

ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



ĐỒ ÁN TỐT NGHIỆP
CHUYÊN NGÀNH KHOA HỌC MÁY TÍNH

**Giải thuật di truyền tính toán đa mục tiêu
trên mạng cảm biến không dây**

Họ và tên : Trần Sơn Tùng
MSSV : 20154282
Lớp : AS - K60
GVHD : TS. Trần Vĩnh Đức

Hà Nội, tháng 6 năm 2020

Thẻ nhiệm vụ tốt nghiệp

Thông tin sinh viên

- **Họ và tên:** Trần Sơn Tùng
- **Email:** tung.ts154282@sis.hust.edu.vn
- **Số điện thoại:** 0705869458
- **Lớp học:** AS K60
- **Chương trình:** Kỹ sư
- **Thời hạn:** Từ tháng 1 năm 2019 đến tháng 6 năm 2020

Mục tiêu chính của đề án

1. Nghiên cứu về tính toán đa mục tiêu trên mạng cảm biến không dây.
2. Nghiên cứu và áp dụng giải thuật di truyền để tính toán đa mục tiêu trên mạng cảm biến không dây.

Nhiệm vụ cụ thể của đề án

1. Nghiên cứu về các mục tiêu cần quan tâm khi thiết kế mạng cảm biến không dây.
2. Áp dụng giải thuật di truyền và giải thuật di truyền sắp xếp không trội để tính toán đa mục tiêu trên mạng cảm biến không dây.
3. Thực nghiệm, đối chiếu và phân tích kết quả thu được.

Cam kết của sinh viên

Tôi là *Trần Sơn Tùng* cam đoan rằng nội dung trong đề án này là của tôi dưới sự hướng dẫn của TS. Trần Vĩnh Đức.

Các đề xuất và kết quả trong đề án này đều là xác thực và nguyên bản.

Hà Nội, ngày 1 tháng 6 năm 2020

Tác giả đề án

Trần Sơn Tùng

Xác nhận về mức độ hoàn thiện đề án và cho phép đề án được bảo vệ bởi giáo viên hướng dẫn.

.....
.....
.....
.....
.....

Hà Nội, ngày 1 tháng 6 năm 2020

Người hướng dẫn

TS. Trần Vĩnh Đức

Lời cảm ơn

Đồ án này là thành quả nghiên cứu dưới sự hướng dẫn của TS. Trần Vĩnh Đức. Sau 3 kì nghiên cứu khoa học dưới sự hướng dẫn của thầy, tôi đã được làm quen với các kiến thức về mạng cảm biến và giải thuật tiến hoá nói riêng và các khái niệm cơ bản trong nghiên cứu khoa học nói chung. Đồ án tốt nghiệp này là tổng kết kết quả nghiên cứu trong 3 kì đó.

Tôi muốn giành lời cảm ơn trước hết cho thầy Trần Vĩnh Đức, người đã tận tâm hướng dẫn trong 3 kì học để tôi có thể tiến hành nghiên cứu và hoàn thành đồ án. Tôi cũng muốn cảm ơn PGS.TS Huỳnh Thị Thanh Bình và ThS. Nguyễn Thị Tâm đã hỗ trợ rất nhiều khi tôi gặp khó khăn trong việc tìm hiểu các kiến thức mới. Cuối cùng, tôi muốn cảm ơn các thầy cô thuộc viện Công nghệ Thông tin nói riêng và trường Đại học Bách Khoa nói chung đã truyền đạt kiến thức, kinh nghiệm sống quý báu trong suốt 5 năm học, giúp tôi trưởng thành được như ngày hôm nay.

Hà Nội, ngày 1 tháng 6 năm 2020
Trần Sơn Tùng

Tóm tắt

Ý tưởng sử dụng học thuyết tiến hoá trong tính toán tối ưu hình thành từ những năm 1950 - 1960, ban đầu là những nỗ lực cá nhân độc lập, được biết đến dưới các tên gọi Lập trình tiến hoá (Evolutionary programming) bởi Lawrence Fogel, Giải thuật di truyền (Genetic algorithms) bởi John Holland, Lập trình di truyền (Genetic programming) bởi John Koza, và Chiến lược tiến hoá (Evolutionary strategies) bởi Hans-Paul Schwefel, sau đó thống nhất dưới tên gọi chung là Giải thuật tiến hoá (Evolutionary Algorithm - EA). EA được biết tới như một cơ sở nhằm phát triển các lớp giải thuật mô phỏng quy trình lựa chọn tự nhiên. Cùng được thiết kế với một chu trình là lựa chọn quần thể ban đầu, sử dụng các phép lai ghép, đột biến để thu được cá thể của thế hệ sau tốt hơn thế hệ trước, và đánh giá mỗi cá thể để lựa chọn giữ lại những cá thể phù hợp, các thuật toán có điều chỉnh ở một số bước khác nhau để phù hợp với vấn đề cần giải quyết. Nhờ tính tổng quát hoá mà các giải thuật EA có thể được áp dụng cho rất nhiều các lớp bài toán tối ưu khác nhau, mà tối ưu hoá các mục tiêu khi thiết kế một mạng cảm biến không dây là một trong số đó.

Giải thuật di truyền (Genetic Algorithm - GA), một giải thuật phát triển với cơ sở từ EA, được sử dụng lần đầu để tối ưu hoá năng lượng sử dụng trong mạng cảm biến không dây, sau đó được áp dụng để tối ưu hoá các mục tiêu như độ bao phủ, thời gian sống, vị trí đặt các nút cảm biến. Qua thời gian, đi liền với sự phát triển và mở rộng của mạng cảm biến không dây trong đời sống, các giải thuật khác cũng dần được phát triển và sử dụng. Thực nghiệm đã chứng minh tính hiệu quả của các giải thuật EA khi cần thu được các kết quả chấp nhận được trong thời gian cho phép hay tối ưu hoá nhiều mục tiêu cùng lúc.

Đồ án có cấu trúc bao gồm 4 chương, với mục tiêu chính là nghiên cứu và áp dụng hai giải thuật tiến hoá khác nhau để tối ưu hoá các mục tiêu trong mạng cảm biến không dây.

- **Chương 1** Giới thiệu tổng quan về mạng cảm biến không dây.
- **Chương 2** Giới thiệu các mục tiêu cần tối ưu trên mạng cảm biến không dây.
- **Chương 3** Áp dụng giải thuật di truyền và giải thuật di truyền sắp xếp không trội giải bài toán tối ưu đa mục tiêu.
- **Chương 4** Trình bày kết quả thực nghiệm, phân tích và đánh giá hiệu quả của giải thuật di truyền và giải thuật di truyền sắp xếp không trội.

Từ điển chú giải

Notation	Description
BFS	duyet theo chiều rộng
DFS	duyet theo chiều sâu
EA	giải thuật tiến hoá
EWD	quy trình giải mã dịch chuyển khung
GA	giải thuật di truyền
JPSO	giải thuật tối ưu bầy đàn mô phỏng bước nhảy
NSGA-II	giải thuật di truyền sắp xếp không trội
PSO	giải thuật tối ưu bầy đàn
WSN	mạng cảm biến không dây

Danh sách bảng

3.1	Tính chất các địa hình	43
3.2	Tham số sử dụng trong mô phỏng địa hình	43
3.3	Tham số cài đặt trong các giải thuật	43
3.4	Bảng tổng hợp kết quả thực nghiệm.	46

Danh sách hình vẽ

1.1	Các loại mạng cảm biến không dây	15
1.2	Cấu trúc một nút cảm biến	17
1.3	Các giao thức mạng trong thực tế	18
1.4	Đồ thị đơn vô hướng	21
1.5	Cây trong lý thuyết đồ thị.	24
3.1	Mã hóa một cá thể	35
3.2	Áp dụng toán tử lai ghép lên hai cá thể cha mẹ. Các cạnh xuất hiện trong cả cha và mẹ được tô đỏ ở cá thể con.	37
3.3	Ví dụ về toán tử đột biến	38
3.4	Giá trị trung bình hàm mục tiêu của GA và giải thuật tối ưu bầy đàn mô phỏng bước nhảy (Jumping Particle Optimization Problem - JPSO) trên phân phối Gamma.	44
3.5	Giá trị trung bình hàm mục tiêu của GA và JPSO trên phân phối chuẩn.	44
3.6	Giá trị trung bình hàm mục tiêu của GA và JPSO trên phân phối đều.	44
3.7	Gamma - T1	47
3.8	Gamma - T2	47
3.9	Gamma - T3	47
3.10	Gamma - T4	47
3.11	Gamma - T5	48
3.12	Gamma - T6	48
3.13	Gamma - T7	48
3.14	Gamma - T8	48
3.15	Gamma - T9	48
3.16	Gamma - T10	48
3.17	No - T1	49
3.18	No - T2	49
3.19	No - T3	49
3.20	No - T4	49

3.21	No - T5	50
3.22	No - T6	50
3.23	No - T7	50
3.24	No - T8	50
3.25	No - T9	50
3.26	No - T10	50
3.27	Uniform - T1	51
3.28	Uniform - T2	51
3.29	Uniform - T3	51
3.30	Uniform - T4	51
3.31	Uniform - T5	52
3.32	Uniform - T6	52
3.33	Uniform - T7	52
3.34	Uniform - T8	52
3.35	Uniform - T9	52
3.36	Uniform - T10	52

Mục lục

Thẻ nhiệm vụ tốt nghiệp	2
Lời cảm ơn	4
Tóm tắt	5
Từ điển chú giải	7
1 Giới thiệu	13
1.1 Mạng cảm biến không dây	14
1.1.1 Định nghĩa và phân loại	14
1.1.2 Nút cảm biến	15
1.1.3 Các giao thức truyền tin của mạng cảm biến	16
1.1.4 Ứng dụng của mạng cảm biến không dây	18
1.1.5 Khó khăn	19
1.2 Lý thuyết đồ thị	20
1.2.1 Định nghĩa và biểu diễn đồ thị	20
1.2.2 Thuật toán duyệt đồ thị	22
1.2.3 Thuật toán xây dựng cây khung nhỏ nhất	23
1.2.4 Lý thuyết về cây trong đồ thị	23
1.3 Tổng kết	25
2 Tối ưu đa mục tiêu trên cây tổng hợp dữ liệu	26
2.1 Mô hình năng lượng	27
2.2 Phát biểu bài toán	27
2.3 Nghiên cứu liên quan	30
3 Giải thuật di truyền và giải thuật di truyền sắp xếp không trội cho bài toán đa mục tiêu trên cây tổng hợp dữ liệu	34
3.1 Giải thuật di truyền	35
3.1.1 Biểu diễn cá thể	35

3.1.2	Khởi tạo quần thể	36
3.1.3	Toán tử lai ghép	36
3.1.4	Toán tử đột biến	37
3.2	Giải thuật di truyền sắp xếp không trội	38
3.2.1	Cơ sở lý thuyết	38
3.2.2	Áp dụng cho bài toán tối ưu đa mục tiêu	39
3.3	Hàm phạt	42
3.4	Thực nghiệm	42
3.4.1	Mô tả dữ liệu	42
3.4.2	Tham số cố định	43
3.4.3	Kết quả thực nghiệm của GA và JPSO trên các phân phối khác nhau	43
3.4.4	Tổng hợp kết quả thực nghiệm của GA và JPSO	46
3.4.5	Kết quả thực nghiệm của giải thuật di truyền sắp xếp không trội (Non-dominated Sorting Genetic Algorithm II - NSGA-II)	47

Chương 1

Giới thiệu

Mạng cảm biến không dây (Wireless Sensor Network - WSN) xuất hiện lần đầu vào năm 1950, biết tới với tên gọi là Hệ thống giám sát âm thanh (Sound Surveillance System). Đây là hệ thống được quân đội Mỹ triển khai nhằm truy tìm tàu ngầm của quân Xô-viết. Hệ thống này vẫn còn tồn tại cho tới ngày nay, nhưng phục vụ cho mục đích quan sát thế giới động vật dưới biển và tình trạng núi lửa. Nhằm hỗ trợ việc triển khai mạng cảm biến không dây, Chương trình nghiên cứu hệ thống phòng thủ cấp cao (Defense Advanced Research Projects Agency) đã triển khai Mạng cảm biến phân tán (Distributed Sensor Network). Sự ra đời của chương trình này cùng với việc thúc đẩy nghiên cứu từ các học viện và trường đại học lớn, công nghệ mạng cảm biến không dây sớm trở thành một lĩnh vực nghiên cứu khoa học. Chính phủ và các trường đại học bắt đầu sử dụng mạng cảm biến cho việc đo đạc chất lượng không khí, phát hiện thảm họa như cháy rừng, ngập lụt hay giám sát cấu trúc. Các tập đoàn lớn cũng bắt đầu sử dụng mạng cảm biến cho việc phân bố năng lượng, tự động hóa trong nhà máy. Mặc dù cung cấp những tính năng và hiệu suất tuyệt vời, chi phí đắt đỏ và cấu hình phức tạp khiến cho mạng cảm biến chỉ được ứng dụng trong quân sự, nghiên cứu khoa học hoặc các ngành công nghiệp nặng. Nhằm mở rộng thị phần của mạng cảm biến để đáp ứng yêu cầu từ người sử dụng phổ thông, như các hộ gia đình hay các công ty nhỏ, lẻ, một số tổ chức nghiên cứu chuyên môn đã được thành lập. Mục tiêu chính của họ là giảm giá thành và năng lượng tiêu thụ, đồng thời đơn giản hóa quy trình phát triển và bảo trì. Điều này có tác dụng kích thích thị trường dịch chuyển từ các ứng dụng quân sự trên chiến trường hay các thí nghiệm phức tạp trong phòng nghiên cứu thành các ứng dụng thực tiễn và thường thấy trong cuộc sống. [4]

Trong khi thiết kế mạng cảm biến không dây, một vấn đề hay gặp phải là sự xung đột giữa các mục tiêu. Một ví dụ cho vấn đề này là khi tìm cách tăng thời gian sống của mạng qua việc giảm năng lượng tiêu thụ trên mỗi

nút, một số nút trung gian được sử dụng nhằm thu hẹp khoảng cách truyền tin. Việc sử dụng các nút trung gian này làm tăng độ trễ của đường truyền. Một số mục tiêu thường được cân nhắc khi thiết kế mạng cảm biến bao gồm: thời gian sống của mạng, độ bao phủ, và độ trễ. Chương đầu của đồ án đóng vai trò như cái nhìn tổng quan về mạng cảm biến không dây. Phần 1 sẽ bao gồm cấu tạo của một mạng cảm biến, các loại mạng khác nhau, giao thức mạng thường được sử dụng, các ứng dụng và khó khăn hiện thời khi triển khai mạng cảm biến không dây. Nhằm hỗ trợ cho việc giải thích chi tiết trong các chương sau, phần 1.2 sẽ là những kiến thức về khoa học máy tính nhằm giải quyết thách thức trong mạng cảm biến không dây được đề ra ở chương 2, cụ thể bao gồm: lý thuyết đồ thị và một số thuật toán sử dụng trên đồ thị, kiến thức về cấu trúc dữ liệu cây và bài toán cây Steiner để mô hình hóa bài toán mạng cảm biến không dây trong thực tế.

1.1 Mạng cảm biến không dây

1.1.1 Định nghĩa và phân loại

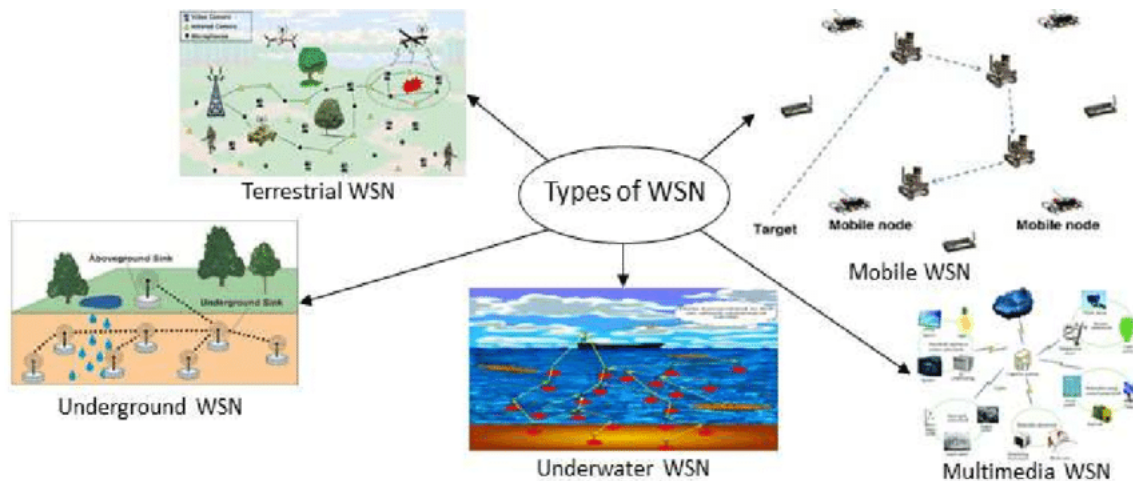
Mạng cảm biến không dây là một tập hợp các nút cảm biến tự động ghi lại dữ liệu của môi trường xung quanh, sau đó tổng hợp lại tại một trạm trung tâm để xử lý nhằm phục vụ cho mục đích giám sát, kiểm tra.[13]

Trong [13], nhóm tác giả phân loại mạng cảm biến theo địa hình được triển khai:

- Mạng cảm biến trên mặt đất: Mô hình mạng này bao gồm hàng trăm cho tới hàng ngàn nút cảm biến, triển khai rải rác trên mặt đất. Các vị trí đặt nút cảm biến có thể là trên chiến trường, trong rừng, trên núi. Có thể triển khai một cách tùy biến hoặc được lên kế hoạch từ trước, có thể sử dụng thiết bị bay để vận chuyển các nút cảm biến tới địa điểm mong muốn hoặc địa điểm con người khó tiếp cận.
- Mạng cảm biến trong lòng đất: Mô hình này bao gồm các nút cảm biến được chôn sâu dưới lòng đất hoặc trong hang động, trong lòng núi nhằm đo đạc số liệu để dự báo thảm họa như núi lửa, động đất hay quan sát tình hình đất đai để có kế hoạch canh tác hợp lý. Vì trở ngại môi trường khó thay mới mà các nút cảm biến được lựa chọn cho địa hình này sẽ rẻ hơn so với loại triển khai trên mặt đất.
- Mạng cảm biến dưới nước: Mô hình này bao gồm các nút cảm biến thả dưới nước, có thể di chuyển được bằng bánh xe hoặc lực đẩy, liên lạc với nhau bằng sóng âm. Một ví dụ thực tế của mô hình này là Hệ thống giám sát âm thanh được nhắc tới ở phần đầu.

- Mạng cảm biến truyền thông: Bao gồm các nút cảm biến thu dữ liệu truyền thông như camera, máy ghi âm. Bởi đặc tính của mạng truyền thông là kích thước gói tin lớn mà giao thức và đặc tính của các nút cảm biến cũng cần thay đổi để kết quả thu được không có độ trễ lớn.
- Mạng cảm biến di động: Các nút cảm biến trong mô hình mạng này có thể di chuyển để trao đổi thông tin hoặc sạc năng lượng. Khó khăn đặt ra trong mô hình này là tính toán độ bao phủ của mạng, giao thức truyền tin hợp lý, ít tiêu tốn năng lượng và chu trình sạc năng lượng của các nút cảm biến.

Hình 1.1 là hình ảnh trực quan về địa hình được triển khai và các nút cảm biến được cài đặt trong thực tế.



Hình 1.1: Các loại mạng cảm biến không dây

1.1.2 Nút cảm biến

Thành phần nhỏ nhất của một mạng cảm biến không dây là nút cảm biến. Cấu tạo của nút cảm biến bao gồm những bộ phận chính sau: bộ phận điều khiển, bộ phận chuyển tiếp, bộ nhớ, nguồn năng lượng, một hoặc nhiều bộ cảm ứng.

- Các thành phần cảm ứng là các thiết bị phần cứng, liên tục đo đạc những thông số của môi trường xung quanh, như là áp suất hay nhiệt độ, sau đó chuyển các tín hiệu liên tục này thành các tín hiệu số để xử lý tiếp bằng các thành phần khác. Có thể phân loại thành phần cảm ứng thành 3 loại: bị động - đa hướng, bị động - đơn hướng hoặc chủ

động. Các thiết bị bị động - đa hướng chỉ ghi nhận dữ liệu mà không tác động trực tiếp với môi trường. Dữ liệu ghi nhận bởi thiết bị này không chỉ ra hướng thu nhận được, khác với các thiết bị bị động - đơn hướng như camera. Các thiết bị chủ động thì tác động tới môi trường, ví dụ bằng tia rada, và chúng cần được liên tục cấp năng lượng.

- Bộ nhớ thì tùy thuộc vào yêu cầu mà được trang bị hay không. Vì mục tiêu tiết kiệm năng lượng mà bộ nhớ trên vi xử lý được ưu tiên hơn là bộ nhớ flash. Hai tác vụ chính của bộ nhớ bao gồm: lưu trữ dữ liệu và phục vụ cho tính toán.
- Bộ phận xử lý có thể coi như là bộ não của nút cảm biến, chịu trách nhiệm thực thi nhiệm vụ, xử lý dữ liệu và điều khiển các bộ phận khác. Lựa chọn phổ biến là sử dụng vi điều khiển bởi tính tiết kiệm, tiêu thụ ít năng lượng và dễ dàng lập trình.
- Bộ phận chuyển tiếp đóng vai trò cầu nối giữa các nút cảm biến, chịu trách nhiệm truyền và nhận các gói dữ liệu. Môi trường truyền tin sử dụng sóng radio, tia laze hoặc tia hồng ngoại. Tia laze thì sử dụng ít năng lượng, nhưng yêu cầu không gian truyền không được có vật cản. Tia hồng ngoại thì bị hạn chế bởi phạm vi giao tiếp. Bởi vậy phù hợp nhất với các ứng dụng của mạng cảm biến không dây là sử dụng sóng radio.

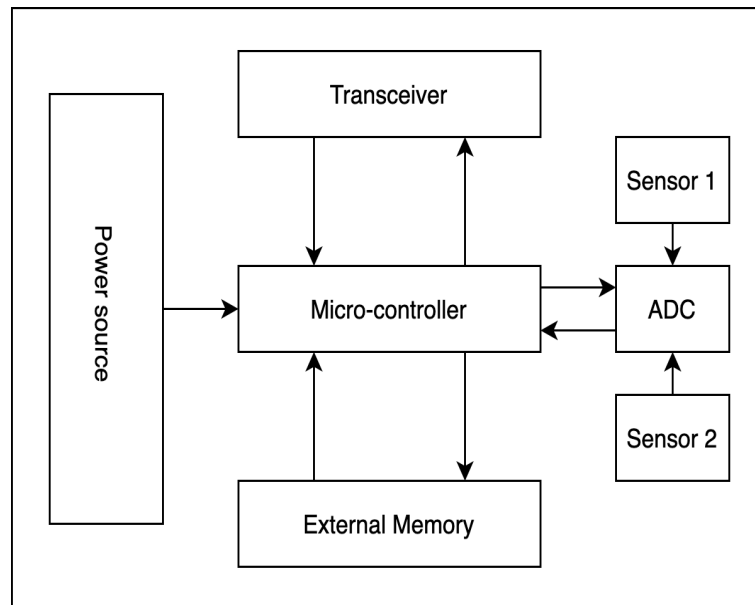
Vì thường được lắp đặt tại các vị trí khó tiếp cận, việc sạc hay thay pin dường như là không khả thi, bởi vậy mà nguồn năng lượng chính của các nút cảm biến thường sử dụng là pin. Các nút cảm biến tiêu tốn năng lượng cho các nhiệm vụ khác nhau, bao gồm: cảm ứng, xử lý dữ liệu và truyền dữ liệu. Trong đó việc truyền dữ liệu tiêu tốn năng lượng nhiều nhất.

Hình 1.2 là mô tả cấu trúc của một nút cảm biến trong mạng cảm biến không dây.

1.1.3 Các giao thức truyền tin của mạng cảm biến

Mạng cảm biến không dây có thể được triển khai bằng nhiều giao thức: [1]

- Dạng bus: Giao thức mạng đơn giản, được sử dụng khi có một số lượng ít nút cảm biến. Các nút thực hiện việc truyền tin qua hình thức quảng bá, tất cả các nút đều nhận được gói tin nhưng chỉ nút cảm biến đích là xử lý gói tin truyền tới. Tuy dễ dàng để cài đặt, giao thức dạng này cần phải có cơ chế xử lý xung đột hợp lý.

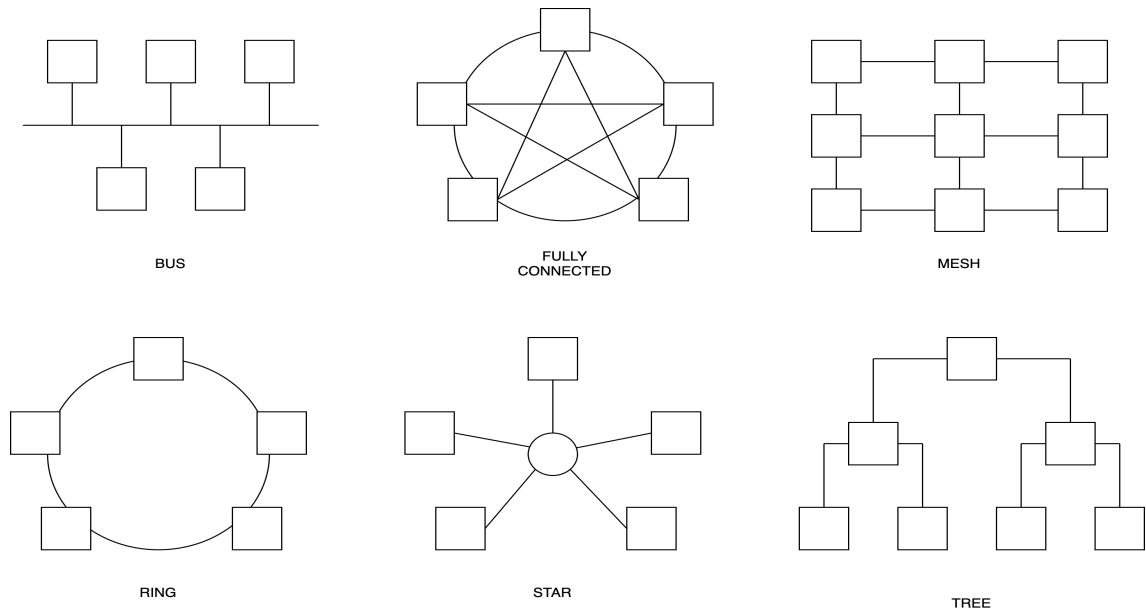


Hình 1.2: Cấu trúc một nút cảm biến

- Dạng cây: Giao thức dạng cây được xây dựng theo cơ chế kế thừa, các nút con (là các nút cảm biến) có trách nhiệm đo đạc thông tin môi trường và truyền ngược lại các nút cha (có thể là các nút cảm biến hoặc chỉ là các nút truyền tin) để xử lý. Vì việc truyền và nhận tin thường tiêu tốn nhiều năng lượng nhất (so với việc xử lý hay lưu trữ), bởi vậy mà cần tìm đường truyền ngắn nhất đi từ nút con tới nút gốc.
- Dạng hình sao: Giao thức này bao gồm một khu xử lý trung tâm nhận tất cả thông tin từ các nút xung quanh để xử lý. Tất cả việc giao tiếp đều phải thực hiện thông qua khu trung tâm này (giống với mô hình client - server).
- Dạng hình vòng: Các nút trong giao thức này được kết nối với chính xác 2 nút khác, việc truyền tin do đó được thực hiện tuần tự theo chiều kim đồng hồ hoặc ngược kim đồng hồ. Nếu có một nút bị chết thì toàn bộ mạng phải có cơ chế hồi phục lại.
- Dạng hỗn tạp (Mesh Network): Giao thức mạng cho phép một nút có thể giao tiếp với các nút kề với nó, đồng thời có thể truyền tin theo nhiều hướng tới nút đích (khác với mạng hình vòng chỉ cho truyền theo 2 hướng).
- Dạng kết nối đầy đủ (Fully connected network): Giao thức mạng mà

một nút kết nối với tất cả các nút khác trong mạng. Số lượng nút và số lượng kết nối giữa tất cả các nút tăng theo hàm lũy thừa khiến cho việc tính toán đường đi tối ưu cho gói tin rất khó khăn.

Hình 1.3 là tổng hợp của các giao thức thường được triển khai trong thực tế.



Hình 1.3: Các giao thức mạng trong thực tế

1.1.4 Ứng dụng của mạng cảm biến không dây

Mạng cảm biến không dây từ những ngày đầu chỉ được ứng dụng trong quân sự và khoa học nay đã phát triển rộng rãi và cho thấy tác dụng lớn khi áp dụng vào trong những mặt khác của đời sống như: nông nghiệp, y tế, môi trường, xây dựng. Việc nhận thấy tiềm năng của mạng cảm biến không dây trong nhiều lĩnh vực của xã hội thúc đẩy khoa học và doanh nghiệp đẩy mạnh nghiên cứu, hợp tác để cho ra đời những nút cảm biến hiệu quả hơn, tiết kiệm hơn và các cơ chế triển khai mạng cảm biến tối ưu hơn. Dưới đây là bản nghiên cứu tác động của mạng cảm biến không dây trong các lĩnh vực khác nhau:[20]

- **Nông nghiệp:** Các mạng cảm biến kết hợp của nhiều loại như dưới đất, trong lòng đất hay trong nước cung cấp thông tin cần thiết cho canh tác như nhiệt độ, độ ẩm, chất lượng đất, số lượng khí CO₂ trong lòng

đất, chất lượng nước. Dữ liệu được thu thập qua camera hoặc cảm biến, sau đó được xử lý và đưa tới chuyên gia để đưa ra quyết định sau cùng. Nhờ thông tin được truyền tải theo thời gian thực mà chuyên gia có thể phân tích tình trạng cây trồng, đất đai, nguồn nước để có cơ chế điều chỉnh thích hợp. Ngoài việc ghi nhận thông tin, các mạng cảm biến còn đóng vai trò tự động hóa quy trình canh tác, điều khiển để giảm thiểu công sức lao động, tăng năng suất và độ chính xác, giống như hệ thống thời gian thực được điều khiển bằng cảm biến tại các khu trồng trọt trong nhà kính.

- **Môi trường:** Nhờ việc ghi nhận dữ liệu và truyền tải qua Internet mà có thể dự đoán trước các thiên tai tự nhiên như động đất, sóng thần, lũ lụt hay tai nạn lao động như rò rỉ khí ga, rò điện, qua đó giảm thiểu thiệt hại về người và của. Các ứng dụng thực tiễn trong môi trường có thể kể đến giám sát mức độ ô nhiễm môi trường, giám sát chất lượng nước và khí thải trong nhà máy, dự đoán nguy hiểm trong hầm mỏ như khí độc, nguồn lửa ngầm hoặc sập hầm.
- **Giao thông:** Giám sát giao thông là một ví dụ thực tiễn của mạng cảm biến truyền thông, kết hợp nhiều camera giám sát để thu được hình ảnh thời gian thực, hỗ trợ việc điều tiết, phân luồng, truy cứu trách nhiệm khi có tai nạn, xử lý vi phạm giao thông. Hệ thống giám sát giao thông đã được triển khai ở rất nhiều thành phố lớn trên thế giới giúp giảm thiểu lực lượng công quyền cần có mặt trực tiếp trên đường.
- **Y tế:** Các mạng cảm biến không dây được triển khai để giám sát sức khỏe của bệnh nhân, cụ thể là đo đặc thông số cơ thể như áp suất, chất béo, đặc biệt là ở người già, để có thể liên tục theo dõi và chăm sóc tình trạng. Bệnh nhân có thể đeo các thiết bị đo đặc và hệ thống sẽ tự động gửi thông tin về trung tâm liên tục để bác sĩ có thể chẩn đoán và đưa ra phác đồ điều trị một cách chính xác nhất.
- **Quân sự:** Vai trò của mạng cảm biến không dây vẫn chưa hề mờ nhạt trong lĩnh vực quân sự, thậm chí là các mạng cảm biến hiện thời còn mang lại nhiều giá trị hơn. Giám sát kẻ địch, phát hiện xâm nhập, tính toán toán lộ trình là các mục tiêu chính khi triển khai mạng cảm biến không dây trên chiến trường.

1.1.5 Khó khăn

Như đã đề cập ở đầu chương, việc được ứng dụng rộng rãi ở nhiều lĩnh vực thúc đẩy những tiến bộ khoa học nhưng đồng thời cũng mang tới những

thách thức khi triển khai một mạng cảm biến không dây trong thực tế. [6]

Một trong những thách thức lớn nhất và được ưu tiên hàng đầu khi thiết kế mạng cảm biến không dây là duy trì thời gian sống. Bởi đặc tính khó tiếp cận của môi trường đặt các nút cảm biến (trong lòng đất, trên chiến trường) vậy nên việc sạc pin cho các mạng cảm biến hay đổi pin mới dường như là không thể. Hơn thế nữa với các giao thức mạng hình cây hoặc hình vòng, một nút chết trong mạng có thể dẫn tới việc tái định tuyến cho toàn bộ các nút khác. Từ hai nguyên do trên mà trước khi bắt đầu triển khai các nút cảm biến (có thể là chọn vị trí đặt hoặc xác định thuật toán định tuyến, giao thức truyền, giao thức xử lý xung đột), các kỹ sư đều phải lưu ý tới việc giảm thiểu năng lượng tiêu thụ. Nhưng bởi số lượng kênh hạn chế, lại cần phải được chia sẻ giữa rất nhiều nút mạng, điều này có thể dẫn tới độ trễ lớn nếu quá ưu tiên vào việc bảo vệ năng lượng, khiến cho thông tin trở nên vô giá trị. Cân bằng giữa việc duy trì thời gian sống và đảm bảo hiệu quả trong giao tiếp là một bài toán khó, với những yêu cầu thay đổi tùy vào tình hình thực tế.

Ở những môi trường có khả năng xảy ra hư hại cho các nút cảm biến (chiến trường), một cơ chế chịu lỗi hay tái định tuyến là cần thiết để đảm bảo giao tiếp trong toàn mạng khi có vấn đề xảy ra. Ví dụ, một nút cần cập nhật lại bảng địa chỉ các mạng khác khi nhận thấy việc truyền tin như trước không hiệu quả. Cơ chế chịu lỗi cùng với cơ chế tự vận hành, khi các nút trong mạng đã được thiết lập thì có thể tự động thực hiện tác vụ mà không cần tới sự có mặt của con người, hỗ trợ cho tính tự động hóa của mạng cảm biến không dây, giảm thiểu chi phí lao động.

Đôi khi, các mạng cảm biến không dây cần được thiết kế để đảm bảo độ phủ trong truyền tin hay thông tin cần được truyền trong thời gian thực. Độ bao quát của mạng cảm biến phục vụ cho mục đích giám sát tổng quát như các lưới bảo vệ tại các cứ điểm quan trọng để phát hiện quân địch hay các khu vực rộng lớn như đất ruộng canh tác để thu thập đầy đủ dữ liệu. Trong khi đó đảm bảo chất lượng dịch vụ là một trong những yêu cầu của mạng truyền thông bởi con người rất nhạy cảm với độ trễ của âm thanh hay hình ảnh. Việc đảm bảo chất lượng đường truyền, chất lượng gói tin, trong khi vẫn duy trì thời gian sống của mạng là một thách thức không nhỏ.

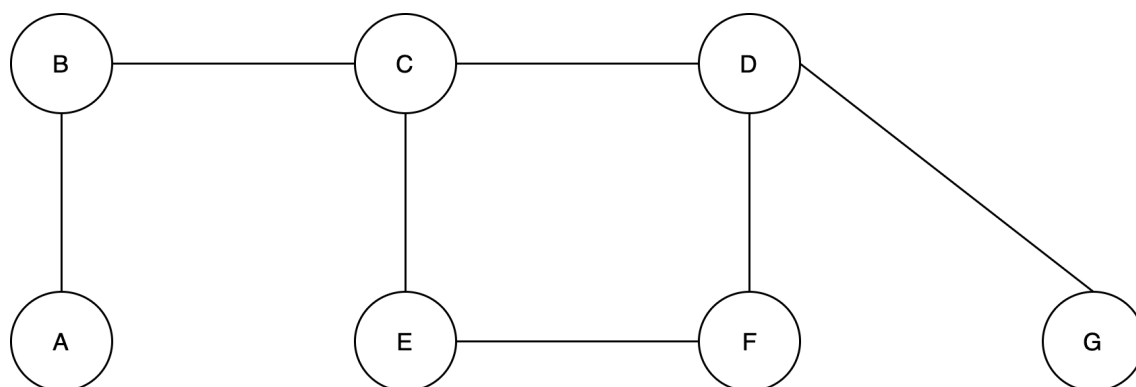
1.2 Lý thuyết đồ thị

1.2.1 Định nghĩa và biểu diễn đồ thị

Đồ thị, trong phạm vi toán học, là một tập hợp những đối tượng được mã hóa thành các đỉnh và mối liên hệ giữa các đối tượng được mã hóa dưới

dạng cạnh.

Hình 1.4 là một đồ thị đơn vô hướng.



Hình 1.4: Đồ thị đơn vô hướng

Đồ thị có thể được sử dụng để mô hình hóa rất nhiều bài toán trong thực tế: [21]

- Bản đồ: Một bản đồ giữa các thành phố có thể là một đồ thị, trong đó các thành phố là các đỉnh, đường đi từ thành phố A tới thành phố B gần đó là một cạnh của đồ thị. Việc mô hình hóa như này có thể giúp giải các bài toán như xác định đường đi ngắn nhất từ một thành phố tới thành phố khác bằng các giải thuật trên đồ thị.
- Internet: Các trang web trên mạng là các đỉnh của đồ thị và các hyperlink là các cạnh của đồ thị. Các thuật toán duyệt đồ thị đóng vai trò quan trọng trong các bộ công cụ tìm kiếm.
- Lập lịch: Cơ chế lập lịch xuất hiện từ những nơi bé nhất như hệ điều hành cho tới nhà máy, sân bay. Việc lập một kế hoạch hợp lý, đáp ứng yêu cầu, đảm bảo một số ràng buộc như tài nguyên hay độ ưu tiên có thể được xử lý trước hết bằng cách mô hình hóa bằng đồ thị có hướng, sau đó áp dụng các thuật toán sắp xếp hoặc tìm kiếm đường đi trên đồ thị.

Tùy thuộc vào yêu cầu bài toán mà có thể lựa chọn mô hình đồ thị phù hợp, vô hướng hoặc có hướng, cạnh có trọng số hoặc không. Với đồ thị vô hướng, chiều cạnh nối 2 đỉnh là không quan trọng, có thể duyệt từ A tới B hoặc từ B tới A đều được, nếu tồn tại cạnh giữa 2 đỉnh A và B. Trong khi đó với đồ thị có hướng, giả sử hướng từ A tới B thì chỉ có thể duyệt từ A tới B. Chúng ta thường mô hình hướng của cạnh trong đồ thị bằng cách thêm dấu mũi tên vào đích của cạnh, biểu thị chiều từ nguồn tới đích. Với trường hợp

trọng số của cạnh, với các bài toán mà việc di chuyển từ đỉnh này tới đỉnh khác tiêu tốn tài nguyên (xăng dầu khi di chuyển trên đường, thời gian hoàn thành tác vụ, v.v...), các cạnh nối giữa 2 đỉnh sẽ được đánh trọng số thể hiện số tài nguyên sẽ tiêu tốn khi di chuyển trên cạnh này.

Cùng tùy thuộc vào yêu cầu của bài toán mà cách lưu trữ đồ thị trong bộ nhớ cũng khác nhau, nhưng có thể lựa chọn cách lai ghép để thu được hiệu quả từ cả 2. Cách lưu trữ sử dụng ma trận nhị phân, mà phần tử a_{ij} là 0 hoặc 1 chỉ việc có tồn tại cạnh giữa 2 đỉnh i và j của đồ thị, sẽ tăng tốc việc truy cập nhưng lại tiêu tốn bộ nhớ. Trong khi đó nếu sử dụng các danh sách liên kết mà phần tử đầu là các đỉnh và các phần tử trong danh sách là các đỉnh liên kết với nó sẽ giúp giảm thiểu bộ nhớ sử dụng, nhưng tất nhiên việc truy xuất sẽ tiêu tốn thời gian hơn.

1.2.2 Thuật toán duyệt đồ thị

Duyệt đồ thị được hiểu là ghé thăm tất cả các đỉnh ở đồ thị (hoặc cho tới khi tìm được đỉnh cần thiết), mỗi đỉnh đúng một lần. Thuật toán duyệt đồ thị có thể được sử dụng như một cơ chế tiền xử lý cho các bài toán phức tạp khác như xác định chu trình hay tìm đường đi ngắn nhất.

Bên dưới là thuật toán Duyệt theo chiều rộng (Breadth-first search - BFS), là thuật toán sẽ thăm tất cả các đỉnh ở mức hiện tại (các nút kề với đỉnh đã xét gần nhất) trước khi thăm các đỉnh ở mức khác.

Algorithm 1 Breadth-first search

```
1: procedure BFS( $Graph, Key$ )
2:   Tạo hàng đợi rỗng
3:   Thêm nút gốc vào hàng đợi
4:   Tạo tập hợp các nút đã thăm
5:   while Hàng đợi vẫn chưa rỗng do
6:     nút  $\leftarrow$  đầu hàng đợi
7:     Kiểm tra khóa của nút
8:     Thêm nút vào tập hợp đã thăm
9:     Thêm vào hàng đợi tất cả các đỉnh lân cận chưa được thăm của
       nút được xét gần nhất
10:  end while
11: end procedure
```

Duyệt theo chiều sâu (Depth-first search - DFS) ngược lại so với BFS khi chúng ta ưu tiên thăm các đỉnh theo chiều sâu trước khi mở rộng ra các đỉnh lân cận.

Algorithm 2 Depth-first search

```
1: procedure DFS(Graph, Key)
2:   Tạo stack rỗng
3:   Thêm vào stack nút gốc
4:   Tạo tập hợp các nút đã thăm
5:   while Stack vẫn chưa rỗng do
6:     nút  $\leftarrow$  pop stack
7:     Kiểm tra khóa của nút
8:     Thêm vào stack tất cả các đỉnh lân cận chưa được thăm của đỉnh
       được xét gần nhất
9:   end while
10: end procedure
```

1.2.3 Thuật toán xây dựng cây khung nhỏ nhất

Cây khung nhỏ nhất là một đồ thị liên thông không có chu trình nối tất cả các đỉnh trong đồ thị. Hai thuật toán xây dựng cây khung nhỏ nhất được trình bày bên dưới là Kruskal và Prim.

Algorithm 3 Thuật toán Kruskal

```
1: procedure KRUSKAL(Graph)
2:   Tạo một rừng với mỗi đỉnh là một cây độc lập.
3:   while Cây khung nhỏ nhất chưa hình thành do
4:     Loại bỏ một cạnh từ tập E với trọng số nhỏ nhất.
5:     if Hai đỉnh trên cạnh thuộc về hai cây khác nhau then
6:       Kết hợp thành một cây.
7:     end if
8:   end while
9: end procedure
```

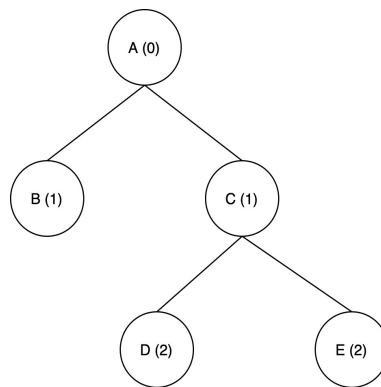
1.2.4 Lý thuyết về cây trong đồ thị

Cây, trong mối liên quan với đồ thị, có thể được xem như một đồ thị liên thông, không có chu trình, trong đó ta chọn ra một đỉnh gọi là gốc. Hình ảnh trực quan là dễ dàng nhất để giúp định nghĩa những thành phần trong cây.

Các thành phần của một cây khi quan sát từ hình 1.5 bên trên sẽ bao gồm: nút A là nút gốc, nút B, C là các nút con của và tương ứng nút A là cha của chúng (nút C là cha của 2 nút D,E và D,E là con của C). (Vì chúng

Algorithm Thuật toán Prim

```
1: procedure PRIM(Graph)
2:   Gán trọng số cho mỗi đỉnh.
3:   while Cây khung nhỏ nhất chưa hình thành do
4:     Lựa chọn đỉnh  $v$  có trọng số nhỏ nhất thêm vào cây.
5:     Cập nhật trọng số cho các đỉnh liền kề với  $v$  nếu chi phí di chuyển
       từ  $v$  nhỏ hơn trọng số hiện tại.
6:   end while
7: end procedure
```



Hình 1.5: Cây trong lý thuyết đồ thị.

ta chỉ đề cập tới cây tổng quát nên sẽ không có khái niệm con trái hay con phải). Các nút D,E được gọi là các nút lá nếu không có bất kỳ con nào. Một cây có nhiều mức, các cây cùng một cha sẽ cùng một mức (hoặc có cha cùng mức thì cũng có cùng mức) và sẽ có mức lớn hơn mức cha là 1. Bắt đầu từ nút gốc có mức 0, các nút B,C sẽ có mức 1, D,E có mức 2. Chiều cao của cây là đường dài nhất đi từ gốc tới nút lá bất kỳ, hay được tính đơn giản bằng mức sâu nhất.

Cây Steiner

Trong phạm vi tổng quát, cây Steiner là một tập các bài toán trong lĩnh vực tối ưu hóa tổ hợp. Nhưng để tập trung vào vấn đề tiếp theo sẽ giải quyết ở chương 2, chúng ta chỉ quan tâm tới cây Steiner trong đồ thị, có thể được diễn giải một cách đơn giản như sau: một cây Steiner là một cây có trọng số nhỏ nhất chứa tất cả những đỉnh cần thiết (là một tập con của tất cả các đỉnh trong đồ thị), cây có thể bao gồm các đỉnh khác hoặc không. Khái niệm trọng số nhỏ nhất ở đây có thể hiểu là tối thiểu hóa hàm mục tiêu (các mục tiêu có thể là thời gian sống trong mạng, độ bao phủ của các nút cảm

biến, độ trễ của đường truyền) đối với các bài toán trong mạng cảm biến không dây. Khi số lượng nút Steiner (các nút bắt buộc xuất hiện trong cây) bằng với số nút trong đồ thị, ta có thể quy về bài toán cây khung nhỏ nhất. Còn khi chỉ có 2 nút Steiner thì ta có thể quy về bài toán đường đi ngắn nhất. Mặc dù vậy, tuy 2 bài toán trên có thể giải trong thời gian đa thức, cây Steiner lại thuộc lớp NP-complete.[12]

1.3 Tổng kết

Chương 1 bao gồm những kiến thức tổng quan về lĩnh vực mà đề án nghiên cứu (mạng cảm biến không dây), và một số phạm trù khoa học máy tính liên quan tới bài toán đặt ra (đồ thị và cây Steiner). Cụ thể, phần 1.1 cho biết những giao thức mạng thường được sử dụng, ứng dụng và khó khăn khi triển khai một mạng cảm biến không dây trong thực tế. Kiến thức trong phần này có thể giúp những người mới tiếp cận dễ dàng hơn với một trong những lĩnh vực đang thu hút nhiều sự quan tâm của giới nghiên cứu cũng như có cái nhìn bao quát về sự hiện diện của mạng cảm biến không dây trong đời sống. Phần 1.2 đóng vai trò như sự bổ sung kiến thức cho phần diễn giải chi tiết ở chương sau, đặc biệt là phần giải mã của các thuật toán sẽ được áp dụng đã được trình bày chi tiết, cụ thể là thuật toán BFS để duyệt cây và Kruskal để xây dựng cây con nhỏ nhất.

Chương 2

Tối ưu đa mục tiêu trên cây tổng hợp dữ liệu

Sự tiến bộ của công nghệ giúp sản xuất các thiết bị cảm biến ngày một hiện đại, chiếm ít không gian, hiệu năng cao, giá thành rẻ, hỗ trợ triển khai số lượng lớn trên cùng một không gian. Đi liền với đó là khó khăn trong việc tận dụng hiệu quả các nút cảm biến khi năng lượng, khả năng tính toán, bộ nhớ vẫn còn hạn chế. Tổng hợp dữ liệu được nghiên cứu nhằm giải quyết khó khăn này. Nhiệm vụ này được định nghĩa như một quy trình toàn cục bao gồm việc tập trung và điều hướng thông tin thông qua các lớp mạng, xử lý dữ liệu tại các nút trung gian nhằm giảm thiểu việc tiêu thụ năng lượng, qua đó duy trì thời gian sống toàn mạng [8].

Các tác vụ trong tổng hợp dữ liệu có thể kể đến: giao tiếp thông tin giữa các nút cảm biến, tính toán, tổng hợp tại một nút và lưu trữ dữ liệu. Trong thực tế, việc giao tiếp thông tin giữa các nút cảm biến thường tiêu hao nhiều năng lượng nhất. Tác vụ này bao gồm rất nhiều phương pháp điều hướng gói tin trong mạng, hỗ trợ cho việc tổng hợp dữ liệu từ nhiều nguồn và truyền tải tới một (hoặc nhiều) đích đến. Một số giao thức hỗ trợ giao tiếp thông tin trong mạng có thể kể đến: giao thức dạng cây, giao thức dạng cụm, giao thức đa đường và giao thức lai.

Giao thức dạng cây là phương thức cổ điển để tổng hợp dữ liệu trong mạng cảm biến không dây, trong đó các nút cảm biến được tổ chức theo cấu trúc phân cấp và kế thừa. Trước hết, một cây với trạm tiếp nhận thông tin là nút gốc được xây dựng. Sau đó việc truyền và nhận thông tin được thực hiện tuần tự theo từng cấp, từ lá tới gốc. Giao thức này phù hợp để áp dụng trong các bài toán tối ưu hóa các hàm tổng hợp dữ liệu và quản lý năng lượng [8].

Trong thực tế, không phải lúc nào tất cả các nút được triển khai cũng truyền tải dữ liệu, chỉ một số các nút được chỉ định là các nút cảm biến có

nhiệm vụ thu thập dữ liệu xung quanh và truyền lên các cấp cao hơn. Bài toán lúc này trở thành bài toán cây Steiner: tìm cây khung nhỏ nhất chứa tập con từ tập các đỉnh cho trước.

2.1 Mô hình năng lượng

Mô hình mất mát năng lượng ở [9] được áp dụng. Công thức tính toán năng lượng nhận và truyền dữ liệu có dạng như sau:

$$E_{RX} = k \times E_{elec}. \quad (2.1)$$

$$E_{TX} = \begin{cases} k \times E_{elec} + k \times \epsilon_{fs} \times d^2, & d < d_0 \\ k \times E_{elec} + k \times \epsilon_{mp} \times d^4, & d \geq d_0. \end{cases} \quad (2.2)$$

d là khoảng cách thực tế giữa các nút. ϵ_{fs} và ϵ_{mp} là các hằng số để tính toán năng lượng tiêu thụ khi truyền gói tin trong khoảng cách ngắn và khoảng cách dài.

Năng lượng tổng hợp dữ liệu E_{DA} tại các nút là một hằng số.

2.2 Phát biểu bài toán

- Đầu vào: Mạng cảm biến không dây được triển khai trên địa hình ba chiều, mô phỏng theo chuẩn Digital Elevation Model [19]. Trong đó, trạm trung tâm được đặt ở chính giữa của địa hình, các nút cảm biến được phân bố theo các phân phối và có vị trí cố định.
 - Tập các nút bao gồm $\{s_1, s_2, \dots, s_n, r_1, r_2, \dots, r_m, BS\}$. s_i và r_j là nút cảm biến thứ i và nút trung gian thứ j , BS là trạm trung tâm. Tọa độ của các nút được kí hiệu là (x, y, z) tương ứng với giá trị theo độ rộng, độ dài và độ cao trong không gian.
 - Bán kính giao tiếp các nút là r_c .

Một nút u có thể giao tiếp với một nút v nếu nút u nằm trong bán kính truyền thông của nút v và ngược lại. Bán kính truyền thông giữa hai nút là tổng của bán kính giao tiếp của từng nút, trong bài toán hiện tại là $2 * r_c$. Do đó, mạng cảm biến không dây có thể được mô hình dưới dạng đồ thị $G = (V, E)$ trong đó V là tập các nút (bao gồm cả nút cảm biến và nút trung gian) và E là tập các cặp nút (u, v) có khả năng giao tiếp với nhau. Tập các nút Steiner được kí hiệu là RN ($|RN| \leq m$).

- Đầu ra: Cây khung $T = (V_{n+|RN|}, E_{n+|RN|})$ với n là số lượng các nút cảm biến.
- Các mục tiêu:
 - Tổng năng lượng tiêu thụ Vấn đề lớn nhất đối với các mạng cảm biến không dây là việc suy hao năng lượng của các nút cảm biến. Bởi các nút cảm biến dự trữ rất ít năng lượng, việc áp dụng các thuật toán nhằm tối thiểu hóa năng lượng tiêu thụ trong quá trình vận hành là vô cùng cần thiết. Năng lượng tiêu thụ của cây tổng hợp dữ liệu có thể được tính bằng tổng năng lượng trên tất cả các nút.

$$f_1(x) = \sum_{i=1}^{n+|RN|} (E_{TX}^i + \sum_{j=1}^{C_{num_i}} (E_{RX}^j + E_{DA}^j)) \quad (2.3)$$

C_{num_i} là tất các nút con của nút thứ i trong cây tổng hợp dữ liệu. Năng lượng tiêu thụ trên mỗi nút sẽ bằng tổng của năng lượng truyền gói tin cho nút cha của nó và năng lượng tổng hợp (bao gồm năng lượng nhận gói tin và năng lượng tổng hợp dữ liệu) từ tất cả các nút con.

- Thời gian sống của mạng Một trong những mục tiêu khi triển khai mạng cảm biến không dây là duy trì thời gian sống toàn mạng. Bởi đặc điểm gồ ghề, khó tiếp cận của một số địa hình khi triển khai gây khó khăn cho việc thay thế, khắc phục sự cố như hết pin, tai nạn hay phá hoại từ bên ngoài mà việc duy trì tính ổn định là vô cùng cần thiết. Tùy thuộc vào yêu cầu thực tế mà định nghĩa thời gian sống của mạng cảm biến thay đổi: thời gian mà mạng cảm biến vẫn còn duy trì việc giao tiếp như thông thường, thời gian mà mạng cảm biến vẫn còn bao quát một không gian, thời gian từ khi bắt đầu tới khi việc thu thập dữ liệu lần đầu thất bại. Khi một nút cảm biến tiêu thụ hết năng lượng thì cây tổng hợp dữ liệu phải tự điều chỉnh lại. Thời gian sống của mạng, do đó, có thể tính toán qua thời gian sống tối thiểu của một nút trên cây.

$$f_2(x) = \frac{1}{\min_{i \in V} L^i} \quad (2.4)$$

với

$$L^i = \frac{E_{residual}^i}{E_{TX}^i + \sum_{j=1}^{C_{num_i}} (E_{RX}^j + E_{DA}^j)} \quad (2.5)$$

$E_{residual}^i$ là năng lượng còn lại của nút thứ i .

- Độ trễ khi tổng hợp dữ liệu Nhược điểm khi sử dụng mô hình tổng hợp dữ liệu dạng cây là thời gian tổng hợp dữ liệu lớn, gây chậm trễ trong quá trình xử lý thông tin. Theo mô hình được đề xuất, các nút trung gian không chỉ có vai trò chuyển phát dữ liệu lên nút cha mà còn phải tổng hợp dữ liệu từ các nút con. Một nút cũng chỉ có thể bắt đầu chuyển dữ liệu khi đã tổng hợp dữ liệu từ tất cả các nút con. Một thực tế rõ ràng là sự phân phối không đồng đều giữa các nút con sẽ dẫn tới tình trạng chờ đợi không cần thiết ở các nút còn lại. Các cách giải quyết nhược điểm này sẽ được bàn kĩ hơn ở phần bên dưới.

Từ định nghĩa bên trên, công thức tính độ trễ cho toàn mạng sẽ tương ứng với tổng độ trễ trên từng lớp, và độ trễ trên mỗi lớp sẽ là độ trễ lớn nhất của một nút ở lớp đó.

$$f_3(x) = \sum_{i=1}^h lt(i-1) = \sum_{i=1}^h CN_{i-1} * \frac{Max(data_i)}{rate_i} \quad (2.6)$$

CN_{i-1} là số lượng nút con lớn nhất ứng với một nút ở lớp thứ $i-1$, $Max(data_i)$ và $rate_i$ là kích thước lớn nhất mà dữ liệu có thể truyền và tốc độ truyền ở lớp thứ i .

- Độ nhiễu trong giao tiếp Các nút trong mạng cảm biến sử dụng kết nối không dây để giao tiếp, hệ quả trực tiếp là phát sinh vấn đề nhiễu trong truyền và nhận dữ liệu. Vấn đề này xuất hiện khi trong bán kính giao tiếp của hai nút đang kết nối với nhau xuất hiện tín hiệu từ một nút thứ ba, gây xung đột trong quá trình truyền tin. Xung đột xảy ra bắt các nút cảm biến phải truyền lại gói tin, gây tổn hao năng lượng và ảnh hưởng tới chất lượng tổng hợp. Giảm thiểu số lượng nút ảnh hưởng trong phạm vi kết nối giữa hai nút, do đó là một trong những mục tiêu cần tối ưu khi thiết kế mạng cảm biến không dây. Công thức được suy ra trực tiếp từ định nghĩa này:

$$f_4(x) = Max|Cov_e| \quad (2.7)$$

với

$$Cov_e = \{n \in V | n \text{ là số lượng nút nằm trong } D(x, |x, y|) \text{ hoặc } D(y, |y, x|)\}$$

- Ràng buộc: Cây khung T cần thoả mãn các ràng buộc:

- Các cạnh của cây có độ lớn không vượt quá $2 * R_c$
- Các nút lá đều là nút cảm biến.
- Tất cả các nút cảm biến đều xuất hiện trong cây.

2.3 Nghiên cứu liên quan

- Tổng năng lượng tiêu thụ: Trong cây tổng hợp dữ liệu, các nút trung gian đóng vai trò là nơi tổng hợp dữ liệu nhằm hạn chế hao tổn nếu phải truyền lại gói tin. Tuy đi kèm với đó là thời gian để nhận gói tin từ tất cả các nút con, và mất mát năng lượng để tổng hợp, nhưng thời gian và năng lượng hao phí này là nhỏ nếu so với việc phải truyền lại gói tin khi không tổng hợp trước. Quan sát này cho thấy cần phải cân nhắc lựa chọn nút nào là nút cha.

Một quan sát quan trọng khác là khoảng cách ảnh hưởng trực tiếp tới năng lượng cần cấp phát để truyền gói tin. Nói cách khác, các nút ở gần (nếu còn đủ năng lượng) nên được ưu tiên làm nút cha để tối thiểu hóa năng lượng tiêu thụ. Quan sát này là tiền đề cho thuật toán đề xuất trong [7]. Trong đây, tác giả đề cập tới việc sử dụng ba tiêu chí với độ ưu tiên giảm dần để lựa chọn nút cha cho các nút như sau: năng lượng còn lại, khoảng cách truyền và số lượng nút con tối đa. Hai tham số đầu tiên đồng thời là tiêu chí lựa chọn nút cha của các thuật toán đề xuất trong [14] và [15]. Tham số thứ ba được đề xuất để tránh tình trạng một nút có quá nhiều nút con dẫn tới mất mát năng lượng quá nhanh, hỗ trợ cho việc duy trì thời gian sống toàn mạng.

- Thời gian sống của mạng: Một trong những lý do chính dẫn tới việc tiêu hao năng lượng quá nhanh (dẫn tới việc nút "chết" trong mạng cảm biến) là do sự mất cân bằng trong việc tổng hợp và giao tiếp thông tin giữa các nút (một nút phải tổng hợp từ quá nhiều nút hoặc truyền dữ liệu tới một nút rất xa). Thuật toán tham lam trong [16] đề xuất một giải pháp đơn giản cho việc mất cân bằng giữa các nút trong mạng. Giải pháp trong bài báo trước hết là xây dựng một cây khung nhỏ nhất để đảm bảo khoảng cách giữa các nút không quá lớn, sau đó cân nhắc lựa chọn trong lân cận một nút còn lại nhiều năng lượng nhất là nút cha. Để điều chỉnh cho phù hợp với mô hình ban đầu, chúng ta sẽ loại bỏ các nút lá không phải là các nút cảm biến để tránh tình trạng dư thừa không cần thiết.

Trong số các thuật toán điều hướng theo mô hình phân cấp, mô hình Phân cấp các cụm thích nghi với năng lượng thấp (Low Energy Adap-

tive Cluster Hierarchy - LEACH) là một mô hình vô cùng nổi tiếng [10]. Trong đó, các nút cảm biến sẽ được phân làm các cụm phụ thuộc vào mức độ mạnh/yếu của tín hiệu nhận và lựa chọn một nút làm nút điều hướng của cụm đó. Việc này sẽ giúp tiết kiệm năng lượng một cách đáng kể bởi việc truyền thông tin chỉ được thực hiện bởi nút điều hướng chứ không phải tất cả các nút cảm biến. Việc lựa chọn nút nào làm nút đầu cũng sẽ được thay đổi liên tục để phù hợp với mục tiêu kéo dài thời gian sống toàn mạng.

Có một số giải thuật cải tiến từ giải pháp trên, hầu hết là cách lựa chọn nút điều hướng sao cho phù hợp để duy trì sự cân bằng trong tiêu thụ năng lượng, qua đó kéo dài thời gian sống toàn mạng. [3] đề xuất việc lựa chọn nút điều hướng và hình thành các cụm bằng thuật toán K-medoids, trong đó nút điều hướng là nút gần tâm cụm nhất tính toán theo khoảng cách Euclide. Trong khi đó [2] lại đề xuất việc sử dụng nút còn nhiều năng lượng nhất để làm nút điều hướng, và việc lựa chọn sẽ được thực hiện theo mỗi một chu kỳ truyền thông tin.

- Độ trễ khi tổng hợp dữ liệu: Trong [25], tác giả đã đề cập tới một phương pháp mang tính lý thuyết, sử dụng các nút cảm biến di chuyển được đóng vai trò như một cổng giao tiếp (gateway) để tổng hợp dữ liệu và chuyển tới các nút trung gian. Một cách dễ hình dung, các nút cảm biến sẽ được nhóm vào thành các cụm, và các cụm sẽ liên lạc với nhau thông qua các cổng giao tiếp. Số lượng cụm càng nhiều thì khả năng tổng hợp dữ liệu càng tốt (do số lượng nút ứng với từng cụm sẽ ít đi), tuy nhiên thời gian di chuyển từ nút lá tới trạm trung tâm lại tăng lên do gói tin cần đi qua nhiều trung gian. Giải thuật di truyền tối ưu hóa bước nhảy (GAHO - Genetic Algorithm for Hop count Optimization) được đề xuất trong [25] được đề xuất nhằm giảm thiểu khoảng cách truyền này.

Cũng sử dụng cách phân chia địa hình ra từng khu vực nhỏ, đề xuất trong [23] chia địa hình thành từng ô vuông với kích thước bằng nhau. Thuật toán phân cụm và chọn các nút điều hướng trong mỗi cụm chia thành các bước, bước sau sẽ là tổng hợp nhiều ô nhỏ từ bước trước thành một ô lớn hơn, cho tới khi toàn bộ mạng được liên kết với nhau thành một cây tổng hợp dữ liệu. Nhờ việc phân chia địa hình thành các ô vuông, khoảng cách từ các nút trong một ô tới nút điều hướng của cụm luôn được chặn trên (kích thước đường chéo của ô vuông đó), thuật toán tính được độ trễ tối đa dựa vào khoảng cách lớn nhất mà các nút trong mạng có thể truyền được.

Một phương pháp khác để phân chia cụm được đề xuất trong [24].

Thuật toán chia làm hai giai đoạn, giai đoạn thứ nhất là phân chia ra thành các nút mạnh (dominators) và các nút yếu (dominatees), sau đó sử dụng các nút mạnh để truyền dữ liệu tới trạm xử lý trung tâm, các nút yếu thì kết nối với các nút mạnh. Độ mạnh của từng nút được tính bằng khoảng cách từ nút đó tới trạm xử lý, nút càng gần thì càng mạnh, và cách tìm ra các nút mạnh sử dụng thuật toán duyệt theo chiều rộng (BFS - Breath First Search).

- Độ nhiễu trong giao tiếp: Nghiên cứu trong [17] đề xuất một giải pháp xây dựng cây tổng hợp dữ liệu giúp tối thiểu hóa độ nhiễu trung bình, qua đó tìm cận trên cho bài toán tối thiểu hóa độ nhiễu tối đa. Sử dụng cách tiếp cận với các nút phân bố trên địa hình 1 chiều (các nút xếp thẳng hàng trên một đường thẳng), sau đó giải quyết trên địa hình 2 chiều bằng cách kẻ các đường thẳng đi qua các nút cảm biến, và xây dựng kết nối giữa các nút sao cho độ nhiễu không vượt quá số lượng kết nối tối đa của một nút.

Cũng sử dụng các công thức của hình học không gian, nghiên cứu trong [11] phân chia địa hình ra thành từng cụm và lựa chọn đường đi tối ưu trong mỗi cụm. Có ba bước chính được sử dụng. Thứ nhất là mô hình Voronoi (Voronoi diagram) để tìm kiếm lân cận của các nút. Thứ hai là thuật toán tam giác hóa Delaunay (Delauney triangulation) để giảm thiểu độ dài kết nối giữa các nút (từ đó giúp giảm thiểu độ nhiễu trong quá trình truyền tin). Cuối cùng là sử dụng độ lệch chuẩn trong bán kính truyền như một tiêu chí để cắt giảm các kết nối có khoảng cách lớn. Cụ thể thì các cảm biến sẽ được đánh giá mức độ qua các kết nối của chúng, và tổng hợp mức độ từ hai đầu của một kết nối sẽ cho ra mức độ ưu tiên của kết nối đó. Các kết nối có mức độ ưu tiên cao sẽ bị loại bỏ.

Nhóm tác giả trong [5] đề xuất giải thuật Xây dựng rừng với độ nhiễu thấp (Low Interference Forest Establisher). Mạng cảm biến không dây được mô hình thành một tập các cạnh sắp xếp theo thứ tự tăng dần của khoảng cách. Các cạnh được thêm vào khi hai nút cảm biến ứng với cạnh đó chưa được kết nối trong mạng. Làm liên tục như vậy cho đến khi xây dựng được một cây hoàn chỉnh.

Nhóm tác giả trong [18] đề xuất sử dụng giải thuật JPSO để giải quyết bài toán. JPSO là một biến thể của Giải thuật tối ưu bầy đàn (Particle Optimization Problem - PSO) trong miền không gian rời rạc. Điểm nổi bật của giải thuật đề xuất là sử dụng mã hóa hai lớp lên mỗi cá thể. Lớp thứ nhất là một dãy nhị phân sắp xếp theo thứ tự tăng dần của các nút cảm biến, với giá trị 1 biểu thị nút đó được sử dụng và 0 là ngược lại. Lớp thứ hai là một

chuỗi tuần tự số thứ tự của các nút được sử dụng, được giải mã thành cây khung dựa vào Quy trình giải mã dịch chuyển khung (Edge window decoder - EWD). Quy trình của thuật toán là sử dụng một cửa sổ có độ lớn là 2, dịch từng bước một từ trái sang phải. Ở mỗi bước dịch xác định một cạnh tạo bởi 2 nút trong mạng cảm biến. Cạnh này sẽ được thêm vào cây khung nếu không xuất hiện chu trình. Làm tuần tự tới khi kết thúc toàn bộ lớp thứ 2 sẽ thu được một cây khung thỏa mãn.

Chương 3

Giải thuật di truyền và giải thuật di truyền sắp xếp không trội cho bài toán đa mục tiêu trên cây tổng hợp dữ liệu

Giải thuật di truyền được trình bày trong mã giả 5.

3.1 Giải thuật di truyền

Algorithm 5 Giải thuật di truyền

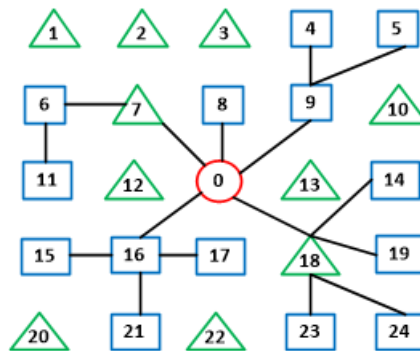
Require: Đồ thị không hoàn chỉnh vô hướng $G=(V,E)$ với nút gốc $v_0 \in V$

Ensure: Cây Steiner T của G

- 1: Sinh n cá thể của quần thể P_0
 - 2: **for** cá thể $i \in P_0$ **do**
 - 3:
 - Thực hiện giải thuật Kruskal trên tập các cạnh đã xáo trộn để tạo thành một cây;
 - Thực hiện quy trình sửa chữa để thu được cây Steiner;
 - Tính toán hàm phạt của i ;
 - 4: **end for**
 - 5: $t \leftarrow 0$
 - 6: **while** điều kiện dừng chưa thỏa mãn **do**
 - $Q_t \leftarrow$ Thực hiện lai ghép và đột biến trên P_t ;
 - $R_t \leftarrow Q_t \cup P_t$;
 - $P_{t+1} \leftarrow$ Chọn n cá thể tốt nhất từ R_t ;
 - $t \leftarrow t + 1$;
 - 7: **end while**
-

3.1.1 Biểu diễn cá thể

Giải thuật sử dụng tập cạnh để biểu diễn một cá thể.



Hình 3.1: Mã hóa một cá thể

Hình 3.1 mô tả mã hóa của một cá thể. Tập cạnh biểu diễn là: $\{(0,7),$

$(0,8), (0,9), (0,16), (7,6), (6,11), (9,4), (9,5), (16,15), (16,17), (16,21), (18,14), (18,19), (18,23), (18,24)\}$.

Việc mã hóa này đảm bảo một số tính chất quan trọng. Thứ nhất là tính bao quát, tất cả các lời giải khả dĩ đều có thể mã hóa được. Thứ hai là tính duy nhất, mỗi mã hóa chỉ có thể giải mã thành một lời giải duy nhất và ngược lại. Hơn thế, mã hóa sử dụng tập cạnh hỗ trợ thực hiện các nhiệm vụ khác như lai ghép, đột biến mà không cần chuyển đổi, do đó rút ngắn thời gian tính toán so với giải thuật đề xuất trong [18].

Để xây dựng một cây từ tập E ban đầu, trước hết ta áp dụng Kruskal, nhưng thay vì với tập các cạnh được sắp xếp theo độ lớn tăng dần, ta sẽ sử dụng tập cạnh được xáo trộn để tăng tính đa dạng của toàn quần thể. Giải thuật Kruskal có thể dừng lại nếu tất cả các nút cảm biến đều đã được duyệt qua, và áp dụng một quy trình cắt tỉa các nốt thừa (các nốt lá không phải nút Steiner) để thu được một lời giải hợp lệ.

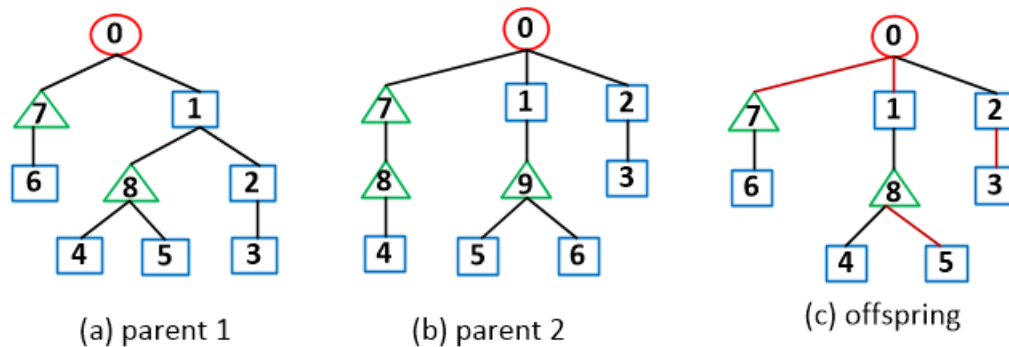
3.1.2 Khởi tạo quần thể

Trong các giải thuật tiến hóa, khởi tạo quần thể là một bước vô cùng quan trọng. Có hai chiến lược thường được sử dụng để khởi tạo quần thể: tham lam hoặc ngẫu nhiên. Chiến lược tham lam được áp dụng với một trong tập các mục tiêu đang xem xét, với mục đích tăng tốc độ hội tụ và tìm được lời giải gần tối ưu theo mục tiêu có trọng số cao nhất. Chiến lược này tạo nên sự mất cân bằng và tính đa dạng cho toàn bộ quần thể khi các cá thể có xu hướng phân cực về mục tiêu có trọng số cao nhất. Chiến lược còn lại, ngẫu nhiên, được sử dụng khi muốn gia tăng phạm vi tìm kiếm của quần thể, đánh đổi lại là thời gian hội tụ sẽ lâu hơn. Do các mục tiêu trong bài đều được đánh giá ngang nhau về mức độ quan trọng, giải thuật KGASL sử dụng chiến lược ngẫu nhiên trong toàn bộ quần thể. Để tạo sự ngẫu nhiên, từ tập cạnh lấy ra ban đầu của đồ thị, một số lượng ngẫu nhiên các cặp cạnh sẽ được lựa chọn để hoán đổi vị trí cho nhau, sau đó dùng tập cạnh mới này là đầu vào cho quy trình khởi tạo một cá thể đã trình bày ở trên.

3.1.3 Toán tử lai ghép

Lai ghép trong giải thuật tiến hóa được sử dụng nhằm trao đổi thông tin giữa các cá thể trong một quần thể, tổng hợp các tính trạng trội để tạo ra các cá thể ở thế hệ sau tốt hơn thế hệ trước. Đầu tiên, tập các cạnh chung giữa hai cá thể được lựa chọn để thêm vào cá thể con. Tập cạnh này được coi như các "tính chất di truyền mạnh" cần được di truyền lại cho thế hệ sau. Để hoàn thiện cá thể con, tập thứ hai bao gồm các cạnh chỉ xuất hiện trong cá thể cha hoặc mẹ được xây dựng, sắp xếp theo thứ tự tăng dần của

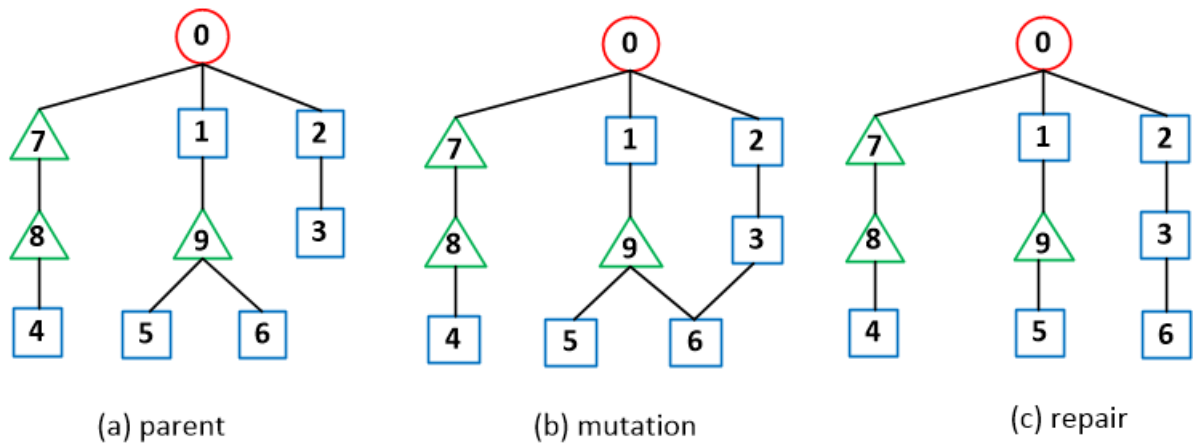
độ lớn và đưa vào cá thể con cho tới khi thu được một cây Steiner. Hình 3.2 mô tả quy trình lai ghép được áp dụng đối với một cặp cá thể cha và mẹ.



Hình 3.2: Áp dụng toán tử lai ghép lên hai cá thể cha mẹ. Các cạnh xuất hiện trong cả cha và mẹ được tô đỏ ở cá thể con.

3.1.4 Toán tử đột biến

Đột biến được sử dụng như một cách giúp giải thuật thoát khỏi tối ưu hóa địa phương. Đây là vấn đề của quy trình khởi tạo và lai ghép. Khởi tạo ngẫu nhiên, tuy mục đích là bao phủ phạm vi tìm kiếm càng rộng càng tốt, nhưng do bản chất ngẫu nhiên mà không thực sự kiểm soát được độ bao phủ. Kết hợp với toán tử lai ghép làm toàn bộ quần thể có hàm mục tiêu hội tụ nhanh về một điểm cực tiểu/cực đại địa phương. Đột biến hỗ trợ tìm kiếm ở những phạm vi mới, không nằm trong lân cận của vùng đang tìm kiếm. Tuy nhiên cần lựa chọn tỉ lệ đột biến một cách hợp lý, vì quy trình tìm kiếm cần càng rộng càng tốt ở những thế hệ đầu, nhưng càng nhanh càng tốt ở những thế hệ sau. Tỉ lệ đột biến sẽ tỉ lệ nghịch với số thứ tự của thế hệ, thế hệ càng lớn thì càng ít cá thể được đột biến. Toán tử đột biến trong giải thuật đề xuất bắt đầu với việc lựa chọn hai nút u và v . Giả sử, không mất tính tổng quát, nút có độ sâu lớn hơn là v . Ta xóa cạnh nối giữa v và nút cha của nó, sau đó nối u với v . Bây giờ u sẽ là cha của v trong cá thể mới. Quy trình này mang tới hai ưu điểm. Thứ nhất là sự toàn vẹn tính hợp lệ của cá thể khi chu trình chắc chắn sẽ không tồn tại. Thứ hai là các nút còn lại được giữ nguyên tính chất, không có gì cần thay đổi.



Hình 3.3: Ví dụ về toán tử đột biến

3.2 Giải thuật di truyền sắp xếp không trội

3.2.1 Cơ sở lý thuyết

Một số khái niệm được sử dụng trong giải thuật di truyền sắp xếp không trội nói riêng và các bài toán tối ưu đa mục tiêu nói chung.

- Lỗi giải trội: Với bài toán tối ưu đa mục tiêu, lời giải A được gọi là trội hơn lời giải B (kí hiệu là $A \prec B$) nếu tất cả các mục tiêu trong A có kết quả tốt hơn mục tiêu trong B.
- Lời giải không trội: Một lời giải được gọi là không trội nếu không tồn tại bất cứ lời giải nào trội hơn trong tập các lời giải.
- Biên pareto: Tập các lời giải không trội.
- Khoảng cách quần thể: Hệ số để hình thành đường bao giữa các lời giải không trội. Theo giải thích hình học, mỗi lời giải không trội được thể hiện là một điểm trên mặt phẳng, và khoảng cách quần thể được tính dựa trên ước lượng khoảng cách với hai điểm liền kề (các lời giải ở đầu mút sẽ có khoảng cách quần thể là vô cùng). Phép tính toán khoảng cách quần thể được trình bày trong mã giả 9
- Hạng: Hạng của mỗi lời giải được tính dựa trên số lượng lời giải trội hơn. Hạng càng nhỏ thì lời giải càng tốt (càng bị ít lời giải khác trội hơn).
- Toán tử so sánh: Khi sắp xếp các lời giải trong NSGA-II, lời giải A tốt hơn lời giải B nếu $A_{rank} < B_{rank}$, nếu $A_{rank} = B_{rank}$ thì khoảng cách

quần thể của A lớn hơn khoảng cách quần thể của B. Toán tử so sánh kí hiệu là $<$.

3.2.2 Áp dụng cho bài toán tối ưu đa mục tiêu

Giải thuật di truyền sắp xếp không trội được trình bày trong mã giả 6.

Algorithm 6 Giải thuật di truyền sắp xếp không trội

Require: Đồ thị không hoàn chỉnh vô hướng $G=(V,E)$ với nút gốc $v_0 \in V$

Ensure: Tập các cây Steiner trên biên Pareto

- 1: Sinh n cá thể của quần thể P_0
 - 2: **for** cá thể $i \in P_0$ **do**
 - 3:
 - Thực hiện giải thuật Kruskal trên tập các cạnh đã xáo trộn để tạo thành một cây;
 - Thực hiện quy trình sửa chữa để thu được cây Steiner;
 - Tính toán hàm phạt của i ;
 - 4: **end for**
 - 5: $t \leftarrow 0$
 - 6: **while** điều kiện dừng chưa thỏa mãn **do**
 - $Q_t \leftarrow$ Thực hiện lai ghép và đột biến trên P_t ;
 - $R_t \leftarrow Q_t \cup P_t$;
 - $F \leftarrow$ Thực hiện sắp xếp nhanh trên R_t (mã giả 7);
 - $P_t + 1 \leftarrow$ Lựa chọn quần thể cho thế hệ $t+1$ (mã giả 8);
 - $t \leftarrow t + 1$;
 - 7: **end while**
-

Algorithm 7 Quy trình sắp xếp nhanh cho NSGA-II

Require: Tập các cây Steiner (danh sách các cá thể ở thế hệ hiện tại).

Ensure: Danh sách F của danh sách các cá thể, các phần tử thuộc cùng danh sách F_i sẽ có hạng là i .

```
1:  $F \leftarrow []$ 
2: for  $p \in P$  do
3:    $S_p \leftarrow []$  (tập các cá thể mà  $p_i$  trội hơn)
4:    $n_p \leftarrow 0$  (số lượng các cá thể trội hơn  $p_i$ )
5:   for cá thể  $q$  của quần thể  $P$  do
6:     if  $p \prec q$  then
7:        $S_p \leftarrow S_p \cup q$ 
8:     else
9:        $n_p \leftarrow n_p + 1$ 
10:    end if
11:  end for
12:  if  $n_p = 0$  then
13:     $p_{rank} \leftarrow 0$ 
14:     $F[0] \leftarrow F[0] \cup p$ 
15:  end if
16: end for
17:  $i \leftarrow 0$ 
18: while  $F[i] \neq \emptyset$  do
19:    $Q \leftarrow []$ 
20:   for  $p \in F[i]$  do
21:     for  $q \in S_p$  do  $n_q \leftarrow n_q - 1$ 
22:     if  $n_q = 0$  then
23:        $q_{rank} = i + 1$ 
24:        $Q \leftarrow Q \cup q$ 
25:     end if
26:   end for
27: end for
28:    $i \leftarrow i + 1$ 
29:    $F[i] \leftarrow Q$ 
30: end while
```

Algorithm 8 Phép lựa chọn quần thể cho NSGA-II

Require: Danh sách F của danh sách các cá thể.

Ensure: Tập các cá thể với kích thước bằng n , bằng kích thước thể ban đầu.

```
1: for  $F[i] \in F$  do
2:   Tính toán khoảng cách quần thể cho danh sách  $F[i]$ 
3: end for
4:  $P \leftarrow []$ 
5:  $last - front \leftarrow 0$ 
6: for  $F[i] \in F$  do
7:   if  $len(P) + len(F[i]) < n$  then
8:     for  $p \in F[i]$   $P \leftarrow P \cup p$ 
9:   end for
10:   $last - front \leftarrow last\_front + 1$ 
11:  end if
12: end for
13:  $remaining - size \leftarrow n - len(P)$ 
14: if  $remaining - size > 0$  then
15:    $sort(F[last - front], <)$ 
16:    $P \leftarrow P \cup F[last - front][: remaining - size + 1]$ 
17: end if
```

Algorithm 9 Phép tính toán khoảng cách quần thể cho NSGA-II

Require: Tập các cá thể ở danh sách F_i (các cá thể có cùng hạng là i).

```
1: for  $j \in len(F_i)$  do
2:    $dist_j \leftarrow 0$ 
3: end for
4: for mục tiêu thứ  $i$  trong  $m$  mục tiêu do
5:    $z_{max} \leftarrow MIN$  (giá trị lớn nhất của mục tiêu thứ  $i$ )
6:    $z_{min} \leftarrow MAX$  (giá trị nhỏ nhất của mục tiêu thứ  $i$ )
7:   for  $p \in F_i$  do
8:      $z_{max} = max(z_{max}, p_i)$ 
9:      $z_{min} = min(z_{min}, p_i)$ 
10:  end for
11:   $dist_0 \leftarrow MAX$ 
12:   $dist_{len(F[i])-1} \leftarrow MAX$ 
13:  for  $j \in [1, len(F[i]) - 2]$  do
14:     $dist_j \leftarrow dist_j + (dist_{j+1} - dist_{j-1}) / (z_{max} - z_{min})$ 
15:  end for
16: end for
```

3.3 Hàm phạt

Có hai cách để đánh giá giữa các lời giải khác nhau trong bài toán đa mục tiêu: thực hiện chuẩn hóa các mục tiêu để đưa về một hàm số duy nhất hoặc sử dụng biên Pareto. Cả hai cách này đều có nhược điểm. Cách thứ nhất thì trọng số đánh giá của từng mục tiêu thường không được đánh giá một cách chính xác, còn cách thứ hai thì chỉ tập trung vào việc phân tích sự khác nhau giữa các phương án trội và bị trội, sự khác nhau giữa các lời giải trên cùng biên Pareto không được đề cập cụ thể.

Một cách đánh giá là tổng hợp từ hai cách trên được đề xuất trong ???. Trong đó, giá trị Pareto được sử dụng như một giá trị phạt để phân biệt giữa các lời giải trội và bị trội. Còn giá trị của các mục tiêu sẽ được chuẩn hóa với các giá trị cực đại và cực tiểu của từng mục tiêu trong suốt quá trình tính toán.

$$fitness(x) = \frac{1}{\sum_{i=1}^4 \overline{f}_i(x) + Pen(x)} \quad (3.1)$$

với

$$\overline{f}_i(x) = \frac{f_i(x) - t_i^{min}}{t_i^{max} - t_i^{min}} \quad (3.2)$$

trong đó t_i^{max} và t_i^{min} đại diện cho giá trị cực đại và cực tiểu của mục tiêu thứ i . $f_i(x)$ là giá trị của mục tiêu thứ i được đề cập ở chương trước.

và

$$Pen(x) = \begin{cases} 0 & x \text{ nằm trên biên Pareto,} \\ 1 & \text{trong trường hợp ngược lại} \end{cases} \quad (3.3)$$

Ở phương trình 3.1, giá trị càng cao càng thể hiện mức độ tối ưu của lời giải.

3.4 Thực nghiệm

3.4.1 Mô tả dữ liệu

Kết quả thực nghiệm được tiến hành trên tập dữ liệu mô phỏng địa hình các tỉnh thành Việt Nam [22]. Tập dữ liệu có tất cả 12- bộ dữ liệu, bao gồm 10 địa hình được đánh thứ tự từ T1 tới T10, mô tả ngắn gọn trong 3.1, bán kính cảm biến của các nút trong mỗi bộ dữ liệu nhận 1 trong 4 giá trị: 25, 30, 45, hoặc 50 mét, các nút cảm biến được phân bố trên các địa hình theo 1 trong 3 phân phối: phân phối chuẩn, phân phối đều, hoặc phân phối gamma. Dữ liệu được xây dựng theo chuẩn Digital Elevation Model.

Bảng 3.1: Tính chất các địa hình

Địa hình	Địa điểm	Tính chất
T1	Hồ Chí Minh	Thành phố nhiều tòa nhà đa dạng về độ cao, không có đồi núi, sông
T2	Vũng Tàu	Thành phố nhiều tòa nhà với độ cao cân đối, có đồi núi, và một vùng biển nằm về một hướng
T3	Phú Quốc	Vùng đảo có đồi núi với độ cao thấp khác nhau, vùng biển là chính
T4	Đồng Tháp	Vùng đồng bằng ít nhà cửa, sông, kênh rạch nhiều, không đồi núi
T5	Vĩnh Long	Ít nhà cửa, không đồi núi, kênh rạch nhiều, đặc biệt có dòng sông lớn Mekong
T6	Lâm Đồng	Vùng đồi núi nhiều, ít nhà cửa, có núi cao, có sông xen kẽ
T7	Cao Nguyên	Vùng có nhiều đồi núi có độ cao tăng dần, ít nhà cửa, không sông hồ
T8	Đà Nẵng	Vùng ít nhà cửa, núi cao tiếp giáp vùng biển
T9	Hà Nội	Vùng nhiều nhà cao, có ao hồ
T10	Hạ Long	Vùng biển có nhiều đảo nhỏ độ cao khác nhau, không có nhà

3.4.2 Tham số cố định

Các tham số cố định được sử dụng trong quá trình thực nghiệm được đề cập trong bảng 3.2 và 3.3.

Bảng 3.2: Tham số sử dụng trong mô phỏng địa hình

Tham số	Giá trị
Kích thước địa hình	$500 \times 500m^2$
Số lượng nút cảm biến	100
Số lượng nút trung gian	100
Bán kính cảm biến	25-50 m

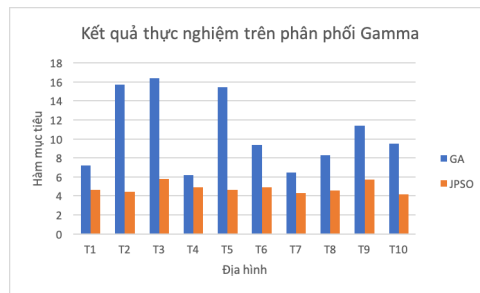
Bảng 3.3: Tham số cài đặt trong các giải thuật

Tham số	Giá trị
Kích thước quần thể	50
Tỉ lệ lai ghép	0.8
Tỉ lệ đột biến	0.05
Số lượng thế hệ	150

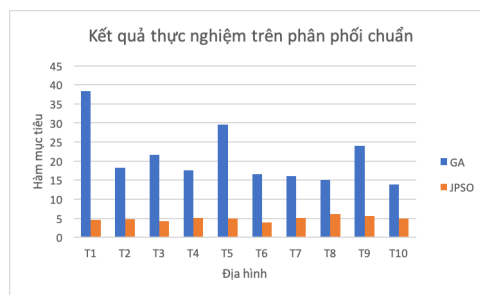
Các giải thuật được cài đặt bằng Python3 và chạy trên Intel Core i5-5200u 2.20 GHz, 8GB RAM, hệ điều hành Ubuntu 16.04.

3.4.3 Kết quả thực nghiệm của GA và JPSO trên các phân phối khác nhau

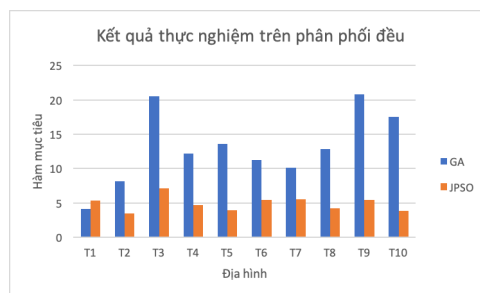
Dưới đây là kết quả thực nghiệm của giải thuật GA và JPSO nêu trong [18] khi áp dụng trên ba phân phối: phân phối chuẩn, phân phối đều và phân phối Gamma, với bán kính giao tiếp của các cảm biến đều là 30 mét.



Hình 3.4: Giá trị trung bình hàm mục tiêu của GA và JPSO trên phân phối Gamma.



Hình 3.5: Giá trị trung bình hàm mục tiêu của GA và JPSO trên phân phối chuẩn.



Hình 3.6: Giá trị trung bình hàm mục tiêu của GA và JPSO trên phân phối đều.

Nhận thấy trên cả ba phân phối, giá trị hàm mục tiêu thu được từ GA luôn tốt hơn JPSO, với sự khác biệt được thể hiện rõ nhất ở phân phối chuẩn. Một điều đáng lưu ý là kết quả của JPSO không có sự sai khác quá nhiều giữa các địa hình, cho thấy thuật toán có khả năng đã rơi vào tối ưu lân cận trên bộ dữ liệu thực nghiệm. Sự khác biệt nhiều nhất giữa hai giải thuật

được thể hiện ở phân phối chuẩn, với các nút cảm biến được bố trí gần hơn với trạm xử lý trung tâm so với hai phân phối còn lại.

3.4.4 Tổng hợp kết quả thực nghiệm của GA và JPSO

Bảng 3.4.4 tổng hợp kết quả thực nghiệm của GA và JPSO trên 120 bộ dữ liệu, bao gồm 10 loại địa hình, 3 loại phân phối và 4 loại bán kính khác nhau.

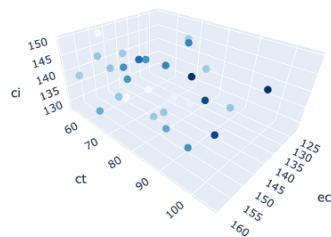
Bảng 3.4: Bảng tổng hợp kết quả thực nghiệm.

Phân phối	Địa hình	r=25		r=30		r=45		r=50	
		GA	JPSO	GA	JPSO	GA	JPSO	GA	JPSO
Phân phối Gamma	T1	8.83	4.37	7.21	4.61	10.28	5.49	17.18	4.95
	T2	7.44	5.76	15.68	4.43	11.13	4.67	8.05	5.87
	T3	9.31	4.92	16.41	5.81	29.86	11.15	20.66	4.49
	T4	19.69	3.51	6.22	4.93	5.17	5.4	28.81	7.31
	T5	9.55	5.85	15.44	4.61	6.63	4.01	6.58	5.49
	T6	9.4	5.8	9.36	4.88	14.39	4.43	17.85	3.78
	T7	6.57	5.63	6.45	4.31	10.99	4.31	9.96	4.92
	T8	7.87	7.94	8.31	4.6	26.15	4.44	12.11	5.28
	T9	15.81	5.15	11.38	5.72	7.25	8.23	8.01	6.53
	T10	13.71	5.17	9.52	4.18	9.54	5.35	8.39	5.27
Phân phối chuẩn	T1	22.44	7.44	38.41	4.57	20.5	4	15.12	4.28
	T2	23.83	5.27	18.22	4.76	5.89	6.86	11.26	3.42
	T3	15.8	4.85	21.63	4.26	18.14	7.3	43.12	4.4
	T4	20.78	5.05	17.61	5.15	26.91	4.02	11.96	3.77
	T5	11.77	7.31	29.63	4.86	12.4	3.34	25.33	4.94
	T6	23.92	5.38	16.55	3.96	20.99	4.67	13.31	5.52
	T7	20.12	4.84	16.09	5.07	20.53	6.67	8.28	6.85
	T8	25.85	4.44	15	6.18	22.17	5.45	18.11	4.42
	T9	14.86	6.13	24.01	5.67	6.48	3.52	8.05	3.72
	T10	11.09	4.29	13.91	4.92	13.93	5.27	22.76	4.68
Phân phối đều	T1	27.23	5.02	4.14	5.37	11.38	6	12.29	4.49
	T2	6.7	5.65	8.17	3.43	7.34	3.97	8.01	4.81
	T3	31.4	5.5	20.5	7.11	5.08	3.64	8.74	4.27
	T4	16.55	4.34	12.16	4.66	20.31	5.31	14.69	4.14
	T5	9.43	4.18	13.57	3.95	10.52	4.53	18.19	4.15
	T6	28.29	5.01	11.28	5.4	17.66	3.95	12.74	5.83
	T7	11.73	6.15	10.12	5.55	6.8	6.1	18.77	3.86
	T8	17.95	4.97	12.85	4.18	14.34	5.02	8.11	6.06
	T9	27.23	5.23	20.83	5.45	16.54	5.53	30.81	3.93
	T10	13.79	6.86	17.54	3.82	14.02	4.18	10.39	5.49

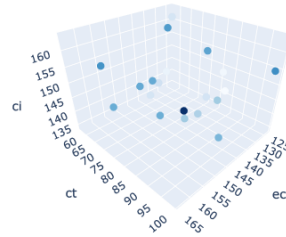
3.4.5 Kết quả thực nghiệm của NSGA-II

Dưới đây là hình vẽ biên Pareto trên các bộ dữ liệu với phân phối khác nhau, bán kính các nút cảm biến đều là 30 mét.

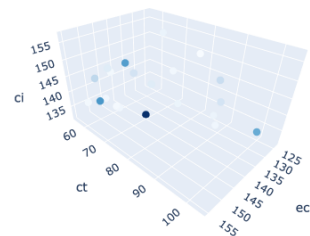
- Phân phối Gamma



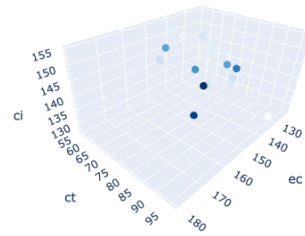
Hình 3.7: Gamma - T1



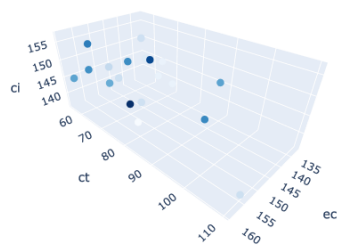
Hình 3.8: Gamma - T2



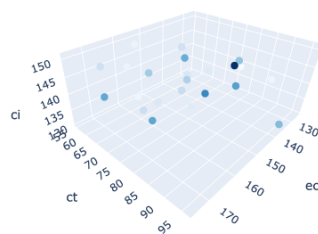
Hình 3.9: Gamma - T3



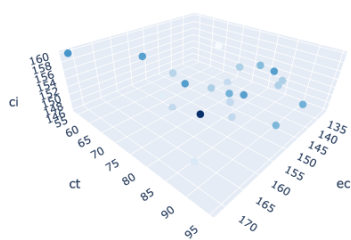
Hình 3.10: Gamma - T4



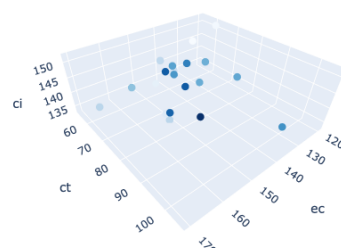
Hình 3.11: Gamma - T5



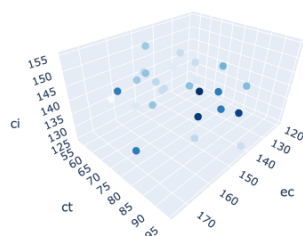
Hình 3.12: Gamma - T6



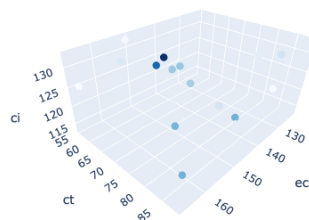
Hình 3.13: Gamma - T7



Hình 3.14: Gamma - T8

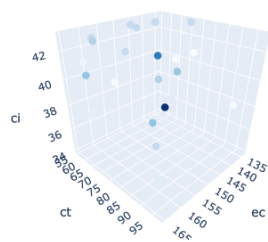


Hình 3.15: Gamma - T9

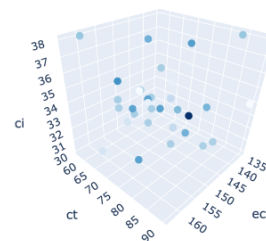


Hình 3.16: Gamma - T10

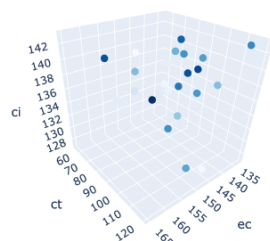
- Phân phối Chuẩn



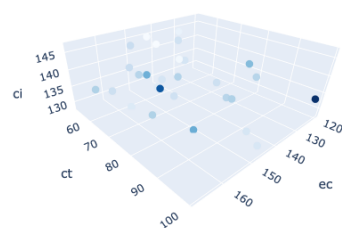
Hình 3.17: No - T1



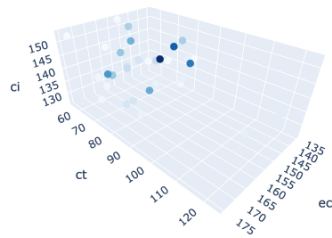
Hình 3.18: No - T2



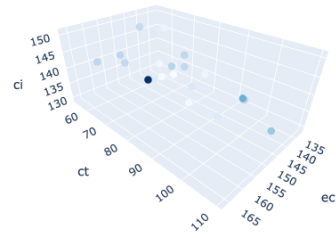
Hình 3.19: No - T3



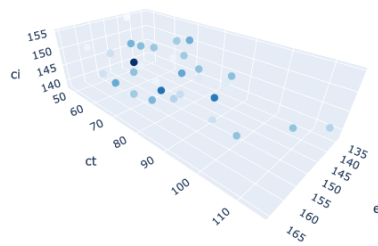
Hình 3.20: No - T4



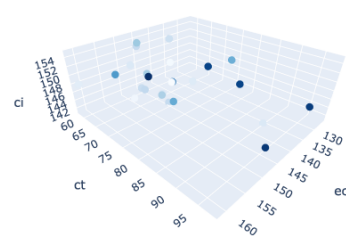
Hình 3.21: No - T5



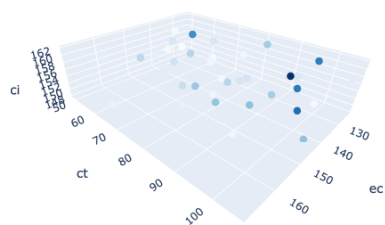
Hình 3.22: No - T6



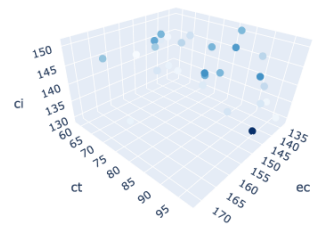
Hình 3.23: No - T7



Hình 3.24: No - T8

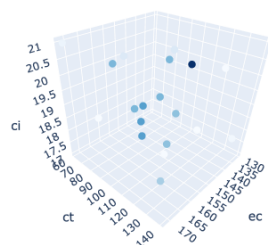


Hình 3.25: No - T9

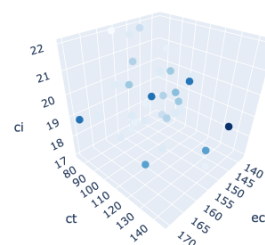


Hình 3.26: No - T10

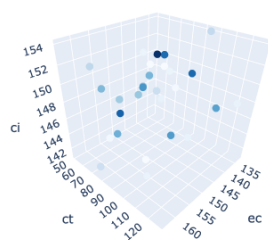
- Phân phối Đều



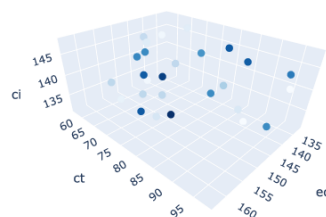
Hình 3.27: Uniform - T1



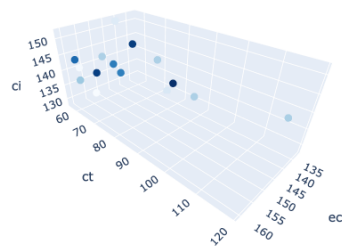
Hình 3.28: Uniform - T2



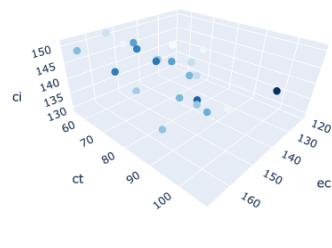
Hình 3.29: Uniform - T3



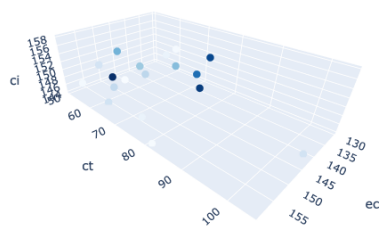
Hình 3.30: Uniform - T4



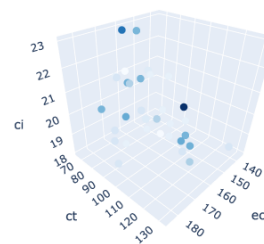
Hình 3.31: Uniform - T5



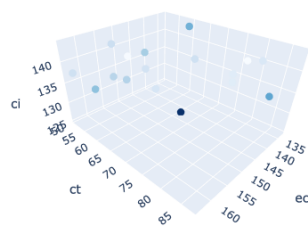
Hình 3.32: Uniform - T6



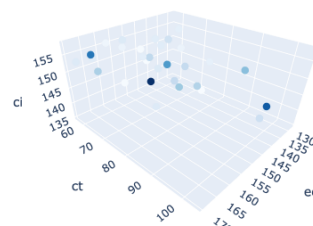
Hình 3.33: Uniform - T7



Hình 3.34: Uniform - T8



Hình 3.35: Uniform - T9



Hình 3.36: Uniform - T10

Kết luận

Đồ án tập trung nghiên cứu về giải thuật tiến hoá trên mạng cảm biến không dây, cụ thể là giải thuật di truyền và giải thuật di truyền sắp xếp không trội. Kiến thức tổng quan về mạng cảm biến không dây, bao gồm kiến trúc các loại mạng cảm biến, cấu trúc một nút cảm biến được đề cập ở chương 1, các mục tiêu được quan tâm và nghiên cứu của đồ án được đề cập ở chương 2. Chương 3 và chương 4 là phần giải thích và trình bày kết quả khi thực nghiệm với hai giải thuật, giải thuật di truyền và giải thuật di truyền sắp xếp không trội.

Đồ án có thể được mở rộng theo ba hướng. Hướng thứ nhất là áp dụng các giải thuật tiến hoá đa mục tiêu khác trên bộ dữ liệu hiện có để so sánh với kết quả hiện tại, lựa chọn giải thuật mang lại kết quả tốt nhất. Hướng thứ hai là thay đổi quy trình thực hiện của hai giải thuật đã áp dụng, có thể thay đổi ở bước mã hoá, lai ghép hoặc đột biến. Hướng còn lại là kết hợp từ cả hai hướng trên.

Dẫu còn nhiều thiếu sót cần cải thiện, đồ án đã đóng góp một phần nhỏ trong lĩnh vực nghiên cứu sử dụng giải thuật tiến hoá nhằm tối ưu các mục tiêu trên mạng cảm biến không dây.

Tài liệu tham khảo

- [1] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325–349, 2005.
- [2] Gopi Saminathan Arumugam and Thirumurugan Ponnuchamy. Ee-leach: development of energy-efficient leach protocol for data gathering in wsn. *EURASIP Journal on Wireless Communications and Networking*, 2015(1):1–9, 2015.
- [3] Parul Bakaraniya and Sheetal Mehta. K-leach: An improved leach protocol for lifetime improvement in wsn. *International Journal of Engineering Trends and Technology (IJETT)*, 4(5):1521–1526, 2013.
- [4] Application Benefit. The evolution of wireless sensor networks. *Silicon Lab. Inc., Austin, TX, USA, White Paper Rev*, 1:1–5, 2013.
- [5] Martin Burkhart, Pascal Von Rickenbach, Rogert Wattenhofer, and Aaron Zollinger. Does topology control reduce interference? In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 9–19, 2004.
- [6] IAlaa Eisa, Hazem M El-Bakry, and Samir M Abd Elrazik. Challenges in wireless sensor networks. *International Journal of Advanced Research in Computer Science and Technology*, 2013.
- [7] Zahra Eskandari, Mohammad Hossein Yaghmaee, and AmirHossien Mohajerzadeh. Energy efficient spanning tree for data aggregation in wireless sensor networks. In *2008 Proceedings of 17th International Conference on Computer Communications and Networks*, pages 1–5. IEEE, 2008.
- [8] Elena Fasolo, Michele Rossi, Jorg Widmer, and Michele Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, 14(2):70–87, 2007.

- [9] Wendi B Heinzelman, Anantha P Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications*, 1(4):660–670, 2002.
- [10] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual Hawaii international conference on system sciences*, pages 10–pp. IEEE, 2000.
- [11] Hung-Chin Jang. Applications of geometric algorithms to reduce interference in wireless mesh network. *arXiv preprint arXiv:1003.3569*, 2010.
- [12] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.
- [13] Sai Krishna Kovi, Pavankumar Jangam, Sai Kumar Goud Kosgi, and MD Frostburg. Wireless sensor networks and applications.
- [14] Marc Lee and Vincent WS Wong. An energy-aware spanning tree algorithm for data aggregation in wireless sensor networks. In *PACRIM. 2005 IEEE Pacific Rim Conference on Communications, Computers and signal Processing, 2005.*, pages 300–303. IEEE, 2005.
- [15] Marc Lee and Vincent WS Wong. Lpt for data aggregation in wireless sensor networks. In *GLOBECOM’05. IEEE Global Telecommunications Conference, 2005.*, volume 5, pages 6–pp. IEEE, 2005.
- [16] Bing-Hong Liu, Van-Trung Pham, Tu N Nguyen, and Yi-Sheng Luo. A heuristic for maximizing the lifetime of data aggregation in wireless sensor networks. *arXiv preprint arXiv:1910.05310*, 2019.
- [17] Tiancheng Lou, Haisheng Tan, Yuexuan Wang, and Francis CM Lau. Minimizing average interference through topology control. In *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pages 115–129. Springer, 2011.
- [18] Yao Lu, Jianping Chen, Ioan Comsa, Pierre Kuonen, and Beat Hirsbrunner. Construction of data aggregation tree for multi-objectives in wireless sensor networks through jump particle swarm optimization. *Procedia Computer Science*, 35:73–82, 2014.

- [19] Sandip Mukherjee, Pawan Kumar Joshi, Samadrita Mukherjee, Anirudha Ghosh, RD Garg, and Anirban Mukhopadhyay. Evaluation of vertical accuracy of open source digital elevation model (dem). *International Journal of Applied Earth Observation and Geoinformation*, 21:205–217, 2013.
- [20] SR Jino Ramson and D Jackuline Moni. Applications of wireless sensor networks—a survey. In *2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT)*, pages 325–329. IEEE, 2017.
- [21] Robert Sedgewick and Kevin Wayne. *Algorithms*. Addison-wesley professional, 2011.
- [22] Nguyen Thi Tam, Dang Thanh Hai, et al. Improving lifetime and network connections of 3d wireless sensor networks based on fuzzy clustering and particle swarm optimization. *Wireless Networks*, 24(5):1477–1490, 2018.
- [23] Baobing Wang and John S Baras. Minimizing aggregation latency under the physical interference model in wireless sensor networks. In *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pages 19–24. IEEE, 2012.
- [24] XiaoHua Xu, Xiang Yang Li, XuFei Mao, Shaojie Tang, and ShiGuang Wang. A delay-efficient algorithm for data aggregation in multihop wireless sensor networks. *IEEE transactions on parallel and distributed systems*, 22(1):163–175, 2010.
- [25] Waleed Youssef and Mohamed Younis. Intelligent gateways placement for reduced data latency in wireless sensor networks. In *2007 IEEE International Conference on Communications*, pages 3805–3810. IEEE, 2007.