{0\}} q_i, \quad (3.11)$$

$$\sum_{j \in V \setminus \{0\}} y_{n+1,j} = mQ, \quad (3.12)$$

$$y_{ij} + y_{ji} = Qx_{ij} \quad ((i, j) \in \bar{E}), \quad (3.13)$$

$$\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2 \quad (k \in V \setminus \{0\}), \quad (3.14)$$

$$y_{ij} \geq 0, y_{ji} \geq 0 \quad ((i, j) \in \bar{E}), \quad (3.15)$$

$$x_{ij} = 0, 1 \quad ((i, j) \in \bar{E}). \quad (3.16)$$

Bài toán này được giải bằng *branch-and-cut* với các bất đẳng thức VRP được biểu diễn theo các biến x_{ij}

3.1.5 Công thức phân hoạch tập hợp và thuật toán

Công thức phân hoạch tập hợp đơn giản của VRP lần đầu được nghiên cứu bởi Balinski, Quandt (1964) [2]. Gọi r là một tuyến, a_{ir} là hệ số nhị phân có giá trị bằng 1 khi và chỉ khi nút $i \in V \setminus \{0\}$ thuộc tuyến r , gọi c^* là chi phí tối ưu của tuyến r và gọi y_k là biến nhị phân bằng 1 khi và chỉ khi tuyến r được dùng trong lời giải tối ưu. Bài toán được mô hình hóa như sau:

$$\text{cực tiểu } \sum_r c_r^* y_r \quad (3.17)$$

với ràng buộc:

$$\sum_r a_{ir} = 1 \quad (i \in V \setminus \{0\}), \quad (3.18)$$

$$\sum_r y_r = m, \quad (3.19)$$

$$y_r = 0, 1 \quad (\text{mọi } r). \quad (3.20)$$

Nói một cách chặt chẽ thì ràng buộc (3.19) không phải một phần của công thức phân hoạch tập hợp chuẩn, tuy nhiên nó được sử dụng bởi hầu hết các nhà nghiên cứu trong trường hợp VRP.

3.2 Heuristics cổ điển

Nhìn chung thì các thuật toán giải chính xác khó đảm bảo hiệu năng trong thực tế khi mà các tập dữ liệu ngày càng lớn và các doanh nghiệp cần phục vụ khách hàng một cách nhanh chóng và tiết kiệm. Thực tế người ta cần tìm ra một (số) lời giải chấp nhận được đủ tốt trong một khoảng thời gian "hợp lý". Từ những năm 1964 cho đến 1990, rất nhiều heuristics được nghiên cứu. Một số ít là đưa ra thuật toán hoàn toàn mới còn hầu hết là cải tiến thuật toán đã có.

3.2.1 Thuật toán tiết kiệm

Thuật toán tiết kiệm được đưa ra bởi Clark, Wright (1964) [[clarke1964scheduling](#)], mô tả và cài đặt khá đơn giản nhưng vẫn đưa ra được nghiệm tốt. Chính vì thế, thuật toán này được sử dụng rất rộng rãi. Thuật toán bắt đầu với nghiệm ban đầu với n tuyến $(0, i, 0)$ với $i \in V \setminus \{0\}$. Tại mỗi vòng lặp thuật toán nối tuyến kết thúc với i với một tuyến khác bắt đầu với j cực đại hóa đại lượng *tiết kiệm* $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ và lời giải mới thỏa mãn các ràng buộc. Quá trình kết thúc khi không thể nối các tuyến vào nữa.

Một số cải tiến được đề xuất, ví dụ như nhân c_{ij} với một trọng số dương λ (Golden, Magnanti, Nguyen (1977) [[golden1977implementing](#)]), tối ưu tuyến đường hợp nhất toàn cục thông qua việc sử dụng thuật toán phù hợp (Altinkemer, Gavish (1991) [[altinkemer1991parallel](#)] và Wark, Holt (1994) [[wark1994repeated](#)]), tăng tốc tính toán (Paessens (1988) [...]).

3.2.2 Heuristics phân hoạch tập hợp

3.2.3 Phân cụm trước, heuristics tuyến sau

3.2.4 Heuristics cải tiến

3.3 Metaheuristics

3.3.1 Tìm kiếm cục bộ

3.3.2 Tìm kiếm phổ biến

3.3.3 Cơ chế học

4 Phương pháp tìm kiếm lân cận

4.1 Tìm kiếm lân cận

4.2 Tìm kiếm lân cận rộng

4.3 Tìm kiếm lân cận rộng thích ứng

5 Ứng dụng ALNS vào CVRPTW

5.1 Thuật toán hủy

5.2 Thuật toán sửa

5.3 Tiêu chí chấp nhận nghiệm

6 Thực nghiệm và kết quả

7 Kết luận

Tài liệu tham khảo

- [1] Roberto Baldacci, Eleni Hadjiconstantinou, and Aristide Mingozzi. “An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation”. In: *Operations research* 52.5 (2004), pp. 723–738.
- [2] Michel L Balinski and Richard E Quandt. “On an integer program for a delivery problem”. In: *Operations research* 12.2 (1964), pp. 300–304.
- [3] Nicos Christofides and Samuel Eilon. “An algorithm for the vehicle-dispatching problem”. In: *Journal of the Operational Research Society* 20 (1969), pp. 309–318.
- [4] George Dantzig, Ray Fulkerson, and Selmer Johnson. “Solution of a large-scale traveling-salesman problem”. In: *Journal of the operations research society of America* 2.4 (1954), pp. 393–410.
- [5] Gerd Finke. “A two-commodity network flow approach to the traveling salesman problem”. In: *Congresses Numeration* 41 (1984), pp. 167–178.
- [6] Bezalel Gavish and Stephen C Graves. “The travelling salesman problem and related problems”. In: (1978).
- [7] Gilbert Laporte. “Fifty years of vehicle routing”. In: *Transportation science* 43.4 (2009), pp. 408–416.
- [8] Gilbert Laporte and Yves Nobert. “A branch and bound algorithm for the capacitated vehicle routing problem”. In: *Operations-Research-Spektrum* 5 (1983), pp. 77–85.
- [9] Gilbert Laporte, Yves Nobert, and Martin Desrochers. “Optimal routing under capacity and distance restrictions”. In: *Operations research* 33.5 (1985), pp. 1050–1073.
- [10] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2002.