

# Electric Vehicle Routing Problems (EVRP)

Lê Ngọc Mỹ Trang  
Computer Science  
University of Information Technology  
Vietnam National University  
Ho Chi Minh City, Vietnam  
20520817@gm.uit.edu.vn

Nguyễn Thế Vinh  
Computer Science  
University of Information Technology  
Vietnam National University  
Ho Chi Minh City, Vietnam  
20520862@gm.uit.edu.vn

Chu Kim Chí  
Computer Science  
University of Information Technology  
Vietnam National University  
Ho Chi Minh City, Vietnam  
20521129@gm.uit.edu.vn

Lê Minh  
Computer Science  
University of Information Technology  
Vietnam National University  
Ho Chi Minh City, Vietnam  
20521599@gm.uit.edu.vn

## I. THỰC TRẠNG

Việc phát triển công nghệ xe điện đang tăng lên, đồng thời yêu cầu về bảo vệ môi trường và giảm khí thải ô nhiễm đang trở thành một ưu tiên quan trọng. Điều này dẫn đến sự gia tăng của các phương tiện điện và sự phụ thuộc ngày càng tăng vào việc xây dựng hệ thống sạc hạ tầng hiệu quả.

Tuy nhiên, việc quản lý và vận hành một hệ thống vận chuyển dựa trên xe điện không chỉ đơn giản là đặt điểm sạc ở các vị trí và định tuyến các xe. EVRP đòi hỏi cân nhắc các yếu tố như năng lượng, dung lượng, khoảng cách và yêu cầu khách hàng. Các hạn chế về năng lượng và thời gian sạc cũng phải được xem xét để đảm bảo hiệu quả hoạt động của hệ thống.

## II. GIỚI THIỆU

### A. Giới thiệu EVRP

Bài toán Electric Vehicle Routing Problems (EVRP) là một bài toán quan trọng trong lĩnh vực quản lý và tối ưu hóa hệ thống vận tải hàng hóa, đặc biệt là trong ngữ cảnh xe điện. EVRP tập trung vào việc xác định lộ trình tối ưu cho các xe điện trong quá trình giao hàng, đảm bảo tính hiệu quả và tiết kiệm năng lượng.

EVRP kết hợp các yếu tố cơ bản của bài toán Routing Problems và các yếu tố đặc thù của xe điện. Nhiệm vụ chính của EVRP là tìm lộ trình tối ưu cho các xe điện sao cho đảm bảo việc giao hàng đúng thời gian và tiết kiệm năng lượng, đồng thời thỏa mãn các ràng buộc về khoảng cách, thời gian và sức chứa của xe.

Các phương pháp giải quyết EVRP thường sử dụng các kỹ thuật tối ưu hóa, thuật toán di truyền, lập lịch và các phương pháp thông minh nhân tạo để tìm lời giải hiệu quả. Mục tiêu cuối cùng là cải thiện hiệu quả vận chuyển và hỗ trợ sự phát

triển của công nghệ và hạ tầng xe điện trong hệ thống giao thông hiện đại.

### B. Các phương pháp hiện tại

Heuristics và Metaheuristics: Các phương pháp này tập trung vào việc tìm kiếm các giải pháp gần tối ưu trong thời gian hợp lý. Một số phương pháp phổ biến bao gồm:

- Random Heuristic: Tạo các giải pháp ngẫu nhiên và cải thiện dần chúng.
- Greedy Heuristic: Xây dựng giải pháp bằng cách lựa chọn khách hàng gần nhất cho từng xe và thực hiện các lựa chọn tiếp theo dựa trên tiêu chí nhất định.
- Local Search: Cải thiện giải pháp bằng cách tìm kiếm các lân cận của nó và thay thế chúng bằng các giải pháp tốt hơn.
- Simulated Annealing: Tìm kiếm các giải pháp lân cận và chấp nhận các giải pháp tốt hơn hoặc có xác suất chấp nhận các giải pháp tồi hơn dựa trên nhiệt độ giảm dần.
- Genetic Algorithms: Sử dụng các toán tử di truyền như lai ghép và đột biến để tạo ra các thế hệ giải pháp và tìm kiếm giải pháp tốt nhất trong quần thể.
- Ant Colony Optimization: Mô phỏng quá trình tìm kiếm thức ăn của kiến và sử dụng một số phép toán để cập nhật và tìm kiếm giải pháp tốt nhất.

Chuỗi công cụ tối ưu hóa: Kết hợp nhiều phương pháp tối ưu hóa để giải quyết bài toán EVRP, bao gồm các phương pháp tìm kiếm cục bộ, tìm kiếm toàn cục và quy hoạch tuyến tính. Mô hình toán học và phương pháp tìm kiếm chính xác: Xác định các mô hình toán học cho bài toán EVRP và áp dụng các phương pháp tìm kiếm chính xác như quy hoạch nguyên, quy hoạch tuyến tính, lập lịch động, hoặc phân nhánh và cận. Học máy và trí tuệ nhân tạo: Sử dụng các phương pháp học máy và trí tuệ nhân tạo để dự đoán và tối ưu hóa các yếu tố như

nhu cầu khách hàng, mức tiêu thụ năng lượng và đặc điểm địa lý để tạo ra các giải pháp tối ưu cho EVRP.

### C. Các thuật ngữ

**Điểm giao hàng:** Được đặt ở các vị trí khác nhau trong mạng lưới, mỗi điểm giao hàng yêu cầu một lượng hàng hóa cụ thể để được giao đến. **Điểm sạc:** Với xe điện, việc sạc lại pin là cần thiết để duy trì hoạt động của xe. Điểm sạc cung cấp điện năng để xe tiếp tục hành trình và thường được đặt tại các vị trí chiến lược trong mạng lưới. **Ràng buộc:** Bài toán EVRP cần xem xét các ràng buộc như sức chứa của xe, thời gian giao hàng, giới hạn sạc pin và các hạn chế khác liên quan đến việc vận chuyển hàng hóa bằng xe điện. **Mục tiêu tối ưu:** Mục tiêu của EVRP là tìm lộ trình tối ưu để giảm thiểu chi phí hoặc thời gian vận chuyển, đồng thời tối ưu hóa việc sử dụng năng lượng và tăng cường sự bền vững của hệ thống vận tải.

## III. DỮ LIỆU THỰC NGHIỆM

Gồm 17 file .evrp có các thông tin (Bảng I):

- **OPTIMAL VALUE:** giá trị tối ưu (nếu có)
- **VEHICLES:** số chuyến xe tối thiểu
- **DIMENSION:** số lượng khách hàng (tính luôn tổng kho)
- **STATIONS:** số lượng trạm sạc
- **CAPACITY:** số lượng hàng hóa tối đa xe có thể chở
- **ENERGY CAPACITY:** số lượng pin tối đa của xe điện
- **ENERGY CONSUMPTION:** tỉ lệ tiêu thụ pin của xe điện
- **EDGE WEIGHT FORMAT:** là dạng khoảng cách tính toán giữa các nút trong đồ thị (ở đây là Euclid)
- **NODE COORD SECTION:** chứa các thông tin về các nút, được biểu diễn dưới dạng node id, x, y
- **DEMAND SECTION:** chứa thông tin về yêu cầu hàng hóa của các khách hàng, được biểu diễn dưới dạng node id và demand  $b_i$
- **STATIONS COORD SECTION:** chứa các node id là các trạm sạc điện
- **DEPOT SECTION:** chứa node id của tổng kho

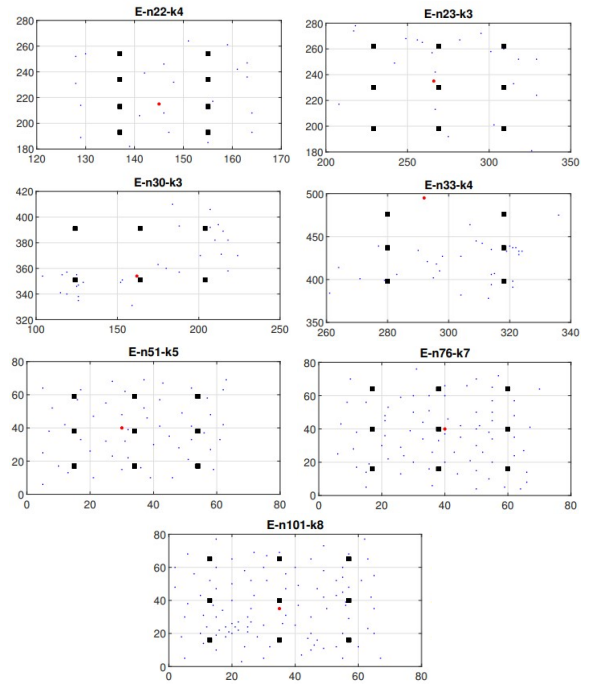
## IV. HEURISTIC

### A. Heuristic

Heuristic là một phương pháp tiếp cận giải quyết bài toán dựa trên các quy tắc, chiến lược hoặc kinh nghiệm thông thường. Thay vì tìm kiếm lời giải tối ưu toàn cục, heuristic tìm kiếm lời giải gần đúng trong một thời gian hợp lý. Các thuật toán heuristic thường dựa trên quyết định thông minh và sự đánh giá hướng dẫn để xây dựng lời giải. Heuristic thường đơn giản và nhanh chóng, nhưng không đảm bảo tìm kiếm lời giải tốt nhất. Các phương pháp heuristic phổ biến bao gồm: Greedy, Random, ...

### B. MetaHeuristic

Metaheuristic là một phương pháp tìm kiếm lời giải tối ưu cho các bài toán tối ưu hóa phức tạp. Metaheuristic không dựa trên thông tin cụ thể của bài toán, mà tìm kiếm trong không gian giải pháp bằng cách sử dụng các chiến lược tìm kiếm và luật điều chỉnh. Metaheuristic có khả năng tìm kiếm lời giải tốt hơn so với heuristic thông thường và thường được áp dụng cho



Hình 1: Một số ví dụ về dữ liệu.

các bài toán khó tính toán. Các phương pháp metaheuristic phổ biến bao gồm: Simulated Annealing, Genetic Algorithm, Particle Swarm Optimization, Ant Colony Optimization, và Tabu Search...

## V. PHƯƠNG PHÁP SỬ DỤNG TRONG EVRP

### A. Các ràng buộc

Một đồ thị được kết nối đầy đủ và có trọng số  $G = (V; A)$ , với  $V = 0 \cup I \cup F'$  là một chuỗi các nút ở trong đồ thị:

- $A = \{(i, j) | i, j \in V, i \neq j\}$  là danh sách các cung nối giữa các nút trong đồ thị
- $I$  là danh sách các khách hàng (điểm giao hàng)
- $F'$  là một danh sách các giá trị của  $\beta_i$  của các trạm sạc điện, với 0 là vị trí của tổng kho
- Mỗi cung sẽ có một giá trị  $d_{ij}$  là khoảng cách Euclid từ nút  $i$  tới nút  $j$
- Việc di chuyển bằng các cung sẽ tốn một khoảng pin  $hd_{ij}$ , với  $h$  là tỉ lệ tiêu thụ pin của các chiếc xe điện
- Mỗi khách hàng  $i \in I$  sẽ được đánh dấu bằng một lượng hàng hóa yêu cầu  $b_i$

Thống kê các ràng buộc của bài toán:

- Mỗi khách hàng sẽ được cung cấp bởi 1 xe duy nhất và chỉ được gặp khách hàng đó đúng 1 lần
- Mọi xe điện đều sẽ được cung cấp đầy đủ hàng hóa và lượng pin cần thiết khi bắt đầu ở tổng kho
- Với mỗi đoạn đường đi, tổng số lượng hàng hóa mà các khách hàng yêu cầu không vượt quá trọng lượng mà xe có thể mang theo
- Với mỗi đoạn đường đi, tổng số năng lượng tiêu thụ không được vượt quá lượng pin mà xe điện có thể chứa

Bảng I: Thông kê dữ liệu

stt	name	customers	depots	stations	routes	C	Q	h	UB
1	E-n22-k4	21	1	8	4	6000	94	1.2	384.67
2	E-n23-k3	22	1	9	3	4500	190	1.2	384.67
3	E-n30-k3	29	1	6	4	4500	178	1.2	384.67
4	E-n33-k4	32	1	6	4	8000	209	1.2	384.67
5	E-n51-k5	50	1	5	5	160	105	1.2	384.67
6	E-n76-k7	75	1	7	7	220	198	1.2	384.67
7	E-n101-k8	100	1	9	8	200	103	1.2	-
8	X-n143-k7	142	1	4	7	1190	2243	1.0	-
9	X-n214-k11	213	1	9	11	944	987	1.0	-
10	X-n351-k40	351	1	35	40	436	649	1.0	-
11	X-n459-k26	485	1	20	26	1160	929	1.0	-
12	X-n573-k30	572	1	6	30	210	1691	1.0	-
13	X-n685-k75	684	1	25	75	408	911	1.0	-
14	X-n749-k98	748	1	30	98	396	790	1.0	-
15	X-n819-k171	818	1	25	171	358	926	1.0	-
16	X-n916-k207	915	1	9	207	33	1591	1.0	-
17	X-n1001-k43	1000	1	9	43	131	1684	1.0	-

- Các xe điện sẽ luôn rời khỏi trạm sạc với lượng pin đầy đủ (tổng kho cũng tính là một trạm sạc)
- Trạm sạc (tính cả tổng kho) có thể được viếng thăm nhiều lần bởi các xe điện

Công thức:

- Hàm mục tiêu của EVRP

$$\min \left( \sum_{i \in V, j \in V, i \neq j} d_{ij} \cdot x_{ij} \right)$$

- Đường đi từ điểm giao hàng này đến điểm giao hàng khác

$$\sum_{j \in V, i \neq j} (x_{ij} = 1), \forall i \in I$$

- Đường đi từ trạm sạc này đến trạm sạc khác

$$\sum_{j \in V, i \neq j} (x_{ij} \leq 1), \forall i \in F'$$

- Số lượng các cung vào ra là như nhau

$$\sum_{j \in V, i \neq j} x_{ij} - \sum_{j \in V, i \neq j} x_{ji}, \forall i \in V$$

- Tải trọng giảm dần qua các đỉnh và không âm

$$u_j \leq u_i - b_j x_{ij} + C(1 - x_{ij}), \forall i, j \in V, i \neq j$$

$$0 \leq u_i \leq C, \forall i \in V$$

- Đảm bảo mức pin không âm

$$y_j \leq y_i - h d_{ij} x_{ij} + Q(1 - x_{ij}), \forall i \in F' \cup 0, \forall j \in V, i \neq j$$

$$y_j \leq Q - h d_{ij} x_{ij}, \forall i \in I, \forall j \in V, i \neq j$$

$$0 \leq y_i \leq Q, \forall i \in V$$

- Xác định một cung đã được đi qua hay chưa

$$x_{ij} \in \{0, 1\}, \forall i, j \in V, i \neq j$$

## B. Random Heuristic

Random heuristic là một phương pháp heuristics đơn giản được sử dụng trong bài toán Electric Vehicle Routing Problem (EVRP) để tạo ra các lời giải xấp xỉ. Phương pháp này dựa trên việc tạo ra các lời giải ngẫu nhiên và đánh giá chúng dựa trên các ràng buộc của bài toán EVRP. Các khái niệm quan trọng:

- Lời giải ngẫu nhiên: Random heuristic tạo ra các lời giải ngẫu nhiên bằng cách thay đổi thứ tự hoặc vị trí của các khách hàng trong tuyến đường.
- Đánh giá lời giải: Các lời giải được tạo ra bởi random heuristic được đánh giá dựa trên các ràng buộc của bài toán EVRP, bao gồm khả năng chở hàng, sức chứa của xe và khả năng tiêu thụ năng lượng.
- Khám phá không gian lời giải: Random heuristic khám phá không gian lời giải bằng cách tạo ra các lời giải khác nhau và kiểm tra tính khả thi và chất lượng của chúng.
- Quá trình cải thiện: Sau khi tạo ra các lời giải ngẫu nhiên, random heuristic có thể tiến hành quá trình cải thiện bằng cách thực hiện các phép biến đổi và đánh giá lại lời giải.
- Số lượng lời giải: Hiệu suất của random heuristic phụ thuộc vào số lượng lời giải được tạo ra. Việc lựa chọn số lượng lời giải phù hợp là quan trọng để đạt được kết quả tốt nhất trong thời gian hợp lý.

Ưu điểm

- Random heuristic là một phương pháp đơn giản và dễ thực hiện, không yêu cầu kiến thức chuyên sâu về bài toán và có thể được áp dụng một cách nhanh chóng.
- Random heuristic tạo ra các lời giải ngẫu nhiên, cho phép khám phá không gian lời giải rộng hơn
- Sự ngẫu nhiên trong random heuristic giúp tránh rơi vào cực tiểu cục bộ và tìm kiếm khám phá không gian lời giải.

Nhược điểm

- Random heuristic không đảm bảo tìm được lời giải tối ưu toàn cục cho bài toán EVRP. Các lời giải được tạo ra chỉ là xấp xỉ và có thể không tối ưu.

- Hiệu suất của random heuristic phụ thuộc vào số lượng lời giải được tạo ra. Nếu số lượng lời giải tạo ra ít, khả năng tìm ra lời giải tốt cũng sẽ giảm.

### C. Local Search

Local search là một phương pháp tìm kiếm cục bộ được sử dụng trong bài toán EVRP (Electric Vehicle Routing Problem) để cải thiện lời giải hiện tại bằng cách thay đổi một số lựa chọn gần lời giải hiện tại. Mục tiêu của local search là tìm ra một lời giải tốt hơn trong một khu vực cục bộ của không gian lời giải. Trong bài toán EVRP, local search có thể được áp dụng để tối ưu hóa lộ trình của các xe điện trong việc phục vụ các khách hàng và sử dụng các trạm sạc điện. Các khái niệm quan trọng:

- Neighborhood: Là tập hợp các lời giải lân cận của lời giải hiện tại. Neighborhood quyết định các phương pháp thay đổi và tìm kiếm lân cận trong quá trình local search.
- Objective function: Là hàm mục tiêu được sử dụng để đánh giá chất lượng của mỗi lời giải trong quá trình local search. Mục tiêu của local search là tối ưu hóa hàm mục tiêu này.
- Acceptance criterion: Là tiêu chí để quyết định xem một lời giải lân cận có được chấp nhận hay không. Điều kiện chấp nhận có thể dựa trên sự cải thiện của chất lượng lời giải hoặc một xác suất ngẫu nhiên.
- Perturbation: Là quá trình tạo ra sự đột biến hoặc nhiễu trong lời giải hiện tại để thoát khỏi cực tiểu cục bộ. Perturbation có thể được thực hiện bằng cách thay đổi một số lựa chọn trong lời giải hoặc tạo ra một lời giải mới hoàn toàn.
- Intensification và diversification: Intensification tập trung vào việc tìm kiếm các lời giải tốt hơn trong khu vực cục bộ, trong khi diversification tập trung vào việc khám phá không gian lời giải rộng hơn để thoát khỏi cực tiểu cục bộ. Kết hợp cả hai giai đoạn này có thể cải thiện khả năng tìm kiếm của local search.
- Stopping criterion: Là tiêu chí để dừng quá trình local search, ví dụ như đạt đủ số lần lặp hoặc không có cải thiện đáng kể sau một số lần lặp.

#### Ưu điểm

- Local search là một phương pháp tìm kiếm cục bộ đơn giản và dễ hiểu, không yêu cầu thông tin toàn cục về bài toán và không đòi hỏi lưu trữ một tập lời giải lớn.
- Local search tập trung vào việc cải thiện lời giải hiện tại bằng cách thay đổi một số lựa chọn gần lời giải hiện tại. Do đó, nó có khả năng tìm ra lời giải tốt trong một khoảng thời gian ngắn và phù hợp cho các bài toán có không gian lời giải lớn.
- Tránh bị rơi vào các tối ưu cục bộ và có khả năng tìm kiếm các tối ưu toàn cục.
- Local search có thể được kết hợp với các phương pháp khác để tạo thành các thuật toán tối ưu hóa phức tạp hơn. Nó có thể được tùy chỉnh và điều chỉnh để phù hợp với yêu cầu cụ thể của bài toán.

#### Nhược điểm

- Local search không đảm bảo tìm được lời giải tốt nhất toàn cục, mà chỉ tìm kiếm trong khu vực cục bộ gần lời giải hiện tại. Do đó, nó có thể rơi vào cực tiểu cục bộ và không thể thoát khỏi nó.
- Hiệu suất của local search có thể phụ thuộc vào lời giải ban đầu. Nếu lời giải ban đầu không tốt, local search có thể không tìm ra lời giải tốt nhất

### D. Simulated Annealing (SA)

Simulated Annealing là một thuật toán tối ưu hóa đặc biệt trong số các thuật toán tìm kiếm. Thuật toán này lấy cảm hứng từ quá trình làm mát kim loại trong công nghiệp. Nó là sự kết hợp giữa thuật toán Hill Climbing và cơ chế giảm nhiệt và được sử dụng để giải quyết các bài toán tối ưu hóa với không gian tìm kiếm lớn và hàm mục tiêu phức tạp.

Các khái niệm quan trọng:

- Hàm mục tiêu (Objective function): là một hàm số được sử dụng để đánh giá độ tốt của một giải pháp. Nó có thể là một hàm đơn giản hoặc phức tạp, tùy thuộc vào bài toán cụ thể mà ta đang giải quyết. Trong bài toán tối ưu ta thường muốn tìm ra giải pháp tốt nhất, tức là giá trị của hàm mục tiêu đạt giá trị nhỏ nhất hoặc lớn nhất
- Giải pháp (Solution): là một tập hợp các giá trị được đề xuất để giải quyết một bài toán. Với mỗi bài toán, giải pháp có thể được biểu diễn bằng các đối tượng khác nhau, ví dụ như một chuỗi các ký tự, một mảng các số, hoặc một đồ thị,... Mỗi giải pháp sẽ được đánh giá độ tốt của nó thông qua hàm mục tiêu. Hàm mục tiêu đạt được giá trị tối ưu nhất thì giải pháp đó là giải pháp tối ưu.
- Vùng lân cận (Neighborhood): là tập hợp các giải pháp khác mà có thể tạo ra bằng cách thực hiện một số thay đổi nhỏ trên giải pháp ban đầu. Việc xác định và khai thác vùng lân cận là một yếu tố quan trọng trong việc áp dụng các phương pháp tìm kiếm cục bộ để tìm kiếm giải pháp tốt hơn. Thông thường, vùng lân cận được xác định dựa trên tính chất cấu trúc của bài toán, và phải được thiết kế để có thể khả thi và hiệu quả cho việc khai thác
- Quá trình làm mát (Annealing process): quá trình điều chỉnh nhiệt độ của hệ thống giả lập để giảm dần độ "nóng" của nó theo một lịch trình xác định. Trong quá trình này, thuật toán chấp nhận các giải pháp kém hơn một cách ngẫu nhiên để tránh bị mắc kẹt ở tối ưu cục bộ, nhưng giảm dần tần suất chấp nhận các giải pháp kém hơn khi nhiệt độ giảm xuống.
- Biến nhiệt độ (Temperature): Biến nhiệt độ là một yếu tố quan trọng trong quá trình làm mát. Biến này quyết định mức độ chấp nhận của các giải pháp tối ưu tạm thời. Nó giảm dần trong quá trình tìm kiếm, điều này có nghĩa là các giải pháp tối ưu tạm thời được chấp nhận dễ dàng hơn ở những giai đoạn đầu của quá trình tìm kiếm, nhưng khi nhiệt độ giảm dần, thì xác suất chấp nhận của các giải pháp này cũng giảm dần
- Hàm giảm nhiệt độ (Cooling Schedule): Hàm giảm nhiệt độ là một hàm số quyết định tốc độ giảm nhiệt của hệ thống giả lập trong quá trình làm mát. Hàm giảm nhiệt

độ có thể là một hàm số đơn giản, chẳng hạn như hàm tuyến tính hoặc cũng có thể là một hàm số phức tạp hơn

- Hệ số giảm nhiệt (Cooling rate): Hệ số giảm nhiệt là một hằng số quyết định tốc độ giảm nhiệt của hệ thống giải lập trong quá trình làm mát.

#### Ưu điểm

- SA có thể giải quyết các bài toán không có giải pháp tối ưu hoặc không gian tìm kiếm quá lớn và phức tạp bởi đơn giản SA chỉ cần các kết quả ở mức chấp nhận được chứ không cần phải là tối ưu nhất.
- Không yêu cầu kiến thức chuyên môn sâu rộng về bài toán, mà chỉ cần một mô hình tìm kiếm đơn giản.
- Tránh bị rơi vào các tối ưu cục bộ và có khả năng tìm kiếm các tối ưu toàn cục.
- Dễ dàng tùy chỉnh và tối ưu hóa với các thông số như nhiệt độ ban đầu, hệ số giảm nhiệt, điều kiện dừng.
- Cho phép tìm kiếm giải pháp tối ưu mà không cần đánh giá toàn bộ không gian tìm kiếm, do đó có thể tiết kiệm thời gian và tài nguyên tính toán.
- Cho phép tìm kiếm trên không gian liên tục hoặc không liên tục, nên nó có thể được sử dụng cho nhiều loại bài toán.

#### Nhược điểm

- Việc lựa chọn các tham số là rất quan trọng, ảnh hưởng nhiều đến kết quả của thuật toán. SA có thể không tìm được giải pháp tối ưu khi bị kẹt trong các cực tiểu địa phương do sự lựa chọn kém của các tham số như tốc độ giảm nhiệt, nhiệt độ ban đầu dẫn đến nhiệt độ giảm quá nhanh hoặc quá chậm.
- Để đạt được kết quả tối ưu, SA yêu cầu thời gian chạy lâu hơn so với một số thuật toán tối ưu hóa khác.
- SA không giải quyết được các bài toán có ràng buộc chặt chẽ, vì nó không hỗ trợ các ràng buộc cứng.
- SA có thể bị ảnh hưởng bởi nhiễu hoặc sai số trong dữ liệu đầu vào, đặc biệt là trong các bài toán đa chiều phức tạp.
- SA không đảm bảo tìm được giải pháp tối ưu cho mọi bài toán, mà chỉ tìm được giải pháp tối ưu đối với một phạm vi giá trị tham số đủ lớn.

## VI. KẾT QUẢ THỰC NGHIỆM VÀ NHẬN XÉT

### A. Đánh giá

- Chạy code độc lập: nếu hàm heuristic của mình có mang tính chất ngẫu nhiên thì chạy 20 lần độc lập (mỗi lần chạy là một seed ngẫu nhiên từ 1 – 20); nếu không chỉ chạy một lần duy nhất. Biến giữ giá trị lần chạy là biến MAX\_TRIALS
- Số lần đánh giá tối đa là 25000n, trong đó  $n = |I| + 1 + |F|$  là kích thước của bài toán cụ thể. Một lần đánh giá có độ phức tạp  $O(n^2)$ .
- Thuật toán sẽ dừng lại khi chạy đủ số lần đánh giá tối đa, tức là số lần gọi objective function. Được định nghĩa là TERMINATION trong code.

- Công thức:

$$\bar{P} = \frac{1}{R} \sum_{i=1}^R P_i^*$$

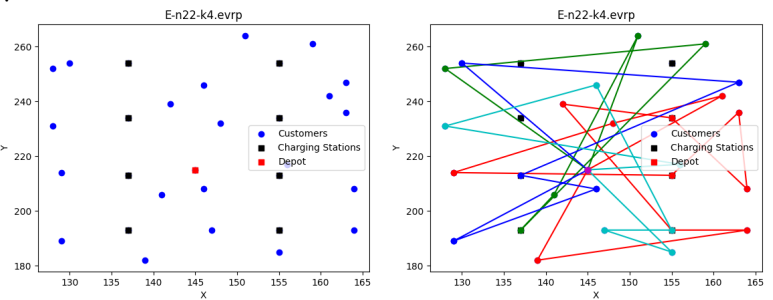
trong đó  $R$  là số lần chạy độc lập ( $R = 20$  khi cách tiếp cận mang tính ngẫu nhiên và  $R = 1$  nếu ngược lại),  $P_i^*$  là lời giải tốt nhất tìm được trong lần chạy thứ  $i$

### B. Kết quả thực nghiệm

Tiến hành thực nghiệm đối với Random Heuristic và Simulated Annealing:

- Nhóm 1: Chạy Random Heuristic riêng và ghi lại kết quả
- Nhóm 2: Chạy Random Heuristic trước để tìm lời giải ban đầu, sau đó chạy Simulated Annealing với lời giải này và thống kê kết quả
- Nhóm 3: Chạy Random Heuristic trước để tìm lời giải ban đầu, sau đó chạy Local Search với lời giải này và thống kê kết quả

\* Kết quả thực nghiệm ở Bảng II - Thống kê kết quả



Hình 2: Một số ví dụ về kết quả thực nghiệm.

### C. Nhận xét

Từ bảng kết quả ở phần trên, có thể thấy rằng kết quả của nhóm 2 tốt hơn so với nhóm 1 và nhóm 3, việc tối ưu hóa lời giải bằng Simulated Annealing đã thực sự giúp cải thiện kết quả so với lời giải được tìm ngẫu nhiên của Random Heuristic. Về thời gian chạy, nhóm 1 tìm ra lời giải nhanh chóng còn nhóm 2 do thời gian chạy của Simulated Annealing dài vì cần các công đoạn điều chỉnh tham số phù hợp để tìm ra lời giải tối ưu và còn bao gồm thời gian chạy của Random Heuristic nên nhóm 2 cần nhiều thời gian hơn nhóm 1, nhóm 3 cũng tương tự như nhóm 2, đều có hao phí thời gian do chạy Local Search cộng thêm Random Heuristic.

Lời giải của nhóm 1 vẫn chưa đáp ứng được tính tối ưu do tính ngẫu nhiên của nó, thêm vào đó giải pháp tìm được không thỏa mãn được tất cả các ràng buộc của bài toán nếu khi các tham chiếu về OPTIMAL\_VALUE của Bảng I vẫn cách xa nhau.

Lời giải của nhóm 2 khi các tham chiếu về OPTIMAL\_VALUE có tồn tại của Bảng I vẫn có sự chênh lệch nhưng nhỏ hơn so với nhóm 2, điều này xảy ra có thể là do lúc lập trình vẫn chưa đảm bảo tính chất đầy đủ của các ràng buộc và việc tinh chỉnh các tham số của thuật toán Simulated

Bảng II: Thống kê kết quả

	Random heuristic				Random heuristic + Simulated Annealing				Random heuristic + Local Search			
Name	Min	Max	Mean	stdev	Min	Max	Mean	stdev	Min	Max	Mean	stdev
E-n22-k4	554.6	645.5	606.2	22.96	270.9	372.5	334.5	24.0	554.6	645.5	606.2	22.96
E-n23-k3	972.2	1050.2	1014.6	22.75	270.9	372.5	334.5	24.0	972.2	1050.2	1014.6	22.75
E-n30-k3	1012.9	1134.4	1080.1	39.64	477.7	669.9	589.8	52.1	1012.9	1134.4	1080.1	39.64
E-n33-k4	1299.3	1366.4	1334.9	21.87	632.6	857.3	752.5	51.5	1299.3	1366.4	1334.9	21.87
E-n51-k5	1451.4	1558.3	1529.1	25.31	795.2	923.1	877.2	37.3	1451.4	1558.3	1529.1	25.31
E-n76-k7	2559.3	2665.6	2604.3	33.24	1018.1	1106.9	1064.9	24.7	2559.3	2665.6	2604.3	33.24
E-n101-k8	3568.3	3698.2	3613.1	44.02	1745.8	1852.7	1797.5	30.0	3568.3	3698.2	3613.1	44.02
X-n143-k7	76271.1	78131.5	77704.5	627.54	2430.4	2609.2	2566.0	41.9	76271.1	78131.5	77704.5	627.54
X-n214-k11	59243.3	59504.1	59321.5	122.58	61291.4	63377.5	62366.1	651.6	59243.3	59504.1	59321.5	122.58
X-n351-k40	165811.20	165811.20	165811.20	0	99413.7	103387.6	102281.2	968.7	165811.20	165811.20	165811.20	0
X-n459-k26	217985.5	217985.5	217985.5	0	146576.4	151983.9	150125.1	1358.1	217985.5	217985.5	217985.5	0

Annealing vẫn chưa hoàn toàn chính xác, khiến thuật toán này kẹt ở các cực trị địa phương, đi xa rời kết quả chính xác.

Lời giải của nhóm 3 giống với lời giải của nhóm 1 mặc dù đã áp dụng thêm Local Search, lý giải cho việc này có thể là do sự ảnh hưởng của lời giải ban đầu được tìm ra bởi Random Heuristic. Random Heuristic chỉ tạo ra một lời giải ban đầu dựa trên cách tiếp cận ngẫu nhiên và không đảm bảo tính tối ưu. Vì vậy, nếu lời giải ban đầu không tốt, tức là không đáp ứng được ràng buộc và gần như không có cải tiến, thì Local Search cũng không thể tìm ra lời giải tối ưu.

## VII. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Những phân tích trên đã cho thấy mức độ hiệu quả của từng phương pháp đối với bài toán Electric Vehicle Routing Problems, kết hợp được các phương pháp để cải thiện kết quả, biết được ưu nhược điểm của từng phương pháp để có thể vận dụng chúng một cách hiệu quả đối với những trường hợp khác nhau.

Ngoài ra đối với bài toán EVRP này, dù cho Random Heuristic + Simulated Annealing đã cho kết quả cao hơn 2 nhóm còn lại nhưng phần ràng buộc và việc tinh chỉnh các tham số vẫn còn là một vấn đề cần quan tâm để cải thiện hơn nữa độ chính xác.

### HƯỚNG PHÁT TRIỂN:

- Xây dựng lại lời giải ban đầu thành lời giải tối ưu hoặc gần tối ưu bằng cách áp dụng Greedy Heuristic kết hợp Clustering để gom cụm các người dùng lại, sau đó dựa vào constraints để đưa các trạm sạc vào trong lời giải khi cần thiết.
- Nâng cấp Local Search thành Variable Neighborhood Search với nhiều phiên bản Local Search khác nhau để tìm ra lời giải tốt hơn hoặc áp dụng các thuật toán

tối ưu khác như Genetic Algorithm (GA) hoặc Greedy randomized adaptive search procedure (GRASP)

## VIII. LỜI CẢM ƠN

Đồ án này là kết quả quá trình tìm hiểu và thực hành của các thành viên trong nhóm. Xin gửi lời cảm ơn đến thầy và các bạn đã giúp đỡ nhóm trong suốt thời gian học tập và hoàn thiện đồ án.

Tuy trong thời gian làm việc nhóm đã cố hết sức nhưng bài báo cáo vẫn chưa được hoàn thiện mà còn một số thiếu sót, chúng em rất mong nhận được các ý kiến, nhận xét bổ sung đến từ thầy để bài làm càng hoàn chỉnh hơn.

## REFERENCES

- [1] Benchmark Set for the IEEE WCCI-2020 Competition on Evolutionary Computation for the Electric Vehicle Routing Problem, 2020, CEC-12 competition on Electric Vehicle Routing Problem, Available at: <https://mavrovouniotis.github.io/EVRPcompetition2020/>
- [2] Erick Rodríguez-Esparza et al, 7 Jun 2022, A new Hyper-heuristic based on Adaptive Simulated Annealing and Reinforcement Learning for the Capacitated Electric Vehicle Routing Problem. Available at: <https://arxiv.org/pdf/2206.03185.pdf>
- [3] David Woller (DW), Vaclav Vávra (VV) and Viktor Kozak (VK), Electric vehicle routing problem - CW.FEL.CVUT.CZ. Available at: [https://cw.fel.cvut.cz/wiki/\\_media/courses/ko/evrp\\_kozak.pdf](https://cw.fel.cvut.cz/wiki/_media/courses/ko/evrp_kozak.pdf)
- [4] David Woller, Viktor Kozák and Miroslav Kulic, The grasp meta-heuristic for the electric vehicle routing problem. Available at: <http://imr.ciirc.cvut.cz/uploads/Papers-pdfs/Woller20mesas.pdf>