

BỘ THÔNG TIN VÀ TRUYỀN THÔNG  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Trần Việt Chương

NGHIÊN CỨU PHÁT TRIỂN THUẬT TOÁN  
METAHEURISTIC GIẢI BÀI TOÁN CÂY  
STEINER NHỎ NHẤT ĐỊNH HƯỚNG ỨNG  
DỤNG CHO THIẾT KẾ HỆ THỐNG MẠNG

Chuyên ngành: Hệ thống thông tin

Mã số: 9.48.01.04

TÓM TẮT LUẬN ÁN TIẾN SĨ KỸ THUẬT

Hà Nội – 2023

Công trình được hoàn thành tại:

**Học viện Công nghệ Bưu chính Viễn thông**

Người hướng dẫn khoa học: **1. PGS.TS. Hà Hải Nam**

**2. TS. Phan Tấn Quốc**

**Phản biện 1:** .....

**Phản biện 2:** .....

**Phản biện 3:** .....

Luận án được bảo vệ trước Hội đồng chấm luận án cấp Học viện tại:

Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: .....giờ..... ngày..... tháng..... năm 2023

Có thể tìm hiểu luận án tại thư viện:.....

## MỞ ĐẦU

### 1. Tính cấp thiết của đề tài

Việc kết nối một tập điểm cho trước với chi phí tối thiểu được xem như một trong những bài toán quan trọng nhất của thiết kế mạng truyền thông. Mạng truyền thông có thể được mô hình hóa bởi đồ thị vô hướng, liên thông và có trọng số. Do vậy, từ những năm 70 của thế kỷ trước, bài toán Cây Steiner (Steiner Tree Problem - *STP*) trong đồ thị hay bài toán Cây Steiner nhỏ nhất (Steiner Minimal Trees Problem - *SMT*), là bài toán tối ưu tổ hợp, đã được các nhà khoa học quan tâm nghiên cứu để áp dụng cho thiết kế mạng và nhiều ứng dụng quan trọng khác trong khoa học và kỹ thuật. Phần lớn các bài toán tối ưu là bài toán thuộc lớp *NP-hard*, không thể giải trong thời gian đa thức. Chỉ với bài toán quy mô nhỏ thì có thể giải bằng các phương pháp toán chính xác. Các bài toán khác được giải quyết bằng phương pháp xấp xỉ để tạo ra một giải pháp đủ tốt trong một thời gian hợp lý, đó là phương pháp heuristic và metaheuristic.

Do bản chất là bài toán tối ưu thuộc lớp *NP-hard*, nên cho đến nay, bài toán vẫn tiếp tục được các nhà khoa học quan tâm nghiên cứu nhằm làm phong phú hơn nữa cách giải và tìm lời giải hiệu quả hơn cho các ứng dụng thực tế, đặc biệt là ứng dụng trong thiết kế hệ thống mạng. Hiện tại, đã có hàng loạt thuật toán giải bài toán *SMT* được đề xuất, có thể chia chúng thành các hướng tiếp cận sau: Các thuật toán rút gọn đồ thị, các thuật toán cận tỉ lệ, các thuật toán tìm lời giải đúng, các thuật toán heuristic và metaheuristic. Trong khi thuật toán heuristic được áp dụng hiệu quả cho bài toán cụ thể, thì thuật toán metaheuristic được xem như khung thuật toán tổng quát có thể

áp dụng cho nhiều bài toán tối ưu. Cho đến hiện tại, hướng tiếp cận metaheuristic cho chất lượng lời giải tốt nhất trong số các thuật toán giải gần đúng.

## **2. Đối tượng và phạm vi nghiên cứu**

### **- Đối tượng nghiên cứu**

Bài toán *SMT*, các thuật toán heuristic và metaheuristic, các công trình công bố liên quan đến thuật toán heuristic, metaheuristic giải bài toán *SMT* và các ứng dụng của bài toán *SMT*.

### **- Phạm vi nghiên cứu**

Luận án này chỉ giới hạn nghiên cứu về các thuật toán heuristic, metaheuristic giải bài toán Cây Steiner với khoảng cách ngẫu nhiên.

## **3. Mục tiêu nghiên cứu**

Mục tiêu của luận án này là nghiên cứu phát triển một số thuật toán dạng heuristic và metaheuristic nhằm giải bài toán *SMT* một cách hiệu quả và định hướng ứng dụng cho thiết kế hệ thống mạng.

## **4. Phương pháp nghiên cứu**

Luận án kết hợp phương pháp nghiên cứu lý thuyết và thực nghiệm. Trên cơ sở khảo sát các công trình khoa học liên quan đến bài toán *SMT* và ứng dụng trong thiết kế mạng. Các thuật toán đề xuất mới hoặc cải tiến được thực nghiệm, đánh giá dựa trên hệ thống dữ liệu thực nghiệm chuẩn và mở rộng. Việc đánh giá so sánh hiệu năng các thuật toán được dựa trên độ phức tạp thuật toán, chất lượng lời giải và thời gian chạy thực nghiệm.

## **5. Nội dung nghiên cứu**

Đề xuất một số thuật toán heuristic, metaheuristic mới hoặc cải tiến, dựa trên ý tưởng tìm cây đường đi ngắn nhất, cây

khung nhỏ nhất, các chiến lược tìm kiếm lân cận và lược đồ các metaheuristic cơ bản để giải bài toán *SMT* và định hướng ứng dụng cho thiết kế hệ thống mạng.

## 6. Những đóng góp chính của luận án

Sau đây là những đóng góp chính của luận án:

- Đề xuất hai thuật toán heuristic mới: *SPT-Steiner* và *PD-Steiner* giải bài toán *SMT* trong trường hợp đồ thị thưa.

- Đề xuất hai thuật toán heuristic cải tiến: *i-SPT-Steiner* và *i-PD-Steiner* giải bài toán *SMT* trong trường hợp đồ thị thưa kích thước lớn lên đến 100000 đỉnh.

- Đề xuất mới ba thuật toán metaheuristic dạng cá thể, quần thể giải bài toán *SMT* gồm: Thuật toán *Bees-Steiner*, thuật toán tìm kiếm lân cận biến đổi *VNS* và thuật toán tìm kiếm leo đồi Hill climbing search (*HCSMT*).

- Ngoài ra, luận án cũng đề xuất thêm 2 chiến lược tìm kiếm lân cận: Tham lam và có xác suất, sử dụng trong các khung thuật toán metaheuristic nhằm nâng cao hơn nữa chất lượng lời giải.

## 7. Ý nghĩa khoa học và thực tiễn

Việc đề xuất thuật toán dạng heuristic, metaheuristic giải bài toán *SMT* có ý nghĩa quan trọng; một mặt, nhằm giải quyết những bài toán ứng dụng đặt ra trong thực tiễn đặc biệt trong thiết kế hệ thống mạng; mặt khác, còn là cơ sở để giải quyết một số bài toán tối ưu thuộc lớp *NP-hard* khác.

## 8. Bố cục luận án

Bố cục của luận án được tổ chức thành 3 chương và phần kết luận, cụ thể như sau:

- **Chương 1:** Trình bày tổng quan về cơ sở lý thuyết bài toán *SMT* với các nội dung: Một số định nghĩa, định lý cơ bản liên

quan; các dạng của bài toán *SMT*; sơ lược một số hướng tiếp cận. Tiếp theo, khảo sát một số thuật toán metaheuristic giải bài toán *SMT*, cụ thể như: Giới thiệu sơ đồ một số thuật toán metaheuristic thường gặp như: thuật toán local search, thuật toán leo đồi, thuật toán tìm kiếm lân cận biến đổi, thuật toán bầy ong; các tiêu chí đánh giá chất lượng thuật toán metaheuristic; khảo sát kết quả một số thuật toán heuristic, metaheuristic hiện biết giải bài toán *SMT*; định hướng ứng dụng bài toán *SMT* trong thiết kế hệ thống mạng và cuối cùng là giới thiệu hệ thống dữ liệu thực nghiệm chuẩn và mở rộng cho bài toán.

- **Chương 2:** Đề xuất 2 thuật toán heuristic mới *SPT-Steiner*, *PD-Steiner* giải bài toán *SMT* trong trường hợp đồ thị thưa và 2 thuật toán heuristic cải tiến *i-SPT-Steiner*, *i-PD-Steiner* giải bài toán *SMT* trong trường hợp đồ thị thưa kích thước lớn.

- **Chương 3:** Đề xuất 3 thuật toán metaheuristic giải bài toán *SMT*; các thuật toán này dựa trên khung thuật toán metaheuristic gồm: Thuật toán Bees, Hill climbing search và Variable neighborhood search. Luận án cũng đề xuất thêm một số chiến lược tìm kiếm lân cận cho bài toán *SMT*, đồng thời phân tích ưu nhược điểm của từng thuật toán cụ thể và qua đó định hướng áp dụng vào thực tế cho từng thuật toán đề xuất.

Trong phần **Kết luận**, luận án trình bày những kết quả đạt được và định hướng phát triển cho nghiên cứu trong tương lai khi áp dụng kết quả luận án vào thực tiễn.

# CHƯƠNG 1. TỔNG QUAN VỀ BÀI TOÁN CÂY STEINER NHỎ NHẤT VÀ ĐỊNH HƯỚNG ỨNG DỤNG CHO THIẾT KẾ HỆ THỐNG MẠNG

## 1.1. Cơ sở lý thuyết

### 1.1.1. Một số định nghĩa

#### **Định nghĩa 1.1. Cây Steiner**

Cho  $G = (V(G), E(G))$  là một đơn đồ thị vô hướng liên thông, có trọng số không âm trên cạnh, trong đó  $V(G)$  là tập gồm  $n$  đỉnh,  $E(G)$  là tập gồm  $m$  cạnh,  $w(e)$  là trọng số của cạnh  $e$  với  $e \in E(G)$ . Cho  $L$  là tập con các đỉnh của  $V(G)$ , cây  $T$  đi qua tất cả các đỉnh trong  $L$  được gọi là Cây Steiner ứng với tập  $L$  trên đồ thị  $G$ . Tập  $L$  được gọi là tập *terminal*, các đỉnh thuộc tập  $L$  được gọi là đỉnh *terminal*. Đỉnh thuộc cây  $T$  mà không thuộc tập  $L$  được gọi là đỉnh *Steiner*.

#### **Định nghĩa 1.2. Chi phí Cây Steiner**

Cho  $T = (V(T), E(T))$  là một Cây Steiner của đồ thị  $G$ . Chi phí của cây  $T$ , ký hiệu là  $C(T)$ , là tổng trọng số của các cạnh thuộc cây  $T$ , tức là  $C(T) = \sum_{e \in E(T)} w(e)$ .

#### **Định nghĩa 1.3. Cây Steiner nhỏ nhất**

Cho đồ thị  $G$  được mô tả như trên, bài toán tìm Cây Steiner có chi phí nhỏ nhất được gọi là bài toán Cây Steiner nhỏ nhất (Steiner minimal trees problem - *SMT*); hoặc được gọi ngắn gọn là bài toán Cây Steiner (Steiner trees problem).

#### **Định lý 1.1. Định lý về số đỉnh Steiner**

Cho đồ thị  $G$  và tập *terminal*  $L$ , Cây Steiner  $T$  của  $L$  có  $p$  đỉnh thì số đỉnh Steiner của  $T$  không vượt quá  $p - 2$

#### **Định nghĩa 1.4. Cạnh cầu Steiner**

Cho đồ thị  $G$  và tập *terminal*  $L$ , cạnh  $e_{uv}$  được gọi là cạnh cầu Steiner của  $G$  nếu khi loại cạnh  $e_{uv}$  thì tập các đỉnh *terminal*  $L$  không cùng thuộc về một thành phần liên thông.

### **Định nghĩa 1.5. Đồ thị rút gọn Steiner**

Cho đồ thị  $G$  và tập *terminal*  $L$ ,  $G'$  được gọi là đồ thị rút gọn Steiner của  $G$  nếu số đỉnh và số cạnh của  $G'$  nhỏ hơn hoặc bằng số đỉnh và số cạnh của  $G$  và trong  $G'$  tồn tại ít nhất một Cây Steiner nhỏ nhất ứng với tập *terminal*  $L$  của đồ thị  $G$ .

#### **1.1.2. Một số dạng của bài toán SMT**

Bài toán SMT hiện được nghiên cứu ở các dạng sau đây:

- Thứ nhất là bài toán Cây Steiner với khoảng cách Euclide.
- Thứ hai là bài toán Cây Steiner với khoảng cách chữ nhật.
- Thứ ba là bài toán Cây Steiner với khoảng cách ngẫu nhiên (đây cũng chính là dạng bài toán nghiên cứu của luận án).

#### **1.1.3. Tổng quan các nghiên cứu liên quan đến bài toán SMT**

Có nhiều hướng tiếp cận giải bài toán SMT, cụ thể như sau:

- Các thuật toán rút gọn đồ thị
- Các thuật toán tìm lời giải đúng
- Các thuật toán gần đúng cận tỉ lệ
- Các thuật toán heuristic
- Các thuật toán metaheuristic

### **1.2. Tiếp cận thuật toán metaheuristic giải bài toán SMT**

#### **1.2.1. Thuật toán metaheuristic**

Thuật toán metaheuristic được xem như khung thuật toán tổng quát có thể áp dụng cho nhiều bài toán tối ưu. Thuật toán này sử dụng nhiều heuristic kết hợp với các kỹ thuật phụ trợ nhằm khai phá không gian tìm kiếm lời giải của bài toán.

#### **1.2.2. Tính tăng cường và tính đa dạng**



Hai yếu tố quan trọng nhất của một thuật toán metaheuristic là tính tăng cường (intensification) và tính đa dạng (diversification). Tăng cường hóa nhằm mục đích khai thác sâu hơn các vùng không gian lời giải tiềm năng để hy vọng tìm được lời giải có chất lượng tốt hơn. Đa dạng hóa nhằm khai phá những vùng không gian tìm kiếm mới, tránh việc tìm kiếm rơi vào bẫy tối ưu cục bộ.

### **1.2.3. Tiêu chí đánh giá chất lượng thuật toán metaheuristic**

Chất lượng của một thuật toán metaheuristic được đánh giá qua chất lượng lời giải và thời gian tính. Thời gian tính ngoài việc phụ thuộc vào độ phức tạp của thuật toán còn phụ thuộc vào môi trường thực nghiệm. Đánh giá độ phức tạp của thuật toán thường dựa vào tài nguyên bộ nhớ và thời gian mà thuật toán sử dụng. Hiện nay, người sử dụng quan tâm nhiều hơn đến yếu tố thời gian.

### **1.2.4. Phân tích các thành phần của một thuật toán metaheuristic**

- Khởi tạo lời giải ban đầu
- Chiến lược tăng cường hóa lời giải
- Chiến lược đa dạng hóa lời giải
- Điều kiện dừng của thuật toán metaheuristic

### **1.2.5. Giới thiệu sơ đồ một số thuật toán metaheuristic thường gặp**

- Thuật toán *Local Search*
- Thuật toán *Hill Climbing Search*
- Thuật toán tìm kiếm lân cận biến đổi
- Thuật toán *Bees cơ bản*

## **1.3. Khảo sát một số thuật toán tiêu biểu giải bài toán SMT**

Khảo sát, phân tích đánh giá một số thuật toán heuristic và metaheuristic giải bài toán *SMT* hiện biết như thuật toán: *SPH*, *Heu*, *NB*, *PB*, *TS*, *PGA*.

#### **1.4. Định hướng ứng dụng kết quả nghiên cứu cho thiết kế hệ thống mạng**

##### **1.4.1. Giới thiệu bài toán quy hoạch mạng**

- **Bài toán thứ nhất:** Là bài toán nâng cấp, mở rộng hệ thống mạng đã có.

- **Bài toán thứ hai:** Là bài toán xây dựng mới hệ thống mạng.

##### **1.4.2. Giới thiệu bài toán quy hoạch mạng**

Một số mô hình định hướng ứng dụng bài toán *SMT* trong việc thiết kế hệ thống mạng.

- *Mô hình 1: Thiết kế hệ thống mạng cục bộ*
- *Mô hình 2: Cung cấp dịch vụ mạng riêng ảo*
- *Mô hình 3: Bài toán truyền thông đa hướng*
- *Mô hình 4: Thiết kế vi mạch VLSI*
- *Mô hình 5: Bài toán kết nối nhóm*
- *Mô hình 6: Bài toán Steiner nhiều pha*
- *Mô hình 7: Bài toán mạng Steiner nhiều pha*
- *Mô hình 8: Bài toán phát sinh loài*

#### **1.5. Lựa chọn dữ liệu thực nghiệm**

Luận án sử dụng 78 bộ dữ liệu là các đồ thị thưa trong hệ thống dữ liệu thực nghiệm chuẩn dùng cho bài toán Cây Steiner tại địa chỉ: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/steininfo.html>, các đồ thị này có tối đa 2500 đỉnh. Ngoài ra, luận án đề xuất thêm 80 bộ dữ liệu là các đồ thị thưa kích thước lớn lên đến 100000 đỉnh: *steinf*, *steing*, *steinh*, *steini*. Các đồ thị này được sinh từ một cây khung ngẫu nhiên.

## CHƯƠNG 2: ĐỀ XUẤT THUẬT TOÁN HEURISTIC GIẢI BÀI TOÁN CÂY STEINER NHỎ NHẤT

### 2.1. Giới thiệu hướng tiếp cận heuristic giải bài toán *SMT*

Trong chương này, luận án đề xuất hai thuật toán heuristic mới giải bài toán *SMT* và hai heuristic cải tiến giải bài toán *SMT* trong trường hợp đồ thị thưa kích thước lớn lên đến 100000 đỉnh. Các thuật toán đề xuất này được cài đặt thực nghiệm và so sánh chất lượng với thuật toán heuristic tốt hiện biết *MST-Steiner* của Bang Ye Wu và Kun-Mao Chao.

### 2.2. Thuật toán *SPT-Steiner*

---

#### Algorithm 2.1: *SPT-Steiner* (Shortest Path Tree-Steiner)

---

**Input:** Đồ thị  $G = (V(G), E(G))$ ;

**Output:** Cây Steiner nhỏ nhất  $T$ ;

1. Tìm các cây đường đi ngắn nhất xuất phát tại các đỉnh thuộc tập  $V(G)$ :  $SPT_1, SPT_2, \dots, SPT_n$
2. **for** ( $T \in \{SPT_1, SPT_2, \dots, SPT_n\}$ ) **do**
3. Với mỗi đỉnh treo  $u \in T$ , nếu  $u \notin L$  thì xóa cạnh chứa đỉnh  $u$  khỏi  $E(T)$ , xóa đỉnh  $u$  trong  $V(T)$  và cập nhật bậc của đỉnh kề với đỉnh  $u$  trong  $T$ .
4. Lặp lại bước 3 đến khi  $T$  không còn thay đổi (bước này gọi là bước xóa các cạnh dư thừa); sau bước này thu được các  $SPT_i'$  tương ứng với các  $SPT_i$ .
5. **end for**
6. Tìm một cây  $SPT_i'$  có tổng trọng số các cạnh nhỏ nhất là  $SPT_{min}$ ;
7. **return**  $SPT_{min}$ ;

Thuật toán *SPT-Steiner* có độ phức tạp thời gian tính là:  $O(mn \log n)$ .

### 2.3. Thuật toán *PD-Steiner*

---

#### Algorithm 2.2: *PD-Steiner* (Prim + Dijkstra-Steiner)

---

**Input:** Đồ thị  $G = (V(G), E(G))$ ;

**Output:** Cây Steiner nhỏ nhất  $T$ ;

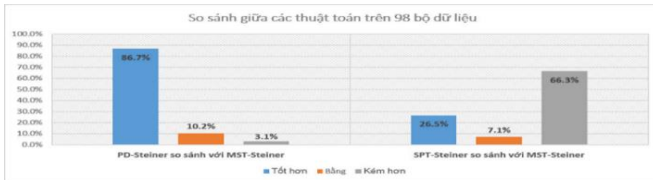
1. Tìm các đường đi ngắn nhất từ các đỉnh thuộc tập  $L$  đến các đỉnh thuộc tập  $V$ ;
2. Chọn một đỉnh  $u \in L$ ; đặt  $T = \{u\}$ ;
3. **while** ( $T$  chưa chứa tất cả các đỉnh thuộc tập  $L$ ) **do**
4. Từ mỗi đỉnh  $v \in L$  và  $v$  chưa thuộc  $T$ , tìm các đường đi ngắn nhất từ đỉnh  $v$  đến các đỉnh  $z \in T$ ;
5. Chọn ra một đường đi ngắn nhất  $P$ ;
6. Kết nạp các đỉnh và các cạnh trên đường đi  $P$  vào cây  $T$ ;
7. **end while**
8. **Output**  $T$ .

Thuật toán *PD-Steiner* có độ phức tạp thời gian tính là:  $O(|L|n^2)$ .

### 2.4. Thực nghiệm và đánh giá

Đánh giá chung trên toàn bộ 98 bộ dữ liệu; thuật toán *SPT-Steiner*, *PD-Steiner* cho chất lượng lời giải tốt hơn, bằng, kém hơn so với thuật toán *MST-Steiner* lần lượt là: (26.5%, 7.1%, 66.3%), (86.7%, 10.2%, 3.1%) số bộ dữ liệu. Kết quả so sánh giữa thuật toán *SPT-Steiner* và *PD-Steiner* với *MST-Steiner* được minh họa như Hình 2.1.

Thuật toán *SPT-Steiner* và *PD-Steiner* có thời gian chạy chậm hơn so với thuật toán *MST-Steiner*.



Hình 2.1. So sánh giữa các thuật toán trên 98 bộ dữ liệu

## 2.5. Cải tiến thuật toán heuristic giải bài toán *SMT* trong trường hợp đồ thị thừa kích thước lớn

### 2.5.1. Thuật toán *i-SPT-Steiner*

---

#### Algorithm 2.3: *i-SPT-Steiner*

---

**Input:** Đồ thị  $G = (V(G), E(G))$ ;

**Output:** Cây Steiner nhỏ nhất  $T$ ;

1. Sử dụng Thuật toán **Dial** tìm các cây đường đi ngắn nhất xuất phát tại các đỉnh thuộc tập terminal  $L$ :  $SPT_1, SPT_2, \dots, SPT_L$
2. **for** ( $T \in \{SPT_1, SPT_2, \dots, SPT_L\}$ ) **do**
3. Với mỗi đỉnh treo  $u \in T$ , nếu  $u \notin L$  thì xóa cạnh chứa đỉnh  $u$  khỏi  $E(T)$ , xóa đỉnh  $u$  trong  $V(T)$  và cập nhật bậc của đỉnh kề với đỉnh  $u$  trong  $T$ .
4. Lặp lại bước 3 đến khi  $T$  không còn thay đổi (bước này gọi là bước xóa các cạnh dư thừa); sau bước này, thu được các cây  $SPT_{i'}$  tương ứng với các cây  $SPT_i$ .
5. **end for**
6. Tìm một cây  $SPT_{i'}$  có tổng trọng số các cạnh nhỏ nhất là  $SPT_{min}$ ;
7. **return**  $SPT_{min}$ ;

Thuật toán *i-SPT-Steiner* có độ phức tạp thời gian tính là:  $O(|L|(m + nC))$ .

### 2.5.2. Thuật toán *i-PD-Steiner*

---

#### Algorithm 2.4: *i-PD-Steiner*

---

**Input:** Đồ thị  $G = (V(G), E(G))$ ;

**Output:** Cây Steiner nhỏ nhất  $T$ ;

1. Chọn một đỉnh  $u \in L$ , đặt  $V(T) = \{u\}$ ;
2. **for** mỗi đỉnh  $v \in L$  **do**

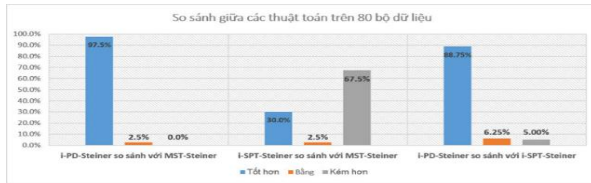
3. Tìm cây đường đi ngắn nhất xuất phát từ đỉnh  $v$ ;  
(Ở bước này sử dụng Thuật toán **Dial** để tìm)
4. **end for**
5. **while** ( $T$  chưa chứa tất cả các đỉnh thuộc tập  $L$ ) **do**
6. Từ mỗi đỉnh  $v \in L$  và  $v \notin V(T)$ , Chọn đường đi ngắn nhất  $P$  xuất phát từ đỉnh  $v$  đến đỉnh  $z \in V(T)$ , với  $z$  là đỉnh kết thúc của đường đi ngắn nhất trong tất cả các đường đi ngắn nhất xuất phát từ  $v$  đến các đỉnh của  $V(T)$ ;
7. Kết nạp các đỉnh và các cạnh trên đường đi  $P$  vào cây  $T$ ;
8. **end while**
9. **Output**  $T$ .

Thuật toán ***i*-PD-Steiner** có độ phức tạp là:  $O(|L|/\max(m + nC, |L|/|V(T)|))$ .

## 2.6. Thực nghiệm và đánh giá

Đánh giá trên tổng số 80 bộ dữ liệu, thuật toán *i*-PD-Steiner, *i*-SPT-Steiner cho lời giải có chất lượng tốt hơn, tương đương, kém hơn thuật toán *MST*-Steiner lần lượt là: (97.5%, 2.5%, 0.0%), (30.0%, 2.5%, 67.5%); thuật toán *i*-PD-Steiner cho lời giải có chất lượng tốt hơn, tương đương, kém hơn thuật toán *i*-SPT-Steiner là: (88.75%, 6.25%, 5.0%) số bộ dữ liệu.

Kết quả so sánh giữa các thuật toán *i*-SPT-Steiner, *i*-PD-Steiner và *MST*-Steiner được minh họa bằng biểu đồ như Hình 2.2.



Hình 2.2. So sánh giữa các thuật toán trên 80 bộ dữ liệu

Thuật toán *i*-PD-Steiner có thời gian chạy nhanh hơn so với thuật toán MST-Steiner và thuật toán *i*-SPT-Steiner. Thuật toán *i*-SPT-Steiner có thời gian chạy chậm hơn so với thuật toán MST-Steiner.

## 2.7. Đánh giá các thuật toán thông qua độ phức tạp

Thông qua Bảng 2.1 ta thấy, thuật toán SPT-Steiner có độ phức tạp thời gian lớn nhất, thuật toán *i*-PD-Steiner có độ phức tạp thời gian nhỏ nhất, qua đó cung cấp thêm thông tin về tính hiệu quả của các thuật toán này.

Bảng 2.1. Độ phức tạp thời gian của các thuật toán

Thuật toán	Độ phức tạp thời gian
MST-Steiner	$O(n/L ^2)$
SPT-Steiner	$O(mn \log n)$
PD-Steiner	$O( L n^2)$
<i>i</i> -SPT-Steiner	$O( L (m + nC))$
<i>i</i> -PD-Steiner	$O( L  \max(m + nC,  L / V(T) ))$
Các thuật toán được xếp theo độ phức tạp thời gian giảm dần:	<b>SPT-Steiner, <i>i</i>-SPT-Steiner, PD-Steiner, MST-Steiner, <i>i</i>-PD-Steiner</b>

## CHƯƠNG 3: ĐỀ XUẤT THUẬT TOÁN METAHEURISTIC GIẢI BÀI TOÁN CÂY STEINER NHỎ NHẤT

### 3.1. Giới thiệu hướng tiếp cận metaheuristic giải bài toán SMT

Thuật toán metaheuristic sử dụng nhiều heuristic kết hợp với các kỹ thuật phụ trợ nhằm khai phá không gian tìm kiếm.

### 3.2. Khởi tạo lời giải ban đầu

- Khởi tạo Cây Steiner theo một heuristic
- Khởi tạo Cây Steiner ngẫu nhiên
- Khởi tạo Cây Steiner dựa vào xác suất

### 3.3. Các chiến lược tìm kiếm Cây Steiner lân cận

#### **Định nghĩa 3.1.** *1-lân cận của Cây Steiner $T$*

Cho đồ thị  $G$  và  $T$  là một Cây Steiner của  $G$ . Ta gọi *1-lân cận* của Cây Steiner  $T$  là tập tất cả các Cây Steiner của đồ thị  $G$  sai khác với  $T$  đúng một cạnh. Nếu  $T'$  là một Cây Steiner thuộc *1-lân cận* của  $T$  thì ta nói  $T$  và  $T'$  là *1-lân cận* với nhau.

#### **Định nghĩa 3.2.** *$k$ -lân cận của Cây Steiner $T$*

Cho đồ thị  $G$  và  $T$  là một Cây Steiner của nó. Ta gọi  *$k$ -lân cận* của Cây Steiner  $T$  là tập tất cả các Cây Steiner của đồ thị  $G$  sai khác với  $T$  không quá  $k$  cạnh. Nếu  $T'$  là một Cây Steiner thuộc  *$k$ -lân cận* của  $T$  thì ta nói  $T$  và  $T'$  là  *$k$ -lân cận* với nhau.

➤ Tiếp theo, luận án sẽ trình bày một số chiến lược tìm kiếm Cây Steiner lân cận hiện biết.

- Chiến lược chèn cạnh - xóa cạnh
- Chiến lược tìm lân cận tốt hơn
- Chiến lược tìm lân cận ngẫu nhiên
- Chiến lược tìm lân cận Node-base
- Chiến lược tìm lân cận Path-based
- Chiến lược tìm kiếm lân cận tham lam
- Chiến lược tìm kiếm lân cận có xác suất

➤ Các hàm sử dụng trong thuật toán: **InitPopulation** ( $V(G)$ ,  $E(G)$ ,  $N$ ): Hàm tạo quần thể ban đầu; **SortPopulation** ( $P$ ,  $N$ ,  $h$ ,  $p$ ): Hàm sắp xếp các cá thể trong quần thể; **NeighSearch** ( $T_i$ ,  $k_i$ ): Hàm tìm kiếm Cây Steiner lân cận; **RandSearch** ( $T_i$ ,  $k_i$ ): Hàm tìm kiếm Cây Steiner ngẫu nhiên.



### 3.4. Thuật toán Bees giải bài toán SMT

Thuật toán này *Bees-Steiner* giải bài toán SMT

---

#### Algorithm 3.1: *Bees-Steiner* algorithm

---

**Input:** Đồ thị  $G = (V(G), E(G))$ ;

**Output:** Cây Steiner có chi phí nhỏ nhất tìm được  $T_{best}$ ;

1. *InitPopulation* ( $V(G), E(G), N$ ); // Tạo quần thể  $P$  gồm các Cây Steiner  $T_1, T_2, \dots, T_N$
2. **While** (điều kiện dừng chưa thỏa) **do**
3.     *SortPopulation* ( $P, N, h, p$ );
4.     Cập nhật  $T_{best} = T_1$ ;
5.     **For**  $i = 1..h$  **do**
6.         *NeighSearch* ( $T_i, k_1$ );
7.     **For**  $i = h + 1..p$  **do**
8.         *NeighSearch* ( $T_i, k_2$ );
9.     **For**  $i = p + 1..N$  **do**
10.         *RandSearch* ( $T_i, k_3$ );
11.     **end while**
12.     Cập nhật cá thể  $T_{best}$ ;
13. **Return**  $T_{best}$ .

Độ phức tạp của thuật toán **Bees-Steiner** là:  $O(N(N \log(N) + V(hk_1 + (p - h)k_2 + (N - p)k_3)))$ .

### 3.5. Thuật toán tìm kiếm lân cận biến đổi giải bài toán SMT

---

#### Algorithm 3.2: *VNS* algorithm

---

**Input:** Đồ thị  $G = (V(G), E(G))$ ;

**Output:** Cây Steiner có chi phí nhỏ nhất;

1.  $T$  là cây khung ngẫu nhiên được tạo bởi thuật toán tựa Prim;

2. Xóa các cạnh dư thừa trong  $T$  để thu được Cây Steiner;
3. **while** (điều kiện dừng chưa thỏa) **do**
4. Lần lượt thực hiện hai chiến lược tìm kiếm lân cận Node-based và Path-based;
5. Trong quá trình thực hiện các chiến lược tìm kiếm lân cận trên, ghi nhận lại lời giải tốt nhất;
6. Khi thực hiện một chiến lược tìm kiếm lân cận, nếu tìm được kỉ lục mới thì quay trở lại thực hiện thuật toán VNS từ đầu (sau vòng lặp while); ngược lại, chuyển sang chiến lược tìm kiếm lân cận tiếp theo;
7. Thuật toán VNS kết thúc khi điều kiện dừng được thỏa; điều kiện dừng được lựa chọn trong thuật toán này là  $10*n$ , với  $n$  là số đỉnh của đồ thị
8. **end while**
9. Trả về lời giải tốt nhất tìm được.

Độ phức tạp của thuật toán VNS là:  $O(N^2(E \log(E)))$ .

### 3.6. Thuật toán Hill climbing search giải bài toán SMT

---

#### Algorithm 3.3: HCSMT algorithm

---

**Input:** Đồ thị  $G = (V(G), E(G))$ ;

**Output:** Cây Steiner có chi phí nhỏ nhất tìm được  $T_{best}$ ;

1.  $stop = false$ ;

2.  $d = 0$ ; //  $d$  là số lần tăng cường

3.  $T$  là Cây Steiner được khởi tạo ngẫu nhiên;

4.  $T_{best} = T$ ; // lưu lại Cây Steiner tốt nhất trước

khi thực hiện chiến lược đa dạng hóa

5. **while** ( $d < N$ ) **do** //  $N$  là số lần lặp định trước

6.      $stop = true$ ;

```

7.    $S = E - E(T)$ ;
8.   for (với mỗi cạnh  $e \in S$ ) do
9.        $min = +\infty$ ;
10.      if (cạnh  $e$  có 2 đỉnh  $(u, v) \in T$ ) then
11.           $F = T \cup \{e\}$ ;
12.          Xác định chu trình Cycle trong  $F$  chứa  $e$ ;
13.          for (với mỗi cạnh  $e' \in \text{Cycle}$ ) do
14.              if ( $C(F - e') < min$ ) then
15.                   $min = C(F - e')$ ;
16.                  Ghi nhận  $T' = F - e'$ ;
17.              end if
18.          end for
19.          if ( $min < C(T)$ ) then
20.               $T = T'$ ;
21.               $stop = \text{false}$ ;
22.          end if
23.      end if
24.  end for
25.  if ( $stop$ ) then
26.       $d++$ ; //tăng số biến đếm tăng cường
27.      if ( $C(T) < C(T_{best})$ )  $T_{best} = T$ ;
28.      while (Thỏa điều kiện đa dạng hóa) do
29.          Loại bỏ một số cạnh ngẫu nhiên của
Cây Steiner đang xét;
30.          Chọn ngẫu nhiên một số cạnh trong
các cạnh còn lại của đồ thị  $G$  cho đến khi cây  $T$  liên
thông trở lại (điều kiện cạnh thêm vào là: Phải có
đỉnh thuộc  $T$  mới được thêm vào để tránh trường hợp

```

thêm vào quá nhiều lần nhưng  $T$  không liên thông trở lại). Sử dụng định nghĩa  $k$ -lân cận ở trên;

31. Nếu thêm quá nhiều lần mà không làm cho  $T$  liên thông trở lại, thì tiếp tục đa dạng hóa lại với các cạnh khác;

32.            **end while**

33.            **end if**

34. **end while**

35. **return**  $T_{best}$ .

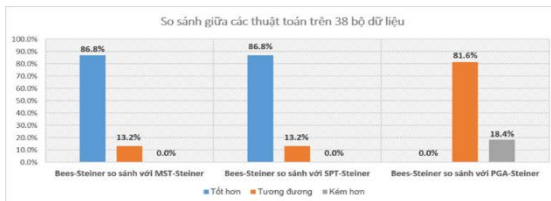
Độ phức tạp của thuật toán **HCSMT** là:  $O(N(E \log(E) + EV))$ .

### 3.7. Thử nghiệm và đánh giá các thuật toán metaheuristic giải bài toán **SMT**

#### 3.7.1. Thuật toán **Bees-Steiner**

Đánh giá chung trên tổng số 38 bộ dữ liệu, thuật toán **Bees-Steiner** cho chất lượng lời giải tốt hơn, tương đương và kém hơn thuật toán: **MST-Steiner**, **SPT-Steiner**, **PGA-Steiner** lần lượt là: (86.8%, 13.2%, 0.0%), (86.8%, 13.2%, 0.0%), (0.0%, 81.6%, 18.4%) số bộ dữ liệu.

Kết quả so sánh giữa thuật toán **Bees-Steiner** với các thuật toán: **MST-Steiner**, **SPT-Steiner** và **PGA-Steiner** được minh họa bằng biểu đồ như Hình 3.1.

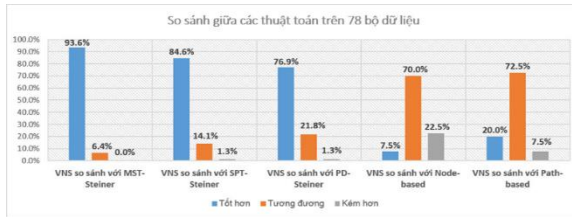


Hình 3.1. So sánh giữa các thuật toán trên 38 bộ dữ liệu

#### 3.7.2. Thuật toán tìm kiếm lân cận biến đổi (**VNS**)

Đánh giá chung trên 78 bộ dữ liệu là các đồ thị thưa trong hệ thống dữ liệu thực nghiệm chuẩn; thuật toán *VNS* cho lời giải chất lượng tốt hơn, tương đương và kém hơn thuật toán *MST-Steiner* lần lượt là: (93.6%, 6.4%, 0.0%) số bộ dữ liệu; thuật toán *VNS* cho lời giải chất lượng tốt hơn, tương đương và kém hơn thuật toán *SPT-Steiner* lần lượt là: (84.6%, 14.1%, 1.3%) số bộ dữ liệu; thuật toán *VNS* cho lời giải chất lượng tốt hơn, tương đương và kém hơn thuật toán *PD-Steiner* lần lượt là: (76.9%, 21.8%, 1.3%) số bộ dữ liệu; thuật toán *VNS* cho lời giải chất lượng tốt hơn, tương đương và kém hơn thuật toán *Node-based* lần lượt là: (7.5%, 70.0%, 22.5%) số bộ dữ liệu; thuật toán *VNS* cho lời giải chất lượng tốt hơn, tương đương và kém hơn thuật toán *Path-based* lần lượt là: (20.0%, 72.5%, 7.5%) số bộ dữ liệu.

Kết quả so sánh giữa thuật toán *VNS* với các thuật toán: *MST-Steiner*, *SPT-Steiner*, *PD-Steiner*, *Node-based* và *Path-based* được minh họa bằng biểu đồ như Hình 3.2.



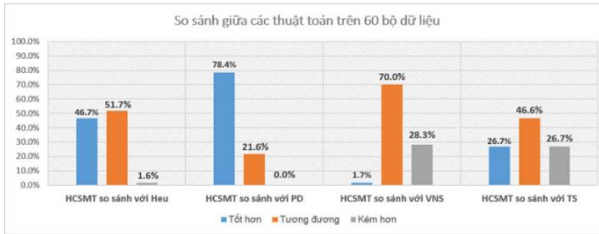
Hình 3.2. So sánh giữa các thuật toán trên 78 bộ dữ liệu

### 3.7.3. Thuật toán Hill Climbing Search (*HCSMT*)

Đánh giá chung trên tổng số 60 bộ dữ liệu; thuật toán *HCSMT* cho chất lượng lời giải tốt hơn, tương đương và kém hơn thuật toán Heu lần lượt là: (46.7%, 51.7%, 1.6%) số bộ dữ liệu; thuật toán *HCSMT* cho chất lượng lời giải tốt hơn, tương

đương và kém hơn thuật toán *PD-Steiner* lần lượt là: (78.4%, 21.6%, 0.0%) số bộ dữ liệu; thuật toán *HCSMT* cho chất lượng lời giải tốt hơn, tương đương và kém hơn thuật toán *VNS* lần lượt là: (1.7%, 70%, 28.3%) số bộ dữ liệu; thuật toán *HCSMT* cho chất lượng lời giải tốt hơn, tương đương và kém hơn thuật toán *TS* lần lượt là: (26.7%, 46.6%, 26.7%) số bộ dữ liệu.

Kết quả so sánh giữa thuật toán *HCSMT* với các thuật toán: *Heu*, *PD-Steiner*, *VNS* và *TS* được minh họa bằng biểu đồ như Hình 3.3.



Hình 3.3. So sánh giữa các thuật toán trên 60 bộ dữ liệu

### 3.8. Đánh giá các thuật toán thông qua độ phức tạp

Thông qua Bảng 3.1 ta thấy, thuật toán *Bees-Steiner* có độ phức tạp thời gian lớn nhất, thuật toán *HCSMT* có độ phức tạp thời gian nhỏ nhất, qua đó cung cấp thêm thông tin về tính hiệu quả của các thuật toán.

Bảng 3.1. Độ phức tạp thời gian của các thuật toán

Thuật toán	Độ phức tạp thời gian
<i>Bees-Steiner</i>	$O(N(N \log(N) + V(hk_1 + (p - h)k_2 + (N - p)k_3)))$
<i>VNS</i>	$O(N^2(E \log(E)))$
<i>HCSMT</i>	$O(N(E \log(E) + EV))$
Các thuật toán được xếp theo độ phức tạp thời gian giảm dần:	<b><i>Bees-Steiner, VNS, HCSMT</i></b>

## KẾT LUẬN

Hiện tại, có rất nhiều hướng tiếp cận để giải bài toán *SMT*, có thể chia thành hai nhóm như sau: Giải chính xác, đối với những bài toán có kích thước nhỏ; và giải gần đúng đối với những bài toán có kích thước lớn, trong đó có phát triển các thuật toán heuristic và metaheuristic để giải bài toán này. Vì vậy bài toán *SMT* đòi hỏi phải được nghiên cứu một cách đa chiều và toàn diện. Từ các nghiên cứu đó, phạm vi ứng dụng của bài toán sẽ được chỉ ra một cách rõ ràng và cụ thể hơn trong thực tế.

Dựa trên phương pháp nghiên cứu được sử dụng như thông qua một số cơ sở lý thuyết toán học, xây dựng phát triển thuật toán và cài đặt thực nghiệm để phân tích, so sánh đánh giá, kết hợp với các công cụ thống kê. Các kết quả chính của luận án được trình bày trong Chương 2 và Chương 3.

### 1. Các đóng góp chính của luận án

*1.1. Đề xuất hai thuật toán heuristic mới giải bài toán SMT trong trường hợp đồ thị thưa, các thuật toán đề xuất được cài đặt thực nghiệm và cho kết quả tốt hơn trên một số bộ dữ liệu khi so sánh với heuristic công bố trước đó, cụ thể là:*

- **Đóng góp thứ nhất:** Đề xuất thuật toán heuristic *SPT-Steiner* giải bài toán *SMT*.

- **Đóng góp thứ hai:** Đề xuất thuật toán heuristic *PD-Steiner* giải bài toán *SMT*.

*1.2. Đề xuất hai thuật toán heuristic cải tiến giải bài toán SMT trong trường hợp đồ thị thưa kích thước lớn lên đến 100000 đỉnh, các thuật toán đề xuất được thực nghiệm và cho kết quả tốt hơn trên một số bộ dữ liệu khi so sánh với heuristic công bố trước đó, cụ thể là:*

- **Đóng góp thứ ba:** Đề xuất thuật toán heuristic cải tiến *i-SPT-Steiner* giải bài toán *SMT*.

- **Đóng góp thứ tư:** Đề xuất thuật toán heuristic cải tiến *i-PD-Steiner* giải bài toán *SMT*.

1.3. Đề xuất ba thuật toán metaheuristic dạng cá thể, quần thể giải bài toán *SMT* trong trường hợp đồ thị thưa, các thuật toán đề xuất được cài đặt thực nghiệm và cho kết quả tốt hơn trên một số bộ dữ liệu khi so sánh với các heuristic, metaheuristic công bố trước đó, cụ thể là:

- **Đóng góp thứ năm:** Đề xuất thuật toán metaheuristic dạng quần thể *Bees-Steine* giải bài toán *SMT*.

- **Đóng góp thứ sáu:** Đề xuất thuật toán metaheuristic dạng cá thể *VNS* giải bài toán *SMT*.

- **Đóng góp thứ bảy:** Đề xuất thuật toán metaheuristic dạng cá thể *Hill Climbing Search (HCSMT)* giải bài toán *SMT*.

1.4. Đề xuất các chiến lược tìm kiếm lân cận sử dụng trong lược đồ thuật toán metaheuristic nhằm nâng cao hiệu quả của thuật toán metaheuristic trong việc giải bài toán *SMT*.

- **Đóng góp thứ tám:** Luận án đã nghiên cứu chi tiết đặc tính của các thuật toán metaheuristic trong việc giải bài toán *SMT*. Đóng góp mới của luận án là sự tổng hợp các thuật toán trên với các chiến lược tìm kiếm lân cận trên cơ sở đặc tính của bài toán *SMT*.

- **Đóng góp thứ chín:** Dựa trên kết quả thực nghiệm các thuật toán được phát triển, luận án đề xuất hướng áp dụng từng thuật toán vào các loại đồ thị. Qua đó làm sâu sắc hơn, phong phú thêm việc giải hiệu quả bài toán *SMT* và định hướng ứng dụng cho thiết kế hệ thống mạng.

## 2. Những nội dung nghiên cứu tiếp theo



Hiện nay, các hướng tiếp cận giải chính xác hoặc gần đúng bài toán *SMT* nói chung đang được các nhà khoa học, các tập toàn công nghệ lớn quan tâm, đầu tư nghiên cứu và phát triển. Do là bài toán tối ưu thuộc lớp *NP-hard*, nên mỗi phương án mới đưa ra có nhiều ưu điểm và ngày càng hiệu quả hơn.

Theo hướng này, trong thời gian tới luận án sẽ tiếp tục phát triển các nội dung sau:

- Tiếp tục phát triển các chiến lược tăng cường hóa, đa dạng hóa lời giải (các chiến lược tìm kiếm lân cận), áp dụng vào sơ đồ các thuật toán đề xuất nhằm nâng cao hơn nữa chất lượng lời giải cho các thuật toán.

- Tiếp tục cải tiến cấu trúc dữ liệu, kỹ thuật lập trình khi cài đặt thực nghiệm thuật toán với hy vọng rút ngắn thời gian chạy các thuật toán khi làm việc với các đồ thị thưa kích thước lớn (cỡ 100000 đỉnh).

- Dựa vào sơ đồ các thuật toán đề xuất và các thuật toán hiện biết của tác giả khác, phát triển thuật toán mới dạng metaheuristic để giải quyết một số bài toán cây khung truyền thông tối ưu, hoặc các bài toán định tuyến trên đồ thị thuộc lớp *NP-hard* khác.

- Dựa vào các kết quả nghiên cứu đạt được, xây dựng một ứng dụng giải bài toán *SMT* và áp dụng kết quả đạt được vào thiết kế một hệ thống mạng cụ thể trong thực tế.

## CÁC CÔNG TRÌNH NGHIÊN CỨU CỦA TÁC GIẢ

### TẠP CHÍ KHOA HỌC

[CT1] **Trần Việt Chương**, Phan Tấn Quốc, Hà Hải Nam (2017), Thuật toán Bees giải bài toán Cây Steiner nhỏ nhất

trong trường hợp đồ thị thưa, *Tạp chí Khoa học Công nghệ thông tin và Truyền thông*, Học viện Công nghệ Bưu chính Viễn thông, Bộ Thông tin và Truyền thông, ISSN: 2525-2224, Số 02 & 03 (CS.01) 2017, trang 24-28.

[CT2] **Tran Viet Chuong**, Ha Hai Nam (2018), A variable neighborhood search algorithm for solving the Steiner minimal tree problem in sparse graphs, *EAI Endorsed Transactions on Context-aware Systems and Applications*, ISSN: 2409-0026, Issue 15, Vol 5, EAI 2018 (<https://eudl.eu/issue/casa/5/15>), pp. 1-6.

[CT3] **Trần Việt Chương**, Phan Tấn Quốc, Hà Hải Nam (2020), Thuật toán tìm kiếm Hill climbing giải bài toán Cây Steiner nhỏ nhất, *Chuyên san Các công trình nghiên cứu, phát triển và ứng dụng Công nghệ thông tin và Truyền thông, Tạp chí Thông tin và Truyền thông, Bộ Thông tin và Truyền thông*, Tập 2020, Số 1, tháng 6, trang 1-8.

[CT4] **Trần Việt Chương**, Phan Tấn Quốc, Hà Hải Nam (2021), Đề xuất chiến lược tìm kiếm lân cận cho bài toán Cây Steiner nhỏ nhất, *Tạp chí Khoa học Công nghệ thông tin và Truyền thông*, Học viện Công nghệ Bưu chính Viễn thông, Bộ Thông tin và Truyền thông, ISSN: 2525-2224, Số 01 (CS.01) 2021, trang 90-97.

[CT5] **Trần Việt Chương**, Phan Tấn Quốc, Hà Hải Nam (2022), Improving the Heuristic Algorithms to Solve a Steiner-minimal-tree Problem in Large Size Sparse Graphs, *Chuyên san các Công trình nghiên cứu, phát triển và ứng dụng Công nghệ thông tin và Truyền thông, Tạp chí Thông tin và Truyền thông, Bộ Thông tin và Truyền thông*, Tập 2022, Số 1, tháng 3, pp. 43-52.

## **HỘI NGHỊ KHOA HỌC QUỐC TẾ**

[CT6] **Trần Việt Chương**, Phan Tấn Quốc, Hà Hải Nam (2017), Đề xuất một số thuật toán Heuristic giải bài toán Cây Steiner nhỏ nhất, *Kỷ yếu Hội nghị Khoa học Công nghệ Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR)*; Trường Đại học Đà Nẵng, ngày 17-

18/08/2017, ISBN: 978-604-913-614-6, NXB Khoa học và Kỹ thuật, trang 138-147.

[CT7] **Tran Viet Chuong**, Ha Hai Nam (2018), A Variable Neighborhood Search Algorithm for Solving the Steiner Minimal Tree Problem, *Kỷ yếu Hội nghị khoa học Quốc tế EAI lần thứ 7 - ICCASA và lần thứ 4 - ICTCC 2018*, chủ đề Context-Aware Systems and Applications, and Nature of Computation and Communication, LNICST 266, Springer, pp. 218-225.

[CT8] **Trần Việt Chương**, Phan Tấn Quốc, Hà Hải Nam (2019), Khảo sát một số thuật toán giải bài toán Cây Steiner nhỏ nhất trong trường hợp đồ thị thưa, *Kỷ yếu Hội thảo Quốc gia lần thứ XXII về Một số vấn đề chọn lọc của Công nghệ thông tin và Truyền thông*, chủ đề Chuyển đổi số điều hành Kinh tế - Xã hội trong cách mạng công nghiệp 4.0, do Viện CNTT và Trường Đại học Thái Bình tổ chức ngày 28-29/6/2019 tại TP Thái Bình, NXB Khoa học và Kỹ thuật, trang 132-137.