

Chương I: Cơ sở lý thuyết

I. Bài toán tối ưu

1. Tối ưu liên tục
2. Tối ưu tổ hợp
- + Khái niệm
- + Bài toán NP khó

II. Các phương pháp giải bài toán tối ưu

1. Các phương pháp giải chính xác
2. Các phương pháp giải xấp xỉ

III. Giải thuật di truyền

1. Mã hóa cá thể
2. Khởi tạo quần thể
3. Các toán tử di truyền
 - 3.1. Lai ghép
 - 3.2. Đột biến
4. Hàm thích nghi
5. Chọn lọc
6. Điều kiện dừng của giải thuật
7. Đánh giá giải thuật di truyền

IV. Lý thuyết đồ thị

1. Các khái niệm cơ bản về đồ thị
2. Bài toán luồng cực đại

Chương II: Bài toán tối ưu thời gian sống của mạng cảm biến không dây

I. Tổng quan về mạng cảm biến không dây

II. Bài toán tối ưu thời gian sống của mạng cảm biến không dây

III. Các nghiên cứu liên quan

IV. Mô hình bài toán mà đề án đang giải quyết

Chương III: Giải thuật di truyền đề xuất giải bài toán tối ưu thời gian sống của mạng cảm biến không dây

I. Mã hóa cá thể

II. Khởi tạo quần thể

III. Các toán tử di truyền

1. Lai ghép

2. Đột biến

IV. Hàm thích nghi

V. Chọn lọc

Chương IV: PP tìm kiếm cục bộ giải bài toán tối ưu thời gian sống của mạng cảm biến không dây

I. Tabu search

II. Luồng cực đại trong mạng với chi phí

Chương V: Kết quả thực nghiệm

I. Đánh giá cận lỏng của bài toán

II. Môi trường thực nghiệm và các tham số

III. Kết quả và các bảng biểu, đồ thị

CHƯƠNG I. CƠ SỞ LÝ THUYẾT

I. Bài toán tối ưu

Trong cuộc sống hàng ngày, ta thường phải giải quyết rất nhiều vấn đề. Những vấn đề này có thể chỉ có một cách giải quyết, tuy nhiên cũng có những vấn đề có nhiều cách giải quyết khác nhau, khi đó một nhu cầu tự nhiên được sinh ra đó là chọn cách giải quyết phù hợp nhất với vấn đề cũng như phù hợp với bản thân. Ví dụ như có nhiều con đường từ trường đến nhà, ta muốn chọn con đường ngắn nhất để tiết kiệm sức lực, xếp đồ vào vali, ta chọn cách sao cho xếp được càng nhiều đồ càng tốt,... Đó được gọi là những bài toán tối ưu hóa.

Trong khoa học máy tính, bài toán tối ưu là bài toán tìm ra lời giải “tốt” nhất trong các lời giải khả thi. Tùy vào bài toán khác nhau sẽ định nghĩa độ tốt khác nhau, thường được gọi chung là hàm mục tiêu của bài toán. Có nhiều cách để phân chia lớp các bài toán tối ưu, một cách thường gặp là dựa vào tính chất của biến trong bài toán để chia bài toán thành hai loại:

- + Bài toán tối ưu hóa liên tục: biến trong bài toán là biến liên tục
- + Bài toán tối ưu hóa rời rạc (hay còn gọi là tối ưu hóa tổ hợp): biến trong bài toán là biến rời rạc

Sau đây sẽ trình bày rõ ràng hơn về hai loại bài toán tối ưu trên

1. Tối ưu hóa liên tục

Dạng tiêu chuẩn của bài toán tối ưu hóa liên tục:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, 2, \dots, m \\ & h_i(x) = 0, \quad i = 1, 2, \dots, p \end{array}$$

Trong đó:

- $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$ là hàm mục tiêu cần đạt được
- $g_i(x) \leq 0$ được gọi là các ràng buộc bất đẳng thức
- $h_i(x) = 0$ được gọi là các ràng buộc đẳng thức
- $m, p \in \mathbb{N}$

Theo quy ước, dạng tiêu chuẩn xác định một bài toán cực tiểu hóa. Ta có thể định nghĩa bài toán cực đại hóa một cách tương tự.

Một ví dụ thường gặp của bài toán tối ưu liên tục là tối ưu giá trị của hàm số trong đoạn/khoảng cho trước

2. Tối ưu hóa tổ hợp

Như đã nêu ở trên, bài toán tối ưu tổ hợp là bài toán có biến quyết định nhận các giá trị rời rạc. Một cách tổng quát, một bài toán tối ưu tổ hợp có thể được định nghĩa bởi bộ ba (S, f, Ω) , trong đó S là tập hữu hạn trạng thái (lời giải tiềm năng hay phương án), f là hàm mục tiêu xác định trên S , Ω là tập các ràng buộc. Mỗi lời giải $s \in S$ thỏa mãn Ω được gọi là một lời giải chấp nhận được. Mục tiêu ở đây là tìm phương án s^* tối ưu hóa toàn cục hàm mục tiêu f . Chẳng hạn với bài toán cực tiểu hóa thì $f(s^*) \leq f(s)$ với mọi phương án chấp nhận được s .

Ta đi đến một số bài toán tiêu biểu của lớp các bài toán tối ưu hóa tổ hợp.

Bài toán người du lịch

Nội dung bài toán: một người du lịch muốn tham quan n thành phố T_1, T_1, \dots, T_n . Xuất phát từ một thành phố bất kì, người du lịch muốn đến tất cả các thành phố còn lại, mỗi thành phố đi qua đúng một lần rồi quay trở lại thành phố xuất phát. Gọi C_{ij} là chi phí đi từ thành phố T_i đến thành phố T_j , hãy tìm một hành trình thỏa mãn yêu cầu bài toán sao cho tổng chi phí di chuyển là nhỏ nhất.

Bài toán được mô hình hóa như sau

Đầu vào:

- n : số thành phố
- $C_{n \times n}$: ma trận kích thước $n \times n$ trong đó C_{ij} là chi phí di chuyển từ thành phố i đến thành phố j

Đầu ra:

- Hành trình đi qua n thành phố $T_{\pi(1)}, T_{\pi(2)}, \dots, T_{\pi(n)}$

Mục tiêu:

- $f = C_{\pi(1), \pi(2)} + \dots + C_{\pi(n), \pi(1)} \rightarrow \text{minimize}$

Ràng buộc:

- $\pi(i) \neq \pi(j) \forall 0 \leq i, j \leq n$

Một cách lí thuyết, bài toán người du lịch sẽ có tất cả n^n lời giải, tuy nhiên sẽ chỉ có $n!$ lời giải hợp lệ do ràng buộc mỗi thành phố chỉ được phép đi qua một lần. Nhiệm vụ tối ưu là tìm lời giải có tổng chi phí di chuyển nhỏ nhất trong $n!$ lời giải hợp lệ.

Bài toán cái túi

Nội dung bài toán: một nhà thám hiểm cần đem theo một cái túi có trọng lượng không quá b . Có n đồ vật có thể đem theo, đồ vật thứ i có trọng lượng a_i và giá trị sử dụng c_i ($i = 1, 2, \dots, n$). Hỏi nhà thám hiểm cần đem theo những đồ vật nào để tổng giá trị sử dụng là lớn nhất.

Một phương án của nhà thám hiểm có thể biểu diễn bởi một vector nhị phân có độ dài n : $z = (z_1, z_2, \dots, z_n)$, trong đó $z_i = 1$ có nghĩa đồ vật thứ i được mang theo và bằng 0 nếu ngược lại. Tổng giá trị các đồ vật đem theo là:

$$f(x) = \sum_{i=1}^n c_i x_i$$

Tổng trọng lượng đồ vật đem theo là:

$$g(x) = \sum_{i=1}^n a_i x_i$$

Như vậy bài toán có thể mô hình hóa như sau:

Đầu vào:

- n đồ vật
- b là trọng lượng tối đa có thể mang theo
- $\mathbf{a} = (a_1, a_2, \dots, a_n)$: a_i là trọng lượng của đồ vật thứ i
- $\mathbf{c} = (c_1, c_2, \dots, c_n)$: c_i là giá trị của đồ vật thứ i

Đầu ra:

- Biến nhị phân $\mathbf{z} = (z_1, z_2, \dots, z_n)$

Hàm mục tiêu:

- $f(x) = \sum_{i=1}^n c_i x_i \rightarrow \max$

Ràng buộc:

- $g(x) = \sum_{i=1}^n a_i x_i \leq b$

II. Các phương pháp giải bài toán tối ưu

Giải bài toán tối ưu chính là đi tìm đáp án tốt nhất trong hàng loạt các đáp khả thi của không gian lời giải. Ta có thể thu được lời giải tối ưu hàm mục tiêu trong không gian này nhưng đôi khi ta cũng chỉ có thể thu được lời giải “được coi là” tốt, liên quan đến hai khái niệm sau

- + Tối ưu toàn cục: lời giải tốt nhất thu được trong không gian lời giải của bài toán
- + Tối ưu cục bộ: lời giải tốt nhất trong một tập con của không gian lời giải.

Vậy vì sao lại dẫn đến trường hợp chỉ có kết quả “được coi là” tốt (tối ưu cục bộ)? Ta đi tìm qua các phương pháp giải bài toán tối ưu để làm rõ vấn đề này

1. Các phương pháp giải chính xác

Như tên gọi, phương pháp giải chính xác cho bài toán tối ưu là phương pháp đem lại lời giải tốt nhất trong toàn bộ không gian lời giải của bài toán. Tiêu biểu là một số phương pháp sau

Vét cạn

Vét cạn là kỹ thuật với ý tưởng rất đơn giản: thử tất cả các phương án có thể của bài toán, kiểm tra liệu lời giải có vi phạm ràng buộc, kết luận phương án nào là tối ưu.

Phương pháp này có thể đảm bảo lời giải được tìm ra là tối ưu do đã xét mọi trường hợp có thể xảy ra, tuy nhiên lại gặp một trở ngại rất lớn: thời gian. Xét với một ví dụ đơn giản là bài toán người du lịch đã nêu ở trên, tập các lời giải khả thi là $n!$, chỉ cần tăng thêm một thành phố vào bài toán thì số lời giải khả thi đã tăng theo cấp số nhân, do đó phương pháp vét cạn hiển nhiên không thể áp dụng để giải bài toán này một cách hiệu quả.

Nhánh cận

Ý tưởng của phương pháp nhánh cận đó là nếu ta dự đoán trước được những phương án có hàm mục tiêu chắc chắn tồi hơn hàm mục tiêu của phương án đã có trước đó, vậy ta không cần thử qua những phương án này nữa, nhờ đó thu hẹp được không gian tìm kiếm. Đây là một cải tiến của phương pháp vét cạn, cho phép tìm ra được lời giải nhanh hơn.

Trong phương pháp nhánh cận, ta sẽ xây dựng từng phần của lời giải cho đến khi tìm ra được lời giải tối ưu. Mô hình hóa lời giải thành vector $x = (x_1, x_2, \dots, x_n)$, giả sử tại bước k ta đã xây dựng được k thành phần của lời giải từ x_1 đến x_k , cần mở rộng thành phần x_{k+1} , nhưng khi đánh giá lại ta thấy tất cả các nghiệm mở rộng từ x_k không có nghiệm nào có giá trị tốt hơn giá trị tối ưu ta đã biết tại thời điểm đó, vậy ta không cần mở rộng nữa, như vậy ta đã cắt bỏ đi một nhánh.

Điều khó là phải đánh giá được các thành phần mở rộng, nếu đánh giá được tốt, thuật toán nhánh cận sẽ chạy nhanh hơn nhiều so với vét cạn. Tuy nhiên nhánh cận vẫn chỉ có thể áp dụng trong các bài toán có quy mô nhỏ.

Chia để trị

Chia để trị là một phương pháp áp dụng cho các bài toán có thể giải quyết bằng cách chia nhỏ ra thành các bài toán con từ việc giải quyết các bài toán này. Sau đó lời giải của bài toán nhỏ được tổng hợp lại thành lời giải cho bài toán ban đầu.

Kỹ thuật chia để trị là cơ sở cho nhiều thuật toán hiệu quả, chẳng hạn như thuật toán sắp xếp, thuật toán nhân, thuật toán phân tích cú pháp.

Kỹ thuật chia để trị thông qua ba bước:

- Chia/tách nhỏ: tại bước này bài toán ban đầu được tách thành các bài toán con cho đến khi không thể tách được nữa, các bài toán con sẽ trở thành một bước nhỏ trong việc giải quyết bài toán lớn

- Trị/giải quyết bài toán con: tại bước này sẽ tìm cách giải quyết bài toán con một cách cụ thể
- Tổng hợp: khi đã giải quyết được các bài toán con, tổng hợp lại để có được lời giải của bài toán ban đầu.

Quy hoạch động

Được nhà toán học Richard Bellman phát minh vào năm 1953, quy hoạch động là một phương pháp mạnh mẽ tìm lời giải chính xác của các bài toán tối ưu. Phương pháp này thường dùng để giải các bài toán có cấu trúc con tối ưu và bài toán có các bài toán con gối nhau.

Bài toán con gối nhau, tương tự như chia để trị, quy hoạch động cũng chia bài toán ban đầu thành các bài toán con nhỏ hơn. Quy hoạch động được sử dụng khi các bài toán con này gọi đi gọi lại, kết quả của các bài toán con được lưu lại, do đó giảm thời gian tính toán.

Quy hoạch động sẽ không thể áp dụng được khi các bài toán con không gối nhau, do mỗi lần chia nhỏ, mỗi bài toán con chỉ được gọi một lần mà không được gọi lại.

Cấu trúc con tối ưu có nghĩa là các lời giải tối ưu cho các bài toán con có thể được sử dụng để tìm các lời giải tối ưu cho bài toán toàn cục. Ví dụ, đường đi ngắn nhất tới một đỉnh trong một đồ thị có thể được tìm thấy bằng cách: trước hết tính đường đi ngắn nhất tới đích từ tất cả các đỉnh kề nó, rồi dùng kết quả này để chọn đường đi toàn cục tốt nhất. Nói chung, ta có thể giải một bài toán với cấu trúc con tối ưu bằng một quy trình ba bước:

- Chia bài toán thành các bài toán con nhỏ hơn.
- Giải các bài toán này một cách tối ưu bằng cách sử dụng đệ quy quy trình ba bước này.
- Sử dụng các kết quả tối ưu đó để xây dựng một lời giải tối ưu cho bài toán ban đầu.

Các bài toán con được giải bằng cách chia chúng thành các bài toán con nhỏ hơn, và cứ tiếp tục như thế cho đến khi ta đến được trường hợp đơn giản để tìm lời giải.

Quy hoạch động thường dùng thường dùng hai cách tiếp cận

- Top-down (từ trên xuống): bài toán ban đầu chia nhỏ thành các bài toán con, các bài toán con này được giải và lời giải được ghi nhớ lại để phòng khi được dùng. Đây là đệ quy và lưu trữ kết hợp với nhau
- Bottom-up (từ dưới lên): tất cả các bài toán con có thể cần đến đều được giải trước, sau đó được dùng để xây dựng lời giải cho bài toán lớn hơn. Cách tiếp cận này tốt hơn về không gian bộ nhớ dùng cho ngăn xếp và số lời gọi hàm. Tuy nhiên đôi khi việc xác định tất cả các bài toán con cần thiết cho việc giải quyết bài toán không trực giác cho lắm.

Trên đây ta đã điểm qua một số phương pháp giải chính xác cho bài toán tối ưu hóa. Tuy nhiên đôi khi không thể tìm ra phương pháp giải chính xác cho bài toán tối ưu ngoại trừ vét cạn, hoặc có thể tìm được

nhưng không gian lời giải bùng nổ cực kì nhanh khi kích thước bài toán tăng lên. Trong trường hợp đó, ta sẽ sử dụng các phương pháp giải xấp xỉ để giải quyết bài toán.

2. Các phương pháp giải xấp xỉ

Đến nay đã có rất nhiều phương pháp giải xấp xỉ được đề xuất để giải quyết các bài toán khác nhau. Dưới đây là một số ví dụ tiêu biểu

Thuật toán tham lam

Thuật toán tham lam giải quyết bài toán theo hướng luôn chọn tối ưu địa phương tại mỗi bước đi với hy vọng tìm được lời giải tối ưu toàn cục. Ví dụ với bài toán người du lịch, tại mỗi bước ta sẽ chọn thành phố tiếp theo bằng cách chọn thành phố gần nhất chưa đi qua.

Nói chung, thuật toán tham lam bao gồm năm phần:

- Tập hợp các ứng viên để tạo ra lời giải
- Một hàm lựa chọn để theo đó chọn ứng viên tốt nhất để bổ sung vào lời giải
- Một hàm khả thi, theo đó quyết định một ứng viên có phù hợp để đưa vào lời giải.
- Hàm mục tiêu, xác định giá trị của lời giải
- Hàm đánh giá, chỉ ra khi nào lời giải hoàn chỉnh

Giải thuật tham lam quyết định sớm hướng đi và không bao giờ xét lại các quyết định cũ. Đối với một số bài toán, đây có thể là một thuật toán không chính xác.

Tìm kiếm cục bộ

Kỹ thuật tìm kiếm cục bộ đưa ra lời giải bằng cách tại mỗi bước cố gắng di chuyển đến lời giải láng giềng của lời giải hiện tại sao cho hàm mục tiêu được cải thiện. Các bước của kỹ thuật này như sau:

- Xuất phát từ một lời giải bất kỳ
- Áp dụng một phép biến đổi lên lời giải hiện hành để thu được một lời giải tốt hơn
- Lặp lại phép biến đổi lên phương án hiện tại cho đến khi không thể cải thiện được nữa.

Thông thường một phép biến đổi chỉ thay đổi một bộ phận nào đó của phương án hiện tại để được một phương án mới nên được gọi là biến đổi địa phương, vì thế kỹ thuật này có tên tìm kiếm cục bộ.

Các thuật toán tiến hóa

Thuật toán tiến hóa là tên gọi chung để chỉ lớp các thuật toán tìm kiếm và tối ưu hóa dựa trên nguyên lý tiến hóa tự nhiên. Một số thuật toán tiến hóa đã được công bố:

- Quy hoạch tiến hóa: do D.B.Pogel đề xuất. Có thể diễn tả quy hoạch tiến hóa như sau: cho một lớp các phương pháp khả dĩ giải quyết được một (số) phần của vấn đề. Dựa vào quy luật tiến hóa, tìm một phương pháp liên hợp đủ khả năng giải quyết trọn vẹn vấn đề đó.

- Chiến lược tiến hóa: do T.Baek, F.H.Hofmeister và H.P.Schwefel đề xuất. Thuật toán này dựa trên một số chiến lược ban đầu, tiến hóa để tạo ra những chiến lược mới phù hợp với môi trường thực tế một cách tốt nhất.
- Giải thuật di truyền: do J.H.Holland đề xuất, được D.E.Goldberg, L.Davis và Z.Michalewicz phát triển. Dựa trên quan niệm cho rằng quá trình tiến hóa tự nhiên là quá trình hoàn hảo nhất, hợp lý nhất và tự nó đã mang tính tối ưu.

Tiến hóa tự nhiên là một quá trình tối ưu. Đây có thể xem như một mệnh đề đúng, không chứng minh được, nhưng phù hợp với thực tế khách quan. Quá trình tiến hóa thể hiện tính tối ưu ở chỗ, thế hệ sau bao giờ cũng tốt hơn (phát triển hơn, hoàn thiện hơn) thế hệ trước. Tiến hóa tự nhiên được duy trì nhờ vào hai quá trình cơ bản: sinh sản và chọn lọc tự nhiên. Xuyên suốt quá trình tiến hóa tự nhiên, các thế hệ mới luôn được sinh ra để bổ sung thay thế thế hệ cũ. Cá thể nào phát triển hơn, thích ứng tốt hơn với môi trường sẽ tồn tại, ngược lại sẽ bị đào thải. Sự thay đổi môi trường là động lực thúc đẩy quá trình tiến hóa, đồng thời quá trình tiến hóa cũng làm thay đổi môi trường.

Các cá thể mới sinh ra nhờ sự lai ghép ở thế hệ cha mẹ. Một cá thể mới có thể mang các đặc tính của cha mẹ (di truyền) nhưng cũng có thể mang các đặc tính hoàn toàn khác (đột biến). Di truyền và đột biến có vai trò quan trọng như nhau trong quá trình tiến hóa, dù rằng đột biến xảy ra với xác suất nhỏ hơn nhiều so với di truyền. Các thuật toán tiến hóa tuy có những điểm khác biệt nhưng đều mô phỏng các quá trình cơ bản: lai ghép, đột biến, chọn lọc.

- Lai ghép: là quá trình sinh ra cá thể mới bằng cách thừa hưởng các đặc tính từ cá thể cha mẹ
- Đột biến: là hiện tượng cá thể con sinh ra mang một số tính trạng không có trong mã di truyền của cha mẹ
- Chọn lọc: dựa trên cơ sở độ thích nghi của các cá thể, chọn ra các cá thể tốt phù hợp với môi trường.

III. Giải thuật di truyền

Xuất hiện lần đầu vào năm 1962 trong công trình nghiên cứu về các hệ thống thích ứng của John Henry Holland, đại học Michigan, giải thuật tuy chưa nổi tiếng vào lúc đó nhưng đã đặt những nền móng đầu tiên, để sau này được tiếp tục phát triển bởi chính học trò của ông là nhà khoa học David Edward Goldberg. Đến năm 1975, giải thuật di truyền chính thức được trình bày trong cuốn sách *Adaptation in Natural and Artificial Systems* (Thích nghi trong tự nhiên và trong các hệ thống nhân tạo).

Thuật giải di truyền sử dụng các thuật ngữ vay mượn của di truyền học. Ta có thể nói về các cá thể (hay kiểu gen, cấu trúc) trong một quần thể, những cá thể này có thể được gọi là chuỗi (hay nhiễm sắc thể). Điều này có thể gây lẫn lộn: mỗi tế bào của một cơ thể chủng loại đã cho mang một số những nhiễm sắc thể nào đó nhưng trong thuật giải di truyền, ta chỉ nói về những cá thể có một nhiễm sắc thể. Các nhiễm

sắc thể được tạo thành từ các đơn vị (các gen) – biểu diễn trong một chuỗi tuyến tính. Mỗi gen kiểm soát một số đặc trưng.

Mỗi nhiễm sắc thể biểu diễn một lời giải của bài toán đang giải, một tiến trình tiến hóa được thực hiện trên một quần thể các các nhiễm sắc thể tương ứng với một quá trình tìm kiếm lời giải trong không gian lời giải. Tìm kiếm đó cân cân đối hai mục tiêu (có vẻ mâu thuẫn nhau): khai thác những lời giải tốt nhất và khảo sát không gian tìm kiếm, điểm mạnh của giải thuật di truyền là duy trì và xử lý một tập các lời giải (quần thể) trong suốt quá trình tìm kiếm, tạo được sự cân đối đáng kể giữa hai yêu tố trên. Nhờ việc duy trì nhiều lời giải này, giải thuật di truyền tìm kiếm trên nhiều điểm song song có khả năng leo lên nhiều cực trị cùng một lúc. Thông qua các toán tử của mình, giải thuật trao đổi thông tin giữa các cực trị với nhau, từ đó làm giảm thiểu khả năng giải thuật kết thúc tại các cực trị địa phương và bỏ qua mất cực trị toàn cục.

Ta đi vào chi tiết các bước của giải thuật di truyền

1. Mã hóa cá thể

Công việc đầu tiên cần thực hiện khi giải bài toán bằng giải thuật di truyền là chọn cách biểu diễn cá thể. Mã hóa cá thể là việc ánh xạ các tham số của bài toán lên một chuỗi có chiều dài xác định, tùy theo từng bài toán cụ thể mà có các cách biểu diễn khác nhau sao cho phù hợp, thuận lợi khi giải toán. Đây là công việc cực kì quan trọng, một cách biểu diễn mã hóa cá thể tốt quyết định tới 50% thành công của giải thuật.

Có nhiều cách mã hóa cá thể nhưng có hai cách biểu diễn thông dụng nhất là biểu diễn nhị phân và biểu diễn sử dụng hoán vị.

Với biểu diễn nhị phân, mỗi cá thể tương ứng với một chuỗi bao gồm các bit 0 và 1, ý nghĩa của các bit này phụ thuộc vào từng tính huống cụ thể. Đây là cách biểu diễn đơn giản và thông dụng nhất trong các cách biểu diễn.

Ví dụ trong bài toán cái túi được nêu ở trên, có một cách mã hóa cá thể đơn giản như sau: đánh thứ tự n đồ vật từ 1 đến n , mỗi cá thể được biểu diễn bằng một xâu nhị phân độ dài n , bit thứ i bằng 1 có nghĩa là đồ vật thứ i được chọn cho vào túi và bằng 0 nếu ngược lại.

Trong biểu diễn hoán vị, mỗi cá thể tương ứng với một hoán vị của tập n ký hiệu nào đó. Chẳng hạn với bài toán người du lịch đã được nêu ở trên, với n thành phố, ký hiệu cho các thành phố lần lượt là T_1, T_2, \dots, T_n . Mỗi cá thể, sự mã hóa của lời giải sẽ là một danh sách hoán vị của T_1, T_2, \dots, T_n biểu diễn lộ trình mà người du lịch đã đi qua.

2. Khởi tạo quần thể

Như đã nêu ở trên, giải thuật di truyền duy trì và xử lý một tập các lời giải trong suốt quá trình tìm kiếm, khởi tạo quần thể chính là quá trình sinh ra các cá thể một số lượng nhất định (kích thước quần thể).

Việc khởi tạo quần thể ban đầu cũng ảnh hưởng lớn tới kết quả cuối cùng, vì vậy việc khởi tạo các cá thể ban đầu của quần thể cũng cần sử dụng phương pháp phù hợp với bài toán. Có hai cách khởi tạo thường gặp là khởi tạo ngẫu nhiên và khởi tạo heuristic

- Khởi tạo ngẫu nhiên: cá thể được sinh ra một cách ngẫu nhiên, miễn là đáp ứng được các ràng buộc của bài toán
- Khởi tạo heuristic: dựa vào các kinh nghiệm đã có, ta khởi tạo tập các cá thể với hi vọng tập này có một khởi đầu tốt hơn nhằm nhanh chóng tìm ra lời giải.

3. Các toán tử di truyền

Các phép lai ghép, đột biến được gọi chung là các toán tử di truyền, các toán tử này sẽ quyết định chiều hướng tiến hóa của quần thể, do đó để đảm bảo kết quả tốt, ta cần có các toán tử lai ghép và đột biến phù hợp

3.1. Lai ghép

Quá trình lai ghép diễn ra bằng cách ghép một hay nhiều đoạn gen từ hai nhiễm sắc thể cha-mẹ để hình thành nhiễm sắc thể mới mang đặc tính cả cha lẫn mẹ. Các cá thể mới thừa hưởng các đặc tính tốt sẽ có khả năng sinh tồn mạnh hơn, nhờ đó duy trì nguồn gen tốt trong quần thể.

Một số phép lai ghép cơ bản: lai ghép một điểm cắt, lai ghép hai điểm cắt

Trong lai ghép một điểm cắt, một điểm phân cách được chọn ra, khi đó phần nằm phía sau điểm phân cách này của hai cá thể cha mẹ được hoán đổi với nhau sinh ra hai con mới. Tương tự với phép lai ghép hai điểm cắt, phần trao đổi là phần nằm giữa hai điểm cắt.

Tuy nhiên hai phép lai ghép trên phần lớn được áp dụng khi cách mã hóa cá thể là mã hóa nhị phân, với cách mã hóa cá thể khác (ví dụ mã hóa hoán vị), cá thể con sinh ra chưa chắc là một cá thể hợp lệ và do đó phép lai ghép coi như vô nghĩa. Trong trường hợp này cần áp dụng một số phép lai ghép khéo léo hơn.

3.2. Đột biến

Khác với đặc điểm của phép lai ghép là tận dụng nguồn gen tốt sẵn có trong quần thể, đột biến là quá trình tiến hóa khi một hoặc một số gen của cá thể con không được thừa hưởng từ gen của cha mẹ. Phép đột biến xảy ra với xác suất thấp hơn nhiều so với phép lai và có thể được mô tả như sau:

- Chọn ngẫu nhiên một cá thể trong quần thể
- Giả sử độ dài mã hóa của cá thể là n , chọn một số k ngẫu nhiên thỏa mãn $1 \leq k \leq n$, thay đổi gen thứ k sao cho cá thể vẫn phù hợp

- Trả cá thể này về quần thể để tham gia quá trình tiến hóa tiếp theo.

4. Hàm thích nghi

Hàm thích nghi phản ánh độ tương thích của cá thể với môi trường, hàm thích nghi có giá trị càng tốt chứng tỏ cá thể càng mạnh, là đối tượng phù hợp để giữ lại trong quá trình tiến hóa.

Có nhiều cách chọn hàm thích nghi khác nhau tùy vào bài toán, một cách chọn hàm thích nghi phổ biến là sử dụng luôn hàm mục tiêu trong mô hình bài toán đã đề ra.

5. Chọn lọc

Chọn lọc là quá trình loại bỏ các cá thể xấu và để lại những cá thể tốt. Một số phương pháp chọn lọc phổ biến như:

- Sắp xếp quần thể theo độ thích nghi, chỉ giữ lại các cá thể tốt nhất.
- Chọn lọc theo bánh xe Roulette: các cá thể được xếp vào một vòng tròn với độ rộng của cung tương ứng với độ thích nghi, quay vòng tròn bằng cách sinh ra một số ngẫu nhiên trong khoảng $(0, 1)$ rồi chọn ra cá thể tương ứng.

6. Điều kiện dừng của giải thuật

Điều kiện được đưa ra để đảm bảo tính dừng của giải thuật. Thường giải thuật di truyền dừng lại sau một số thế hệ xác định hoặc dừng khi sau một số thế hệ không cải thiện được hàm mục tiêu của bài toán. Đây là một tham số của bài toán sẽ được chọn ra một cách phù hợp trong quá trình thực nghiệm.

7. Đánh giá giải thuật di truyền

Ưu điểm:

- Nhanh hơn và tốn ít tài nguyên hơn so với tìm kiếm truyền thống trong một không gian tìm kiếm lớn.
- Đơn giản, nếu có cách mã hóa cá thể chuẩn xác và hàm mục tiêu đúng, ta có thể tìm được lời giải mà không cần thực hiện quá nhiều công việc phức tạp.

Nhược điểm:

- Đây không phải là một giải thuật hoàn chỉnh do không phải lúc nào giải thuật cũng tìm được lời giải phù hợp.
- Có thể bị mắc tại các cực bộ địa phương, mặc dù phép đột biến có thể giúp giảm thiểu trường hợp này nhưng làm giải thuật tiêu tốn nhiều thời gian hơn.
- Đôi khi khó để tìm được cách mã hóa cá thể.

IV. Lý thuyết đồ thị

1. Các khái niệm cơ bản về đồ thị

Lý thuyết đồ thị là một lĩnh vực nghiên cứu đã có từ lâu đời và có nhiều ứng dụng trong thế giới hiện đại. Những tư tưởng cơ bản của lý thuyết đồ thị được đề xuất vào những năm đầu của thế kỷ 18 bởi nhà toán học lỗi lạc người Thụy sỹ - Leonhard Euler. Chính ông là người đã sử dụng đồ thị để giải bài toán nổi tiếng về 7 cây cầu ở thành phố Königsberg.

Đồ thị là cấu trúc rời rạc có tính trực quan cao, rất tiện ích để biểu diễn các quan hệ. Có thể kể đến một số ứng dụng thực tế của đồ thị như:

- Có tiềm năng trong nhiều lĩnh vực do đồ thị có thể dùng để biểu diễn mối quan hệ, nghiên cứu quan hệ giữa các đối tượng là mục tiêu của nhiều lĩnh vực khác nhau.
- Ứng dụng trong mạng máy tính, mạng giao thông, mạng cung cấp nước, mạng điện, lập lịch, tối ưu hóa hệ thống, thiết kế mạch, quy hoạch phát triển,...
- Các ứng dụng khác: phân tích gen, trò chơi máy tính, chương trình dịch, thiết kế hướng đối tượng,...

Sau đây sẽ trình bày các khái niệm của đồ thị, nền tảng cho việc áp dụng lý thuyết đồ thị để giải quyết bài toán kéo dài thời gian sống của mạng cảm biến.

1.1. Định nghĩa đồ thị

Đồ thị là cấu trúc rời rạc gồm các đỉnh và cạnh nối giữa chúng. Các loại đồ thị khác nhau được phân biệt bởi kiểu và số lượng cạnh nối. Ta định nghĩa các loại đồ thị như sau:

Đơn (đa) đồ thị vô hướng $G = (V, E)$ là cặp gồm

- Tập đỉnh V là tập hữu hạn phần tử, các phần tử gọi là đỉnh
- Tập E là tập các bộ không có thứ tự dạng (u, v) , $u, v \in V$, $u \neq v$

Đơn (đa) đồ thị có hướng $G = (V, E)$ là cặp gồm

- Tập đỉnh V là tập hữu hạn phần tử, các phần tử gọi là đỉnh
- Tập E là tập các bộ có thứ tự dạng (u, v) , $u, v \in V$, $u \neq v$

1.2. Đường đi và chu trình

ĐN: đường đi P độ dài n từ đỉnh u đến đỉnh v , trong đó n là nguyên dương, trên đồ thị $G = (V, E)$ là dãy

- $P: x_0, x_1, \dots, x_n$
- Trong đó $u = x_0$, $v = x_n$, $(x_i, x_{i+1}) \in E$, $i = 0, 1, 2, \dots, n-1$
- Đỉnh u gọi là đỉnh đầu, đỉnh v gọi là đỉnh cuối của đường đi

Đường đi nói trên còn có thể biểu diễn dưới dạng dãy các cạnh: $(x_0, x_1), (x_1, x_2), \dots, (x_{n-1}, x_n)$

Đường đi gọi là đường đi đơn nếu không có đỉnh nào bị lặp lại trên nó.

Đường đi gọi là đường đi cơ bản nếu không có cạnh nào bị lặp lại trên nó.

Nếu có đường đi từ đỉnh u đến đỉnh v thì ta nói đỉnh v đạt đến được từ đỉnh u . Ta quan niệm rằng đỉnh v luôn đạt đến được từ chính nó.

ĐN: đường đi cơ bản có đỉnh đầu trùng với đỉnh cuối ($u = v$) được gọi là chu trình.

Chu trình được gọi là đơn nếu như ngoại trừ đỉnh đầu và đỉnh cuối không có đỉnh nào bị lặp lại.

2. Bài toán luồng cực đại

2.1. Các khái niệm

Khái niệm mạng: mạng là đồ thị có hướng $G = (V, E)$

- Có duy nhất một đỉnh s không có cung đi vào gọi là đỉnh phát (nguồn) và duy nhất một đỉnh t không có cung đi ra gọi là đỉnh thu (đích)
- Mỗi cung e của G được gắn với một số không âm $c(e)$ được gọi là khả năng thông qua của e

Luồng trong mạng: luồng f trong mạng $G = (V, E)$ là phép gán số $f(e)$ cho mỗi cạnh e ($f(e)$ được gọi là luồng trên cạnh e) thỏa mãn các điều kiện

i. Hạn chế về khả năng thông qua

Với mỗi cung e , $0 \leq f(e) \leq c(e)$

ii. Điều kiện cân bằng luồng: với mỗi $v \neq s, t$

$$\sum_{e \in E^-(v)} f(e) = \sum_{e \in E^+(v)} f(e)$$

Trong đó $E^-(v)$ và $E^+(v)$ tương ứng là tập cung đi vào và đi ra khỏi đỉnh v

Định nghĩa: giá trị của luồng f là

$$val(f) = \sum_{e \in E^-(s)} f(e) = \sum_{e \in E^+(t)} f(e)$$

Định nghĩa: luồng trong mạng G được gọi là luồng cực đại nếu trong số tất cả các luồng trong mạng G nó là luồng có giá trị lớn nhất. Bài toán tìm luồng cực đại trong mạng G được gọi là bài toán luồng cực đại.

ĐN: lát cắt là cách phân hoạch tập đỉnh (S, T) sao cho $s \in S, t \in T$

Khả năng thông qua $caps(S, T)$ của lát cắt (S, T) là số:

$$caps(S, T) = \sum_{e \in S \rightarrow T} c(e)$$

Trong đó $S \rightarrow T := \{(v, w) \in E: v \in S, w \in T\}$

Lát cắt nhỏ nhất (hẹp nhất) là lát cắt với khả năng thông qua nhỏ nhất.

Bổ đề: giả sử f là luồng, và (S, T) là lát cắt. Khi đó giá trị luồng chảy qua lát cắt chính bằng giá trị luồng

$$\sum_{e \in S \rightarrow T} f(e) - \sum_{e \in T \rightarrow S} f(e) = \sum_{e \in E^+(s)} f(e) = val(f)$$

Trong đó $S \rightarrow T := \{(v, w) \in E: v \in S, w \in T\}$ và $T \rightarrow S := \{(v, w) \in E: v \in T, w \in S\}$

Bổ đề: giả sử f là luồng, còn (S, T) là lát cắt, khi đó $\text{val}(f) \leq \text{caps}(S, T)$

Hệ quả: giả sử f là luồng, còn (S, T) là lát cắt. Nếu $\text{val}(f) = \text{cap}(S, T)$ thì f là luồng cực đại, còn (S, T) là lát cắt hẹp nhất.

2.2. Định lý Ford – Fulkerson và thuật toán giải bài toán luồng cực đại

ĐL Ford - Fulkerson: trong mạng bất kỳ, giá trị luồng cực đại luôn bằng khả năng thông qua của lát cắt nhỏ nhất.

Thuật toán Ford - Fulkerson:

Đầu vào: cho đồ thị $G = (V, E)$ với các khả năng thông qua $c(u, v)$

Đầu ra: luồng f từ s đến t sao cho f có giá trị cực đại

Sau mỗi bước của thuật toán, các điều kiện sau cần được duy trì

- $f(u, v) \leq c(u, v)$. Luồng từ u tới v không vượt quá khả năng thông qua
- $f(u, v) = -f(v, u)$. Ta duy trì cân bằng luồng
- $\sum_v f(u, v) = 0$ cho tất cả các nút ngoại trừ s và t . Lượng đi vào vào một nút bằng lượng đi ra khỏi nút đó.

Điều này có nghĩa là một luồng đi qua mạng là một luồng hợp lệ sau mỗi bước lặp của thuật toán. Ta định nghĩa mạng thặng dư $G_f(V, E_f)$ là mạng với sức chứa $c_f(u, v) = c(u, v) - f(u, v)$ và luồng bằng không.

Begin

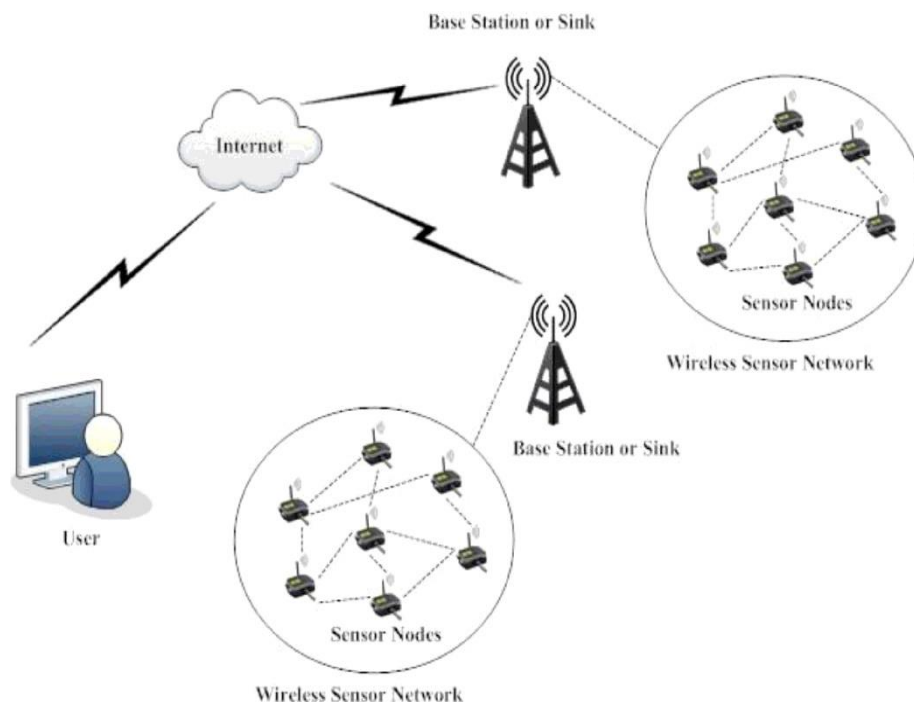
1. $f(u, v) \leftarrow 0 \forall u, v$
2. Trong khi còn có một đường đi từ s đến t trong G
3. Tìm một đường đi u_1, u_2, \dots, u_k với $u_1 = s, u_k = t$ sao cho $c_f(u_i, u_{i+1}) > 0$
4. Tìm $m = \min(c_f(u_i, u_{i+1}))$
5. $f(u_i, u_{i+1}) \leftarrow f(u_i, u_{i+1}) + m$
6. $f(u_{i+1}, u_i) \leftarrow f(u_{i+1}, u_i) - m$

End.

Đường đi trong $G_f(V, E_f)$ có thể tìm được bằng thuật toán tìm kiếm theo chiều rộng (breadth first search - BFS) hoặc tìm kiếm theo chiều sâu (depth first search – DFS).

Chương II. Bài toán tối ưu thời gian sống của mạng cảm biến không dây

I. Tổng quan về mạng cảm biến không dây



Mạng cảm biến không dây (Wireless Sensor Network) bao gồm một tập hợp các thiết bị cảm biến sử dụng các liên kết không dây (vô tuyến, hồng ngoại hoặc quang học) để phối hợp thực hiện nhiệm vụ thu thập thông tin dữ liệu phân tán với quy mô lớn trong nhiều điều kiện khác nhau. Các thiết bị cảm biến thường đơn giản, nhỏ gọn, giá thành rẻ và phân bố rộng khắp trong khu vực cảm biến, sử dụng những nguồn năng lượng giới hạn như pin hay ắc quy. Dữ liệu sau khi thu thập được sẽ được gửi đến các trạm cơ sở (Base Stations) trực tiếp hay gián tiếp thông qua một điểm thu phát (Sinks) và môi trường mạng công cộng như Internet hay vệ tinh.

Các nút cảm biến không dây có thể được triển khai cho các mục đích chuyên dụng như điều khiển giám sát và an ninh, kiểm tra môi trường, tạo ra không gian sống thông minh, khảo sát đánh giá chính xác trong nông nghiệp, trong lĩnh vực y tế,... Lợi thế chủ yếu của chúng là có thể triển khai trong rất nhiều môi trường khác nhau, kể cả trong các môi trường nguy hiểm (nhiễm độc, phóng xạ,...) mà mạng có dây truyền thống không thể thực hiện được.

Các thiết bị cảm biến không dây liên kết thành một mạng đã tạo ra nhiều khả năng mới cho con người. Các đầu đo với bộ vi xử lý và các thiết bị vô tuyến rất nhỏ gọn tạo nên một thiết bị cảm biến không dây có kích thước rất nhỏ, tiết kiệm về không gian. Chúng có thể hoạt động trong môi trường dày đặc với khả năng xử lý tốc độ cao. Ngày nay, các mạng cảm biến không dây được ứng dụng trong nhiều lĩnh vực như nghiên cứu vi sinh vật biển, giám sát việc chuyển chở các chất gây ô nhiễm, kiểm tra giám sát

hệ sinh thái và môi trường sinh vật phức tạp, điều khiển giám sát trong công nghiệp và trong lĩnh vực quân sự, an ninh quốc phòng hay các ứng dụng trong đời sống hàng ngày.

Cấu trúc của mạng cảm biến không dây

Một mạng cảm biến không dây bao gồm lượng lớn các nút được triển khai dày đặc bên trong hoặc ở rất gần đối tượng cần thăm dò, thu thập thông tin dữ liệu. Vị trí các cảm biến không cần định trước vì vậy nó cho phép triển khai ngẫu nhiên trong các vùng không thể tiếp cận hoặc các khu vực nguy hiểm.

Trong mạng cảm biến không dây, các nút cảm biến được triển khai trong một môi trường cảm biến. Mỗi nút cảm biến được phát tán trong mạng có khả năng thu thập thông số liệu, định tuyến, số liệu về bộ thu nhận (sink) để chuyển tới người dùng và định tuyến các bản tin mang theo yêu cầu từ nút sink đến các nút cảm biến

Mỗi nút cảm biến gồm bốn thành phần cơ bản: bộ cảm biến, bộ xử lý, bộ thu phát sóng không dây và nguồn điện. Tùy theo ứng dụng cụ thể, nút cảm biến còn có thể có các thành phần bổ sung như hệ thống tìm vị trí, bộ sinh năng lượng và thiết bị di động. Bộ cảm biến thường gồm hai đơn vị thành phần là đầu đo cảm biến (sensor) và bộ chuyển đổi tương tự/số (ADC). Bộ xử lý thường kết hợp với một bộ nhớ nhỏ, phân tích thông tin cảm biến và quản lý các thủ tục cộng tác với các nút khác. Bộ thu phát đảm bảo thông tin giữa nút cảm biến và mạng bằng kết nối không dây. Một thành phần quan trọng của nút cảm biến là bộ nguồn, có thể là pin hoặc ắc quy, cung cấp năng lượng cho cảm biến và không thay thế được nên nguồn năng lượng của nút thường là giới hạn. Bộ nguồn có thể được hỗ trợ bởi các thiết bị sinh điện, ví dụ như các tấm pin mặt trời nhỏ.

Một số đặc điểm của mạng cảm biến không dây

Ưu điểm

- Kích thước vật lý nhỏ gọn
- Hoạt động đồng thời với độ tập trung cao
- Dễ dàng triển khai, mở rộng
- Chi phí các thiết bị cảm biến không cao
- Tính đa dạng trong thiết kế và sử dụng
- Hoạt động tin cậy

Nhược điểm

- Khả năng liên kết vật lý và phân cấp điều khiển hạn chế
- Thiết bị cảm biến có khả năng tính toán, lưu trữ và nguồn năng lượng hạn chế
- Dư thừa dữ liệu
- Truyền thông không dây dẫn tới một số vấn đề như độ ổn định, mất mát thông tin, bảo mật

II. Bài toán tối ưu thời gian sống của mạng cảm biến không dây

Với sự phát triển mạnh mẽ của mình, mạng cảm biến không dây được ứng dụng vào rất nhiều lĩnh vực khác nhau như nông nghiệp, quân sự, y tế,... Và để sử dụng một cách hiệu quả, rất nhiều vấn đề trong mạng cảm biến không dây được quan tâm nghiên cứu như hiệu suất, độ bao phủ, cấu trúc mạng, phương thức định tuyến,... trong đó có một trong các yếu tố quan trọng bậc nhất đó là vấn đề năng lượng hay nói cách khác là thời gian sống (tuổi thọ) của mạng cảm biến, do một mạng cảm biến chỉ thực thi đầy đủ chức năng của mình khi nó còn “sống”. Cho đến nay có nhiều cách định nghĩa về tuổi thọ của mạng cảm biến không dây khác nhau do nhiều tác giả đưa ra. Ta sẽ đề cập một số cách định nghĩa phổ biến sau.

1. Tuổi thọ của mạng dựa vào số nút còn hoạt động

Cách định nghĩa phổ biến trong các tài liệu đó là n-of-n, trong đó tuổi thọ của mạng T_n^n được định nghĩa bằng tuổi thọ của nút cảm biến có thời gian sống ít nhất.

$$T_n^n = \min_{v \in V} T_v$$

Trong đó T_v là thời gian sống của nút v

T_n^n là cách định nghĩa đơn giản và thuận tiện cho tính toán, các thuật toán dùng trong mạng không cần quan tâm đến bất kì sự thay đổi nào của kiểu hình mạng. Tuy nhiên trong phần lớn trường hợp, cách định nghĩa này có ý nghĩa lớn trong các bài toán lý thuyết hơn là các ứng dụng thực tế của mạng cảm biến. Ví dụ, xét một nút cảm biến có nhiều nút hàng xóm với công năng cảm biến tương đương, nhiều mạng cảm biến hoàn toàn có thể chấp nhận được việc nút này dừng hoạt động và sử dụng các nút hàng xóm, nhưng T_n^n không xét đến các trường hợp như vậy và cho rằng mạng đã “chết”. Do đó, ta nên sử dụng cách định nghĩa này khi các nút trong mạng có độ quan trọng xấp xỉ nhau.

Ngoài ra còn có một số biến thể khác của T_n^n . Tuổi thọ của mạng được định nghĩa là thời gian nhỏ nhất của một nút từ khi bắt đầu hoạt động cho đến khi năng lượng của nút xuống thấp hơn một mức β cho trước, hoặc cũng có thể được định nghĩa mạng còn sống khi ít nhất k nút trong mạng còn hoạt động (k-of-n - T_n^k).

2. Tuổi thọ của mạng dựa vào độ bao phủ của các nút cảm biến

Trong khi đánh giá tuổi thọ của mạng cảm biến dựa vào số nút cảm biến còn hoạt động là một cách định nghĩa khá tổng quát thì việc xét đến các đặc điểm của mạng cảm biến, cụ thể như đánh giá tuổi thọ của mạng dựa vào độ bao phủ của các nút cảm biến với môi trường đang khảo sát có vẻ như là một cách tự nhiên hơn để định nghĩa tuổi thọ của mạng.

Các mô hình bao phủ trong mạng cảm biến có thể chia thành ba loại chính:

- Bao phủ diện tích: mỗi điểm trong không gian khảo sát đều phải nằm trong vùng cảm biến

- Bao phủ đối tượng: một số điểm định trước trong không gian phải nằm trong vùng cảm biến
- Bao phủ barrier: mạng cảm biến cần có chức năng phát hiện đối tượng di chuyển vào hoặc băng qua khu vực được giám sát

Có hai cách để mô tả độ bao phủ của một mạng cảm biến:

- Một số phần trăm α của không gian được bao phủ bởi ít nhất một cảm biến, thường được gọi là α -coverage
- Mỗi điểm trong không gian cần được bao phủ bởi ít nhất k sensor, thường được gọi là k -coverage

Nhiều nghiên cứu định nghĩa tuổi thọ của mạng dựa trên các biến thể của cách định nghĩa trên, trong đó phổ biến nhất là 1-coverage, tức là tuổi thọ của mạng được tính từ lúc bắt đầu triển khai cho đến khi tất cả các điểm cần thiết nằm trong vùng cảm biến của ít nhất một nút cảm biến.

Độ bao phủ thường là một yếu tố cực kì quan trọng để đánh giá hiệu quả của mạng cảm biến, tuy nhiên đánh giá tuổi thọ của mạng chỉ dựa trên độ bao phủ thì không hợp lí, do không thể đảm bảo dữ liệu thu thập được chắc chắn truyền về nút sink.

3. Tuổi thọ của mạng dựa trên kết nối

Baydere et al. [2005] và Yu et al. [2001] định nghĩa tuổi thọ của mạng dựa trên tổng số lượng gói tin có thể truyền tới nút sink. Mặc dù số lượng này có thể đóng vai trò là chỉ số cho sự bền bỉ của mạng, nhưng nó phụ thuộc rất nhiều vào các thuật toán cụ thể được áp dụng trong mạng. Giả sử ta áp dụng thuật toán tập hợp thông tin trong mạng, như vậy số gói tin truyền về nút sink sẽ giảm trong khi lượng thông tin nhận được vẫn là tương đương. Do đó khả năng áp dụng định nghĩa này trong việc so sánh tuổi thọ của các thiết lập mạng khác nhau sẽ bị hạn chế.

Một nhược điểm khác của cách định nghĩa này là số các gói tin được truyền về không phản ánh được mạng cảm biến hoạt động bình thường trong bao lâu (tính theo đơn vị thời gian).

Trong [Giridhar and Kumar 2005] định nghĩa tuổi thọ của mạng dựa theo số lần thu thập dữ liệu từ môi trường thành công, trong đó với mỗi lần thu thập dữ liệu thì không có nút nào hết năng lượng. Cách định nghĩa này có thể quy dẫn về T_n^n đã nêu ở trên, chỉ khác đó là tuổi thọ không được tính theo đơn vị thời gian mà là số lần thu thập dữ liệu.

4. Tuổi thọ của mạng dựa trên độ bao phủ của các nút cảm biến và kết nối

Do một số hạn chế của việc đánh giá tuổi thọ mạng dựa trên độ bao phủ hoặc dựa vào kết nối, một số tác giả đã kết hợp cả hai tiêu chí này lại để đánh giá tuổi thọ của mạng cảm biến. Trong [Wang et al. 2003; Xing et al. 2005], tuổi thọ của mạng định nghĩa bằng thời gian từ khi mạng bắt đầu hoạt động cho tới khi độ bao phủ hoặc kết nối hạ xuống thấp hơn ngưỡng cho trước, trong đó độ bao phủ sử dụng α -

coverage và kết nối sử dụng tỉ lệ số gói nhận được tại sink so với tổng số gói được gửi từ các nút cảm biến.

5. Tuổi thọ của mạng dựa trên chất lượng ứng dụng

Một số nghiên cứu định nghĩa tuổi thọ của mạng cảm biến dựa trên chất lượng ứng dụng, cách định nghĩa này gần với thực tế nhất do mọi thiết kế, kiến trúc của mạng cảm biến đều phụ thuộc vào ứng dụng cụ thể mà mạng thực hiện. Ví dụ trong Kumar et al. [2005] đã nêu “tuổi thọ của mạng cảm biến là khoảng thời gian mà nó thực thi các yêu cầu của ứng dụng một cách bình thường”. Tuy nhiên, cách định nghĩa này khó có thể áp dụng trong các nghiên cứu lí thuyết do nó quá trừu tượng và gần như không thể mô hình hóa được.

III. Các nghiên cứu liên quan

Năm 2017, Bo Yuan và cộng sự đưa ra mô hình bài toán với mạng cảm biến không dây ngầm. Nội dung như sau: xét một mạng cảm biến không dây ngầm, trong đó có một số sensors được chôn dưới lòng đất và trên mặt đất có các vị trí khả thi để đặt nút relays, ngoài ra có các base stations, thông tin được thu nhận từ sensors được chuyển tiếp qua relays để gửi đến base stations, cho biết trước số lượng các relays được triển khai, mục tiêu đặt ra là tìm vị trí thích hợp để triển khai các relays này sao cho tuổi thọ của mạng được tối ưu hóa.

Trong mô hình này, tác giả đưa ra giả thiết trong địa hình đã cho, bất kì cặp sensor-relay nào cũng đều có thể kết nối được với nhau, số relays được triển khai là cố định cho trước, do đó để tối ưu thời gian sống của mạng, tác giả sử dụng điều kiện cân bằng tải, tức là mỗi relay sẽ kết nối đến số sensors bằng nhau.

Bài toán đã được chứng minh là NP-Khó, tác giả đưa ra cách giải quyết theo hai pha

- Pha một: giải quyết bài toán, tạo các kết nối mà không xét đến điều kiện cân bằng tải
- Pha hai: sắp xếp lại các kết nối để duy trì điều kiện cân bằng tải

Với pha một, tác giả đưa ra hai đề xuất heuristics:

- Đề xuất một: lần lượt triển chọn các vị trí để triển khai relay sao cho năng lượng tiêu hao lớn nhất trong mạng là nhỏ nhất, cho đến khi đủ số relay cần triển khai
- Đề xuất hai: kết nối một sensor với relay gần nó nhất, sau đó nếu số relay kết nối nhiều hơn số relay được triển khai, loại bỏ dần bằng cách tính toán sao cho năng lượng tiêu hao lớn nhất trong mạng là nhỏ nhất

Với pha hai, tác giả đưa ra ba đề xuất heuristics:

- Đề xuất một: với các relay đã quá tải, chuyển dần các sensor của relay này cho các relay thiếu tải

- Đề xuất hai: với các relay thiếu tải, lấy lần lượt các sensor ở relay đã quá tải
- Đề xuất ba: tính toán một điểm là trung tâm của mạng, sau đó tính khoảng cách từ các relay đến trung tâm điểm này, xét từ relay xa nhất tới relay gần nhất, nếu relay đó đã quá tải thì sử dụng đề xuất một cho relay đó, nếu relay thiếu tải thì sử dụng đề xuất hai.

Dễ nhận thấy một điểm chưa khả thi trong mô hình này, đó là mỗi cặp relay-sensor trong mạng đều có thể kết nối được với nhau, nếu một cặp relay-sensor nằm ở hai góc khác nhau ở địa hình thì việc có thể kết nối được với nhau là khó xảy ra trong trường hợp địa hình lớn. Trên cơ sở đó đồ án đề xuất một mô hình mới như sau.

IV. Mô hình bài toán mà đồ án đang giải quyết

Nội dung bài toán: cho một mạng cảm biến gồm n cảm biến được triển khai ngầm trong lòng đất, trên mặt đất có một tập các vị trí khả thi để đặt nút chuyển tiếp. Nhiệm vụ của các nút chuyển tiếp là nhận dữ liệu từ các nút cảm biến sau đó gửi đến trạm cơ sở ở trung tâm của mạng. Bài toán đặt ra là cần chọn một tập các vị trí trong các vị trí khả thi để triển khai nút chuyển tiếp nhằm kéo dài thời gian sống của mạng và số các nút chuyển tiếp cần dùng là ít nhất có thể.

Tuổi thọ của mạng sử dụng định nghĩa T_n^n đã nêu ở trên. Với cách định nghĩa này, mạng dừng hoạt động khi có một nút hết năng lượng, có nhiều cách tiếp cận để giải quyết bài toán với định nghĩa tuổi thọ T_n^n , trong mô hình này ta sử dụng cách tiếp cận tối thiểu hóa năng lượng tiêu hao cực đại của một nút trong mạng.

Với mục tiêu tối ưu hai mục tiêu cùng lúc là thời gian sống tối đa và số nút chuyển tiếp cần dùng tối thiểu, nếu giá trị của hàm mục tiêu là một bộ gồm hai giá trị số thực, ta khó có thể so sánh một tập các lời giải gần tương đương nhau để đánh giá lời giải nào tốt hơn, do đó ta sẽ đưa hàm mục tiêu về dạng số thực nhằm thuận tiện cho việc đánh giá.

Đầu vào:

- Không gian 3 chiều cần khảo sát với chiều dài và chiều rộng lần lượt là W và H , chiều cao (sâu) vô hạn
- $S = \{s_1, s_2, \dots, s_n\}$: tập các nút cảm biến được triển khai, mỗi nút cảm biến s_i có các thuộc tính
 - (x_i, y_i) : tọa độ trên mặt Oxy
 - h_i : độ sâu so với mặt đất của cảm biến
 - r : bán kính truyền thông
 - l : số bit mà nút cảm biến gửi tới trạm cơ sở
- $F = \{f_1, f_2, \dots, f_m\}$: tập các vị trí khả thi để triển khai các nút chuyển tiếp, f_i gồm các thuộc tính
 - (x_i, y_i) : tọa độ trên mặt Oxy
 - h_i : độ cao so với mặt đất

- r : bán kính truyền thông
- $D = \{d_{11}, d_{12}, \dots, d_{nm}\}$: ma trận khoảng cách
 - d_{ij} : khoảng cách từ sensor s_i đến relay f_j
- $C = \{c_{11}, c_{12}, \dots, c_{nm}\}$: ma trận kết nối
 - $c_{ij} = \begin{cases} 1 & \text{nếu } d_{ij} \leq r_i + r_j \\ 0 & \text{nếu ngược lại} \end{cases}$
 - Ý nghĩa: sensor s_i có thể kết nối với relay f_j nếu khoảng cách giữa hai nút không vượt quá tổng bán kính truyền thông của chúng
- Năng lượng tiêu hao khi truyền 1 bits dữ liệu từ sensor s_i đến relay triển khai tại f_j
 - $E_{tij} = 1 * (E_{TX} + e_{fs} * d_{ij}^2)$
- Năng lượng tiêu hao của relay triển khai tại f_j nhận dữ liệu từ x_j sensors, tổng hợp và gửi tới trạm cơ sở
 - $E_{rj} = 1 * (x_j * E_{RX} + x_j * E_{DA} + e_{mp} * d_{jtoBS}^4)$
- E_{max} : năng lượng tiêu thụ tối đa mà một nút trong mạng có thể đạt đến

Đầu ra:

- $z = (z_1, z_2, \dots, z_m)$: vector quyết định
 - $z_j = \begin{cases} 1 & \text{nếu có relay triển khai tại } f_j \\ 0 & \text{nếu ngược lại} \end{cases}$
- $A = \{a_{11}, a_{12}, \dots, a_{nm}\}$: ma trận quyết định
 - $a_{ij} = \begin{cases} 1 & \text{nếu } s_i \text{ kết nối tới } f_j \\ 0 & \text{nếu ngược lại} \end{cases}$

Ràng buộc:

- $\sum_{j=0}^m a_{ij} = 1 \forall i = 0, 1, \dots, m$: mỗi sensor chỉ kết nối tới một relay
- $z_j = a_{1j} \wedge a_{2j} \wedge \dots \wedge a_{nj}$

Hàm mục tiêu:

$$\frac{\alpha}{m} * \sum_{j=1}^m z_j + \frac{1-\alpha}{E_{max}} E_x \rightarrow \min$$

Trong đó:

- E_x là năng lượng tiêu hao lớn nhất của một nút trong lời giải
- $0 \leq \alpha \leq 1$: trọng số đánh giá độ quan trọng trong hàm mục tiêu của số nút chuyển tiếp được sử dụng.

Ý tưởng giải quyết bài toán: khi sử dụng hàm mục tiêu đã nêu trong công thức (), giá trị của hàm mục tiêu phụ thuộc vào năng lượng tiêu hao và số relay sử dụng, do đó trong bài toán này ta cần giải quyết hai vấn đề, đó là vị trí nào được chọn để triển khai relay và kết nối giữa các relay và sensor. Đồ án đưa ra phương pháp giải quyết theo hai pha

- Pha một: xác định các vị trí được chọn để triển khai relay
- Pha hai: đặt kết nối giữa các relays và sensors

Chương III: Giải thuật di truyền đề xuất giải bài toán tối ưu thời gian sống của mạng cảm biến không dây

Với việc sử dụng giải thuật di truyền để tối ưu hóa hàm mục tiêu, ta chọn một cá thể là tập các vị trí được chọn để triển khai relay, việc thiết lập kết nối giữa relays và sensors sẽ được thực hiện với từng cá thể, cuối cùng sử dụng hàm thích nghi để đánh giá cá thể này.

I. Mã hóa cá thể

Dựa theo biến quyết định z trong mô hình bài toán, một cách tự nhiên ta chọn mã hóa nhị phân để biểu diễn lời giải.

Nhắc lại: $z = (z_1, z_2, \dots, z_m)$ trong đó $z_j = 1$ nếu vị trí j được chọn để triển khai relay

Ví dụ: một cá thể với 10 vị trí khả thi đặt relay

1	0	1	1	0	0	0	0	1	1
1	2	3	4	5	6	7	8	9	10

II. Xác định các vị trí đặt relay

1. Khởi tạo quần thể

Khởi tạo ngẫu nhiên: đây là cách khởi tạo rất đơn giản, trước tiên ta khởi tạo ngẫu nhiên một số nguyên Y trong đoạn $[1, m]$, tiếp đó chọn ngẫu nhiên Y vị trí trong cá thể để đặt relay

Thuật toán: RandomInitialization (m, r, W, H)

Đầu vào:

Số vị trí khả thi: m

Bán kính truyền thông: r

Kích thước miền không gian: W, H

Đầu ra:

Cá thể sinh ngẫu nhiên $z_{\text{rand}} = (z_1, z_2, \dots, z_m)$ và các kết nối tới sensors

Begin

1. $z_{\text{rand}} \leftarrow \emptyset$

2. $t \leftarrow 0$

3. $Y \leftarrow \text{rand}(1, m)$


```

4. while  $t < Y$ 
5. for  $j = 1$  to  $m$ 
6.      $cx = \text{random}(0, 1)$ 
7.     if  $cx \leq Y/m$  and  $z_j = 0$ 
8.          $t \leftarrow t + 1$ 
9.          $z_j \leftarrow 1$ 
10.    end if
11. end for

```

End

2. Lai ghép

Đề xuất sử dụng phép lai ghép một điểm cắt cho hai cá thể z_1, z_2 được chọn ngẫu nhiên từ quần thể hiện tại. Trong phép lai ghép một điểm cắt, ta chọn tùy ý một điểm làm điểm cắt, sau đó ghép các đoạn gen của cha mẹ lại để tạo ra hai cá thể con

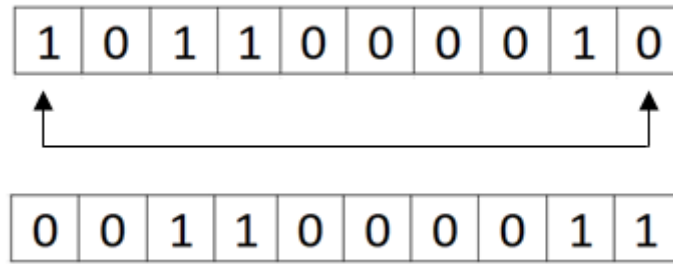
Minh họa cho phép lai ghép như sau

• DAD	1	0	1	1	0	0	0	0	1	0
• MOM	0	0	1	0	1	1	1	0	1	1
⇒ SON	1	0	1	1	1	1	1	0	1	1
⇒ DAUGHTER	0	0	1	0	0	0	0	0	1	0

Sau khi tạo ra hai cá thể con, ta sử dụng thủ tục MakeConnect đã nêu ở trên với hai cá thể này để tạo các kết nối giữa sensor và relay

2. Đột biến

Cá thể con sau khi sinh ra có thể xảy ra đột biến với xác suất đột biến q . Phép đột biến được thực hiện như sau: chọn ngẫu nhiên hai gen trên cá thể trong đó một gen mang giá trị 1 và một gen mang giá trị 0, sau đó đổi chỗ chúng cho nhau



III. Xác định kết nối giữa relays và sensors

Dựa vào các toán tử lai ghép và đột biến của giải thuật di truyền, quần thể có được độ đa dạng tương đối và khả năng lớn sẽ hội tụ về hướng lời giả tối ưu, do đó sau khi cố định các vị trí đặt relay, ta sử dụng một thuật toán heuristic đơn giản để tạo các kết nối giữa các sensors – relays nhằm làm tăng tốc độ giải thuật.

Thuật toán: Connect (S, R, C, Et, Er)

Đầu vào:

$S = \{s_1, s_2, \dots, s_n\}$: tập các sensors

$R = \{r_1, r_2, \dots, r_k\}$: tập các relay

C: ma trận kết nối

$E_{t_{ij}}, E_{r_j}$: năng lượng tiêu hao

Đầu ra:

Các kết nối giữa sensors và relays

Begin

1. **for** s_i **in** (s_1, s_2, \dots, s_n)
2. $\min_max \leftarrow INF$
3. $\text{sel_id} \leftarrow 0$
4. **for** r_j **in** (r_1, r_2, \dots, r_k)
5. $\text{loss_1} \leftarrow E_{t_{i, \text{sel_id}}}$
6. $\text{loss_2} \leftarrow E_{r_j}$
7. $\text{local_max} \leftarrow \max(\text{loss_1}, \text{loss_2})$
8. **if** $\min_max < \text{local_max}$
9. $\min_max \leftarrow \text{local_max}$
10. $\text{sel_id} \leftarrow j$

11. **end if**
12. **end for**
13. **Assign** s_i to r_{sel_id}
14. **end for**

End.

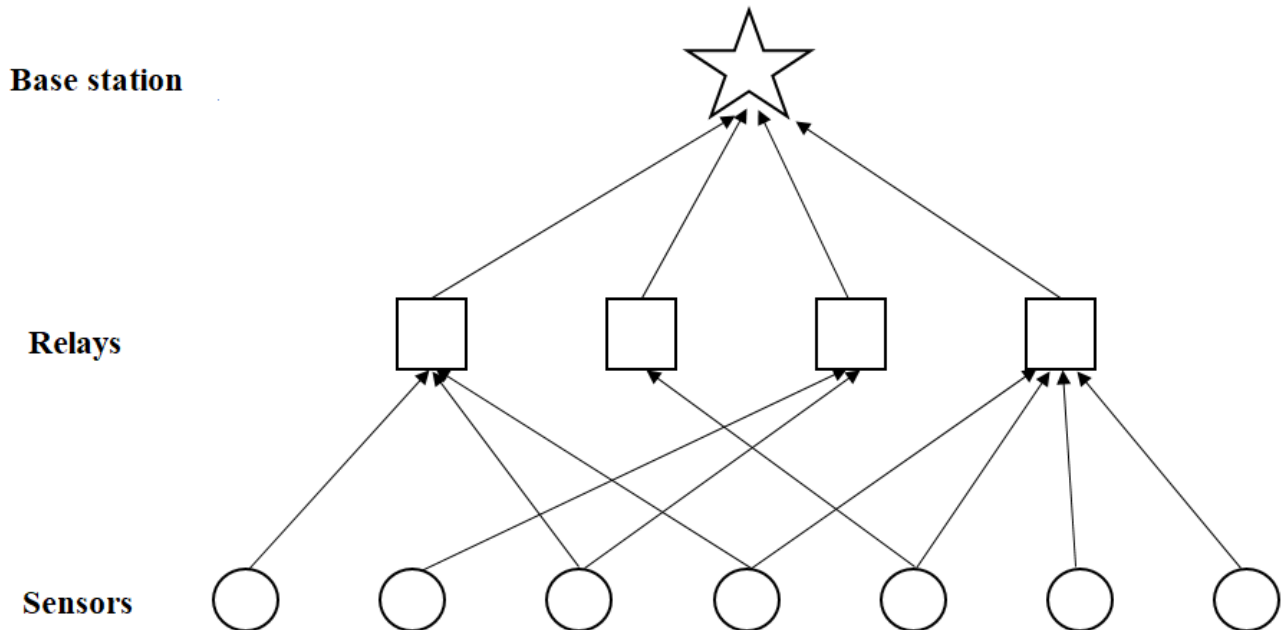
IV. Hàm thích nghi và chọn lọc cá thể

Trong đề xuất này, ta sử dụng hàm mục tiêu của bài toán như là hàm thích nghi của mỗi cá thể.

Sau khi tiến hành lai ghép, đột biến, các cá thể sẽ được tính độ thích nghi và sắp xếp theo giá trị thích nghi này. Các cá thể tốt nhất sẽ được giữ lại để tiếp tục quá trình tiến hóa còn các cá thể tồi hơn sẽ bị loại bỏ. Đây là phương pháp chọn lọc giữ lại những cá thể ưu tú.

Chương IV: Phương pháp tìm kiếm cục bộ giải bài toán tối ưu thời gian sống của mạng cảm biến không dây

Quá trình truyền nhận dữ liệu của mô hình được thể hiện đơn giản như sau:



Nhìn vào hình vẽ trên, dễ dàng liên tưởng đến bài toán luồng trong mạng khi coi môi trường đặt cảm biến là một điểm phát giả và điểm thu là base station, khả năng thông qua của các cung source – sensor là 1, sensor-relay là 1, của các cung relay-base station là số sensor mà relay đó có thể kết nối.

Dựa vào việc giải quyết bài toán luồng trong mạng với chi phí cực tiểu, ta có thể xác định được kết nối giữa các cặp relay-sensor. Tuy nhiên với phương pháp này, trước đó ta cần biết được các vị trí đặt relay và mỗi relay kết nối với bao nhiêu sensor để có được khả năng thông qua của cung.

Đề xuất:

- Sử dụng tìm kiếm cục bộ để xác định vị trí đặt relay và số sensor relay đó kết nối tới.
- Lượng giá lời giải bằng thuật toán luồng cực đại trong mạng với chi phí

I. Tìm kiếm cục bộ

Mã hóa lời giải

Lời giải của bài toán được mã hóa bởi vector: $k = (k_1, k_2, \dots, k_m)$

Trong đó:

- k_j là số sensor mà relay j kết nối, $1 \leq j \leq m$
- $k_j \leq q_j \forall 1 \leq j \leq m$

○ $q_j = \sum_{i=1}^n c_{ij} \forall 1 \leq i \leq n$: q_j là số sensor tối đa relay j có thể kết nối, C là ma trận kết nối đã nêu trên

- $\sum_{j=1}^m k_j = n$: tổng số sensor mà các relay kết nối tới đúng bằng số sensor trong mạng.

Toán tử di chuyển

Đề xuất sử dụng ba toán tử di chuyển

- Swap: chọn 2 chỉ số i, j thỏa mãn $1 \leq i, j \leq m, k_i \leq q_i, k_j \leq q_j$, trao đổi giá trị của k_i, k_j
- Transfer: chọn 2 chỉ số i, j thỏa mãn $1 \leq i, j \leq m, k_i + k_j \leq q_i$, chuyển tất cả sensor từ relay j sang relay i
- Up – down: chọn 2 chỉ số i, j thỏa mãn $1 \leq i, j \leq m, k_i > 0, k_j < q_j$, chuyển 1 sensor từ relay i sang relay j

II. Lượng giá lời giải

Sau khi cố định vị trí đặt các relay và số sensors mỗi relay kết nối tới, dựa vào công thức tiêu hao năng lượng () và (), ta tính được năng lượng tiêu hao của mỗi kết nối trong mạng.

Do đang tiếp cận theo hướng cực tiểu hóa năng lượng tiêu hao cực đại trong mạng, đề án đưa ra cách giải quyết bài toán luồng trong mạng với chi phí theo các bước như sau:

- Sắp xếp tập các kết nối $E = \{e_1, e_2, \dots, e_k\}$ trong mạng theo thứ tự không giảm bằng sắp xếp trộn (merge sort), tập các kết nối đã được sắp xếp $E' = \{e'_1, e'_2, \dots, e'_k\}$
- Tìm ra kết nối có chỉ số nhỏ nhất e_i ($1 \leq i \leq k$) sao cho với các kết nối e_1, e_2, \dots, e_i ta giải được bài toán luồng cực đại trong mạng, việc tìm ra kết nối này dựa trên ý tưởng của tìm kiếm nhị phân.

Sắp xếp trộn là thuật toán phổ biến và đã được trình bày trong nhiều tài liệu do đó không trình bày lại ở đây, thuật toán được cài đặt với độ phức tạp $O(k \log k)$

Thuật toán: FindConnect (E', i, j)

Đầu vào:

E' : tập các kết nối đã được sắp xếp

2 chỉ số i và j : chỉ số bắt đầu và chỉ số kết thúc

Năng lượng tiêu hao của các kết nối trong mạng

Đầu ra:

Các kết nối được chọn

Begin

1. $start \leftarrow i$

2. $end \leftarrow j$

3. $mid \leftarrow \lfloor (start + end) / 2 \rfloor$
4. $l_1 \leftarrow \mathbf{MaxFlow} (e'_1, \dots, e'_{start})$
5. $l_2 \leftarrow \mathbf{MaxFlow} (e'_1, \dots, e'_{mid})$
6. $l_3 \leftarrow \mathbf{MaxFlow} (e'_1, \dots, e'_{end})$
7. **if** l_1 **is True**
8. **return** $\{e'_1, \dots, e'_{start}\}$
9. **end if**
10. **if** l_3 **is False**
11. **return False**
12. **end if**
13. **if** l_2 **is False**
14. **FindConnect** (mid, end)
15. **else**
16. **FindConnect** ($start, mid$)
17. **end if**
18. **return False**

End.

Trong đó **MaxFlow** là thuật toán Ford – Fulkerson đã được trình bày trong chương I, mục IV.2.2.

Tài liệu tham khảo

- [1]. Nguyễn Đức Nghĩa, Nguyễn Tô thành, *Toán rời rạc*, NXB Đại học Quốc gia Hà Nội, 2009.
- [2]. Nguyễn Đình Thúc, *Lập trình tiến hóa*, NXB Giáo dục, 2009.
- [3]. Bo Yuan, Huanhuan Chen, Xin Yao, *Optimal Relay Placement for Lifetime Maximization in Wireless Underground Sensor Networks*, Information Sciences, Volume 418-419, December 2017, Pages 463-479.
- [4]. Edgar H.Callaway Jr., *Wireless Sensor Networks: Architectures and Protocols*, A CRC Press Company, 2004.
- [5]. Anna Ha'c, *Wireless Sensor Network Designs*, University of Hawaii at Manoa, Honolulu, USA, John Wiley & Sons Ltd, 2003.
- [6]. Isabel Dietrich, Falko Dressler, *On the Lifetime of Wireless Sensor Networks*,
- [7]. Boyd, Stephen P.; Vandenberghe, Lieven (2004).
- [8]. Lê Minh Hoàng, *Giải thuật & Lập trình*, NXB Đại học Sư phạm Hà Nội, 2005.
- [9]. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0262032937. Section 26.2: The Ford-Fulkerson method, pp. 651–664.