## **Giới thiệu bài toán VRSPTW**

### **1.1. Mô tả bài toán**

Bài toán Vehicle Routing and Scheduling Problem with Time Windows (VRSPTW) nhấn mạnh việc xây dựng lộ trình cho đội xe nhằm phục vụ tập hợp khách hàng sao cho tổng chi phí là nhỏ nhất, đáp ứng các ràng buộc về sức chứa và khung thời gian phục vụ.

#### **Đặc điểm chính của bài toán:**

1. **Depot và đội xe:**
   * Depot là nơi tập trung đội xe và là điểm khởi hành cũng như điểm kết thúc của mỗi lộ trình.
   * Mỗi xe trong đội xe có sức chứa giới hạn và chỉ có thể phục vụ khách hàng trong phạm vi sức chứa này.
2. **Khách hàng:**
   * Mỗi khách hàng có nhu cầu dịch vụ nhất định và một khung thời gian quy định khi nào dịch vụ có thể bắt đầu.
   * Xe đến trước khung thời gian của khách hàng phải chờ cho đến khi dịch vụ được phép thực hiện.
3. **Ràng buộc:**
   * Tất cả các lộ trình phải không vi phạm khung thời gian quy định của khách hàng.
   * Xe không được phục vụ vượt quá sức chứa quy định.
   * Tổng chi phí là quãng đường di chuyển từ depot đến các khách hàng sau đó quay về depot

### **1.2. Mục tiêu**

Mục tiêu của bài toán là:

* Tối thiểu hóa chi phí tổng
* Phục vụ tất cả khách hàng trong khung thời gian quy định.
* Tuân thủ các ràng buộc về sức chứa và thời gian.

## 

## 

## **2. Thuật toán giải quyết VRSPTW trong bài báo**

### **2.1. Thuật toán tuần tự (Sequential Approach)**

Thuật toán tuần tự của Solomon (1987) hoạt động như sau:

**Khởi tạo tuyến đường**

* Mỗi tuyến bắt đầu với việc lựa chọn một **khách hàng hạt giống** (*seed customer*), thường là khách hàng xa nhất so với kho trung tâm hoặc khách hàng có hạn chót phục vụ sớm nhất.
* Sau khi chọn khách hàng hạt giống, các khách hàng chưa được phục vụ sẽ lần lượt được thêm vào tuyến hiện tại cho đến khi tuyến đạt đến giới hạn về dung lượng xe hoặc thời gian phục vụ.

**Chèn khách hàng vào tuyến**

* Hai tiêu chí chính được sử dụng để chèn khách hàng vào tuyến:
  + Tiêu chí chọn vị trí chèn tốt nhấ**t**: Đối với mỗi khách hàng chưa được phục vụ, phương pháp xác định vị trí chèn khả thi tốt nhất vào tuyến hiện tại sao cho chi phí bổ sung là nhỏ nhất.
  + Tiêu chí chọn khách hàng để chèn: Sau khi tính toán vị trí chèn tốt nhất cho tất cả khách hàng chưa được phục vụ, khách hàng nào có chi phí chèn tốt nhất sẽ được chọn để thêm vào tuyến.

**Quá trình lặp lại**

* Khi không thể thêm khách hàng nào vào tuyến hiện tại do các ràng buộc, một tuyến mới sẽ được khởi tạo với khách hàng hạt giống mới. Quá trình tiếp tục cho đến khi tất cả khách hàng đều được phục vụ.

Thuật toán tuần tự giải quyết được nhiều tình huống thực tế, nhưng có nhược điểm chính là khi gần kết thúc thuật toán, các khách hàng còn lại thường bị phân tán rộng khắp, dẫn đến các lộ trình cuối không tối ưu.

### 

### **2.2. Thuật toán song song (Parallel Route Building Algorithm)**

Thuật toán xây dựng tuyến song song được Potvin và Rousseau đề xuất như một cải tiến so với thuật toán tuần tự của Solomon nhằm giải quyết bài toán định tuyến và lập lịch xe với ràng buộc cửa sổ thời gian (Vehicle Routing and Scheduling Problem with Time Windows – VRPTW). Mục tiêu chính của thuật toán này là khắc phục nhược điểm của phương pháp tuần tự – đó là việc các khách hàng chưa được phục vụ cuối cùng thường bị phân tán không đồng đều, dẫn đến chất lượng tuyến kém.

#### **Bước 1: Khởi tạo số tuyến ban đầu**

Quá trình khởi tạo tuyến ban đầu được thực hiện như sau:

1. Sử dụng thuật toán tuần tự của Solomon để ước lượng ban đầu số tuyến cần thiết. Việc ước lượng này được thực hiện bằng cách áp dụng phương pháp khởi tạo theo “khách hàng xa nhất” và sử dụng một bộ tham số (β = 1, α₁ = 1, α₂ = 0) cho việc chèn khách hàng.
2. Sau khi chạy thuật toán tuần tự, ta thu được một tập hợp các tuyến ban đầu và các khách hàng hạt giống (seed customers) trong mỗi tuyến. Các khách hàng hạt giống này được chọn sao cho phân bố đều trong khu vực địa lý, nhằm giảm thiểu khả năng các khách hàng còn lại bị phân tán rộng khi tiến dần đến giai đoạn sau.

#### **Bước 2: Chèn khách hàng**

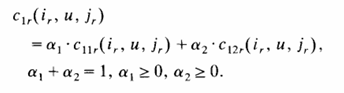
#### Sau khi khởi tạo tuyến, thuật toán sẽ tiến hành chèn các khách hàng chưa được phục vụ vào các tuyến hiện có:

Tính chi phí chèn:

Đối với mỗi khách hàng chưa được phục vụ, chi phí chèn vào tất cả các tuyến hiện tại được tính dựa trên hai thành phần:

* Quãng đường gia tăng khi chèn khách hàng.
* Thời gian chờ gia tăng do cần đáp ứng ràng buộc cửa sổ thời gian.

Công thức chi phí chèn được biểu diễn dưới dạng tổng có trọng số giữa quãng đường và thời gian chờ:  
Trong đó:



* Detour(i, u, j) là quãng đường gia tăng khi chèn khách hàng *u* vào giữa hai khách hàng liên kề *i* và *j*.
* Delay(i, u, j) là thời gian chờ gia tăng tương ứng khi chèn *u*.

#### **Bước 3: Chọn khách hàng dựa trên chỉ số hối tiếc**

Sau khi tính toán chi phí chèn cho tất cả khách hàng chưa được phục vụ, thuật toán sẽ chọn khách hàng có chỉ số hối tiếc cao nhất để chèn trước. Chỉ số hối tiếc của mỗi khách hàng được tính bằng hiệu số giữa chi phí chèn tốt nhất và chi phí chèn tốt thứ hai trong tất cả các tuyến:

Trong đó:

* là chi phí chèn tốt nhất của khách hàng *u*.
* Khách hàng có chỉ số hối tiếc cao nhất đồng nghĩa với việc nếu không chèn ngay lập tức, chi phí sẽ gia tăng đáng kể khi chèn sau này.

#### **Bước 4: Lặp lại cho đến khi tất cả khách hàng được phục vụ**

Quá trình chèn khách hàng vào các tuyến được lặp lại liên tục cho đến khi mời khách hàng đã được phục vụ hoặc không còn khách hàng nào có thể chèn vào tuyến một cách khả thi.

#### **Bước 5: Điều chỉnh số lượng tuyến**

* Trong trường hợp thuật toán không tìm được giải pháp khả thi với số tuyến ban đầu, thuật toán sẽ tăng dần số tuyến cho đến khi tìm được giải pháp khả thi.
* Khi tăng số tuyến, mỗi tuyến mới được khởi tạo bằng cách chọn khách hàng xa nhất so với tập hợp các khách hàng hạt giống hiện tại.

#### **Bước 6: Lặp lại với các bộ tham số khác nhau**

* Thuật toán được thực hiện với ba bộ tham số khác nhau (đối trọng giữa quãng đường và thời gian chờ) nhằm đảm bảo tìm được giải pháp tốt nhất.

## **3. So sánh với thuật toán đã cài đặt được**

| Bước | Thuật toán đã cài đặt | Thuật toán trong bài báo |
| --- | --- | --- |
| 1. Khởi tạo số tuyến ban đầu | - Sử dụng thuật toán tuần tự của Solomon để ước lượng số tuyến cần thiết. | - Sử dụng thuật toán tuần tự của Solomon với tham số cố định (β = 1, α₁ = 1, α₂ = 0) để ước lượng số tuyến và chọn khách hàng hạt giống. |
| - Chọn khách hàng xa nhất từ kho làm hạt giống. | - Khách hàng hạt giống được chọn sao cho phân bố đều để tránh phân tán ở giai đoạn sau |
| 2. Chèn khách hàng | - Tính toán chi phí chèn dựa trên Detour và Delay, có sử dụng trọng số để điều chỉnh mức độ ưu tiên giữa hai yếu tố. | |
| 3. Chọn khách hàng để chèn | - Sử dụng giá trị hối tiếc tổng quát để chọn khách hàng tiếp theo | - Sử dụng chỉ số hối tiếc tổng quát, định nghĩa bằng sự khác biệt giữa chi phí chèn tốt nhất và các chi phí chèn khác |
| 4. Lặp lại | - Lặp lại việc chèn khách hàng cho đến khi không còn khách hàng khả thi hoặc tất cả khách hàng được phục vụ. | - Sử dụng thêm trọng số để cân nhắc tốt hơn giữa chi phí quãng đường và thời gian chờ. |
| 5. Điều chỉnh số lượng tuyến | - Khi không thể tìm được giải pháp khả thi, thêm tuyến mới bằng cách chọn khách hàng xa nhất so với tập khách hàng hạt giống. | - Linh hoạt hơn: Có thể giảm số tuyến bằng cách loại bỏ các khách hàng hạt giống gần nhau hoặc tăng tuyến khi cần thiết. |
| 6. Lặp lại với tham số khác nhau | Đều dùng nhiều bộ tham số để tìm giải pháp tốt nhất | |

**4. Kết quả**

Bảng so sánh kết quả (số tuyến đường - chi phí) giữa thuật toán đã cài đặt được và kết quả tối ưu của bộ mẫu thử

**4.1. Clustered problems**

| Dataset | Kết quả cài đặt | Solution |
| --- | --- | --- |
| C101 | 12 - 935.4 | 10 - 827.3 |
| C102 | 13 - 1452.9 | 10 - 827.3 |
| C103 | 13 - 1433.8 | 10 - 826.3 |
| C204 | 4 - 911.4 | 3 - 588.1 |
| C205 | 4 - 726.2 | 3 - 586.4 |
| C206 | 4 - 740 | 3 - 586 |

**4.2. Random problems**

| Dataset | Kết quả cài đặt | Solution |
| --- | --- | --- |
| R107 | 15 - 1532 | 11 - 1064 |
| R108 | 13 - 1275 | 10 - 932 |
| R109 | 17 - 1836 | 13 - 1146 |
| R201 | 5 - 1645 | 8 - 1143 |
| R202 | 5 - 1450 | 8 - 1029 |
| R203 | 5 - 1215 | 6 - 870 |

**4.3. Mixed problems**

| Dataset | Kết quả cài đặt | Solution |
| --- | --- | --- |
| RC106 | 15 - 1923 | 12 - 1372 |
| RC107 | 15 - 1836 | 12 - 1207 |
| RC203 | 4 - 1343 | 5 - 923 |
| RC204 | 4 - 1090 | 4 - 783 |