

TOÁN RỜI RẠC

Bài toán liệt kê, tối ưu

NGUYỄN HẢI TRIỀU¹

¹Bộ môn Kỹ thuật phần mềm,
Khoa Công nghệ thông tin, Trường ĐH Nha Trang

NhaTrang, February 2022

Tổng quan

- 1 Bài toán liệt kê
- 2 Bài toán tối ưu
- 3 Sinh viên viết báo cáo

Có nhiều bài toán liệt kê mà chúng ta không thể xây dựng được thuật toán sinh. Trong trường hợp như vậy, **thuật toán quay lui (backtracking algorithm)** là giải pháp **thường** được sử dụng.

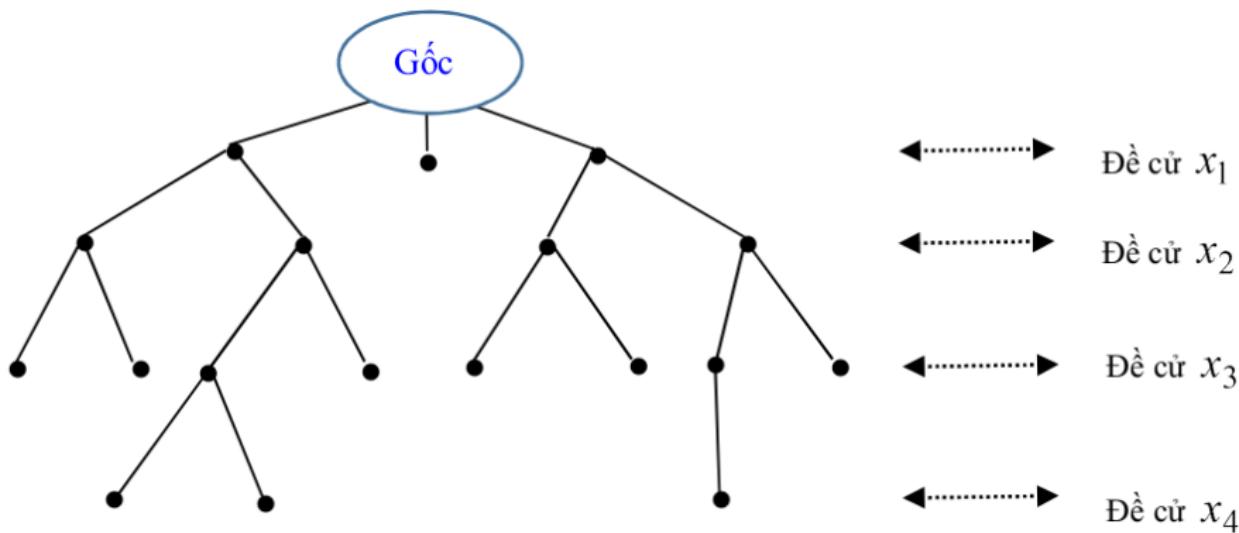
Nội dung chính của thuật toán quay lui là xây dựng tuần tự từng cấu hình. Mỗi cấu hình được xây dựng bằng cách thử từng thành phần của nó với tất cả các khả năng có thể xảy ra. Giả thiết rằng, mỗi cấu hình cần xác định là (x_1, x_2, \dots, x_n) .

- Giả sử ta đã xây dựng được k thành phần đầu tiên của nó là (x_1, x_2, \dots, x_k) . Xác định thành phần x_{k+1} bằng cách duyệt tất cả các khả năng còn lại có thể để đề cử cho nó.

Với mỗi khả năng j (thuộc tập đề cử $1, n_{k+1}$) được đề cử cho x_{k+1} , cần kiểm tra j có được chấp nhận không, có hai trường hợp xảy ra:

- Nếu chấp nhận j thì xác định $x_{k+1} := j$. Sau đó, nếu $k = n$ thì ta nhận được một cấu hình, ngược lại ta tiến hành việc xác định x_{k+2}, \dots
- Ngược lại, nếu thử tất cả các khả năng mà không có khả năng nào được chấp nhận, thì quay lui lại bước trước để đề cử lại giá trị khác cho x_k .

Quá trình tìm kiếm lời giải các cấu hình cần liệt kê theo thuật toán quay lui được mô tả bởi cây tìm kiếm lời giải như trên hình 1. Nhánh của cây có đủ n thành phần sẽ là cấu hình cần tìm.



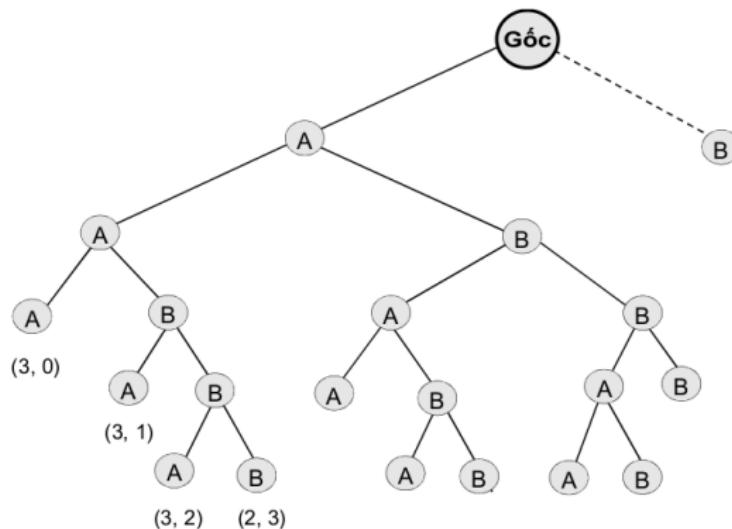
Hình 1: Cây liệt kê lời giải của thuật toán quay lui: nút đầu tiên là gốc xem như nút cấp 0, các nút xuất phát từ gốc được gọi là nút cấp 1, là các giá trị được đề cử cho thành phần x_1 , các nút xuất phát từ các nút cấp 1 được gọi là nút cấp 2. Thuật toán quay lui thực hiện tuần tự từ trên xuống dưới, từ trái sang phải theo cơ chế Stack.

Ví dụ 1.1

Liệt kê các khả năng xảy ra kết quả trận đấu bóng chuyên giữa hai đội A và B thi đấu 5 hiệp, đội thắng trước 3 hiệp sẽ là đội thắng chung cuộc.

Ví dụ 1.1

Liệt kê các khả năng xảy ra kết quả trận đấu bóng chuyên giữa hai đội A và B thi đấu 5 hiệp, đội thắng trước 3 hiệp sẽ là đội thắng chung cuộc.



Hình 2: Cây liệt kê lời giải với đề cử đội A thắng hiệp 1.

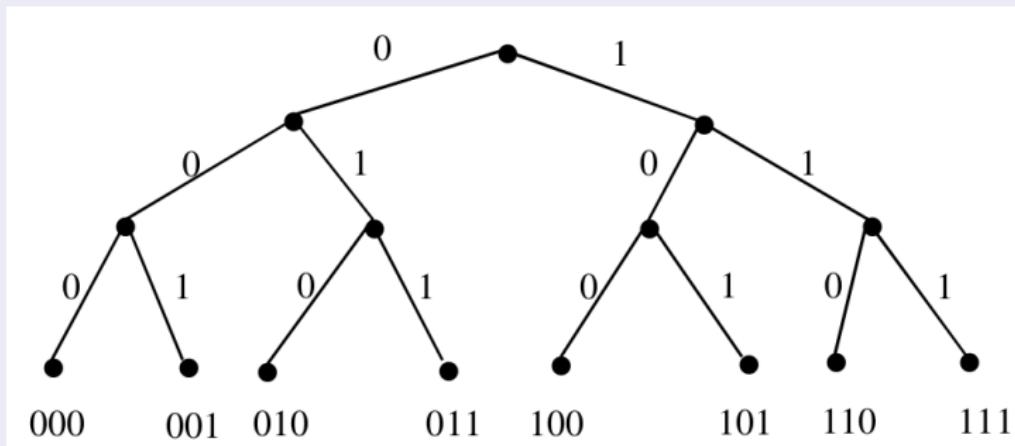
Ví dụ 1.2

Liệt kê các dãy nhị phân n bit.

Ví dụ 1.2

Liệt kê các dãy nhị phân n bit.

Gợi ý: liệt kê dãy nhị phân $x = (x_1, x_2, \dots, x_n)$ với tập để cử cho mỗi bit x_k ($k = 1, n$) là $B = \{0, 1\}$.



Hình 3: Liệt kê các dãy nhị phân 3 bit.

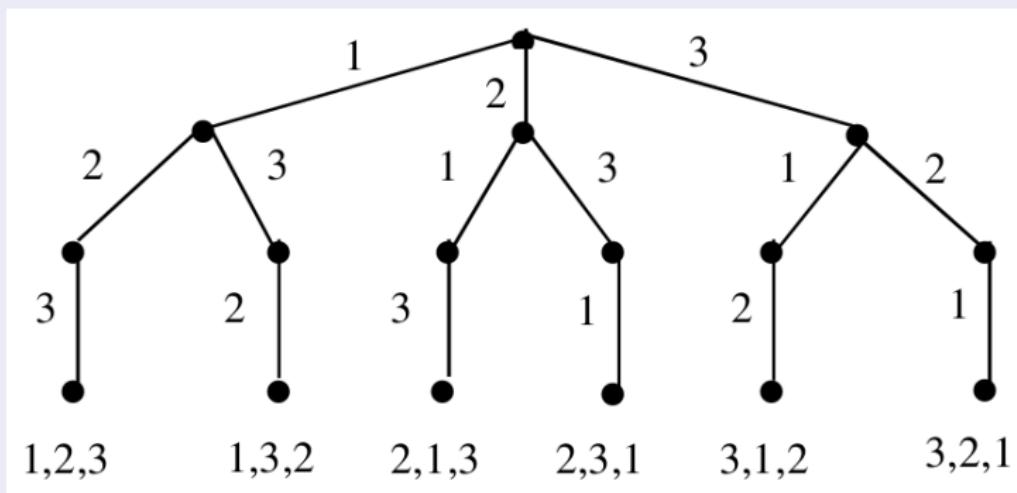
Ví dụ 1.3

Liệt kê các hoán vị của tập $X = \{1, 2, \dots, n\}$.

Ví dụ 1.3

Liệt kê các hoán vị của tập $X = \{1, 2, \dots, n\}$.

Gợi ý: Liệt kê các hoán vị của tập $X = \{1, 2, \dots, n\}$ với tập đê cử cho x_k là $B = \{1, 2, \dots, n\}$.



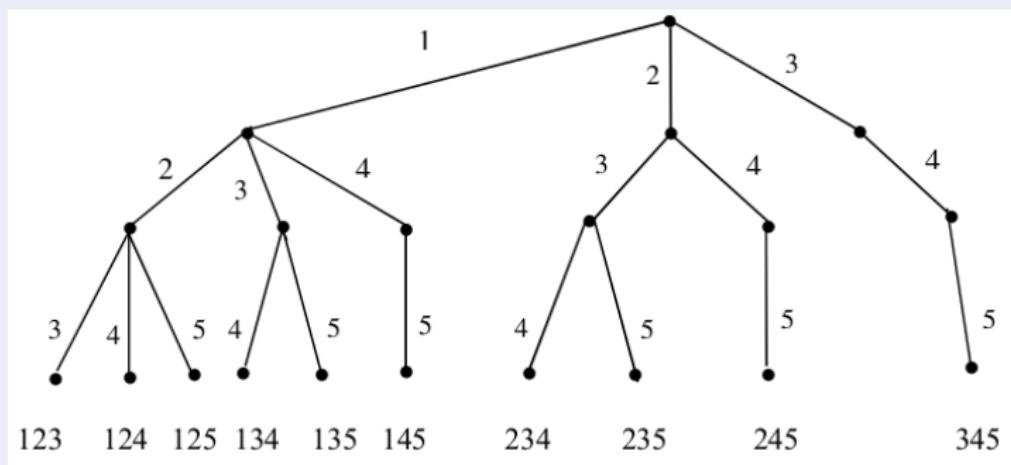
Hình 4: Liệt kê các hoán vị của tập X.

Ví dụ 1.4

Liệt kê các tổ hợp chập m của n phần tử của tập $X = \{1, 2, \dots, n\}$.

Ví dụ 1.4

Liệt kê các tổ hợp chập m của n phần tử của tập $X = \{1, 2, \dots, n\}$.



Hình 5: Liệt kê các tổ hợp chập 3 của $\{1, 2, 3, 4, 5\}$

N-Queens

Ví dụ 1.5

Liệt kê tất cả các cách xếp n quân hậu trên bàn cờ $n \times n$ sao cho chúng không ăn được lẫn nhau. Biết rằng hai quân hậu ăn được nhau nếu chúng nằm trên cùng một hàng, một cột hoặc một đường chéo. Bài toán đặt ra là tìm cách đặt n quân hậu trên bàn cờ sao cho mỗi hàng, mỗi cột và mỗi đường chéo có đúng một quân hậu.

N-Queens

Đặt tập đề cử là $I = \{1, 2, \dots, n\}$, vậy lời giải của bài toán là $X = \{x_1, x_2, \dots, x_n\}$ trong đó $x_i = j$ biểu thị quân hậu ở dòng i được xếp vào cột j ($i, j \in I$).

Lời giải bằng thuật toán quay lui

Lần lượt đề cử cho x_i , ($i = \overline{1, n}$) nhận giá trị j trong I . Ở mỗi bước đề cử giá trị j cho x_i sẽ được chấp nhận nếu ô (i, j) chưa bị các quân hậu trước đó khống chế, nghĩa là không cùng hàng, cột và đường chéo với các quân hậu trước đó đã được đặt lên bàn cờ.

	1	2	3	4
1				
2				
3				
4				

Hình 6: Given 4×4 chessboard.

Example

We have to arrange four queens, Q1, Q2, Q3 and Q4 in 4×4 chess board.

- Let us start with position (1, 1). Q1 is the only queen, so there is no issue. partial solution is $\langle 1 \rangle$
- We cannot place Q2 at positions (2, 1) or (2, 2). Position (2, 3) is acceptable. partial solution is $\langle 1, 3 \rangle$.

Example

- Next, Q3 *cannot* be placed in position (3, 1) as Q1 attacks her. And it *cannot* be placed at (3, 2), (3, 3) or (3, 4) as Q2 attacks her. There is no way to put Q3 in third row. **Hence, the algorithm backtracks and goes back to the previous solution and readjusts the position of queen Q2.** Q2 is moved from positions (2, 3) to (2, 4). **Partial solution is <1, 4>**
- Now, **Q3** can be placed at position (3, 2). **Partial solution is <1, 4, 3>.**
- Queen **Q4** *cannot* be placed anywhere in row four. So again, backtrack to the previous solution and readjust the position of Q3. *Q3 cannot be placed on (3, 3) or(3, 4).* So the algorithm backtracks even further.

Example

- All possible choices for **Q2** are already explored, hence the algorithm goes back to partial solution $\langle 1 \rangle$ and moves the queen Q1 from $(1, 1)$ to $(1, 2)$. And this process continues until a solution is found.

	1	2	3	4
1		Q_1		
2				Q_2
3	Q_3			
4			Q_4	

	1	2	3	4
1			Q_1	
2	Q_2			
3				Q_3
4		Q_4		

Hình 7: All possible solutions for 4-queen

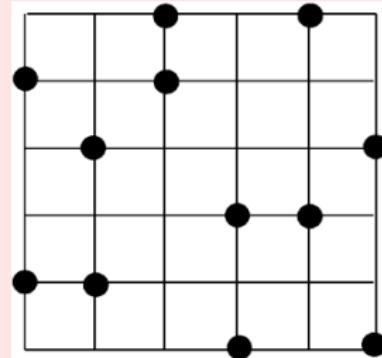
Ứng dụng

Thuật toán quay lui (quay lui-cải tiến) có thể giải được nhiều bài toán liệt kê tổ hợp rời rạc mà chúng ta không có phương pháp để giải, chẳng hạn như bài toán đặt $2n$ điểm trên lưới $n \times n$, bài toán hình lục giác thần bí, bài toán xếp hậu, bài toán người du lịch, bài toán cái túi ...

Bài toán đặt $2n$ điểm trên lưới $n \times n$

Cho một lưới ô vuông kích thước $n \times n$ điểm. Hỏi, có thể đặt $2n$ điểm lên lưới sao cho mỗi hàng, mỗi cột, mỗi đường chéo có chính xác 2 điểm được hay không?

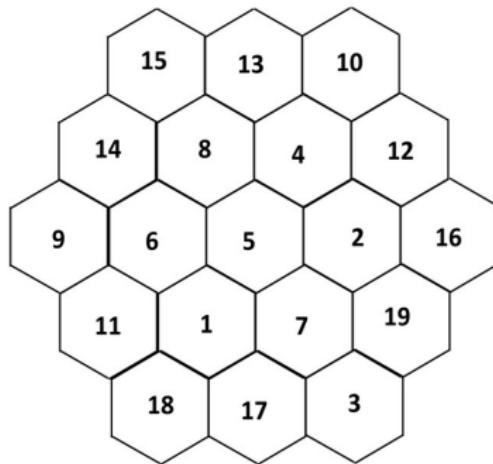
Bài toán trên chỉ có thể được giải bởi thuật toán quay lui có độ phức tạp thời gian tính toán $O(n!)$. Hiện nay đã có lời giải của bài toán này với $n \leq 15$.



Hình 8: Một lời giải
của bài toán lưới
vuông với $n = 6$

Bài toán hình lục giác thần bí

Điền các số từ 1 đến 19 vào các ô của hình lục giác bên dưới sao cho tổng các ô theo các hướng của lục giác đều bằng nhau.



Hình 9: Hình lục giác thần bí.

Bài toán tối ưu: Là bài toán tìm ra tổ hợp tốt nhất trong những tổ hợp có thể tạo ra, thỏa mãn yêu cầu cho trước. Tối ưu tổ hợp có rất nhiều ứng dụng trong thực tế như:

- Xếp ba lô (1): có 1 chiếc ba lô, mang được không quá trọng lượng b . Có n đồ vật với trọng lượng: $a_1 \dots a_n$ và giá trị $c_1 \dots c_n$ tương ứng. Hỏi ta xếp vào ba lô những vật nào để mang được giá trị lớn nhất?
- Xếp ba lô (2): tương tự như bài 1 nhưng mỗi loại đồ vật có thể mang theo từ $0 \rightarrow m$ lần

Bài toán người bán hàng:

- 1 người bán hàng cần giao hàng đến n điểm: T_1, \dots, T_n .
- Đường đi: Xuất phát từ một địa điểm T_i , đi qua tất cả các điểm còn lại, mỗi nơi đi qua đúng 1 lần rồi quay trở lại vị trí xuất phát.
- Biết C_{ij} là chi phí đi từ địa điểm T_i đến T_j .
- Yêu cầu: Hãy tìm một hành trình thỏa mãn yêu cầu có tổng chi phí nhỏ nhất

Bài toán phân công

- Có n công việc và n thợ. C_{ij} là chi phí để trả cho thợ i làm công việc j .
- Hãy tìm cách thuê thợ sao cho tổng chi phí là nhỏ nhất.
- Lưu ý: mỗi thợ chỉ làm 1 việc và 1 việc chỉ làm bởi 1 thợ.

Bài toán trả tiền ATM

- Khách hàng yêu cầu rút số tiền n . Trong cây ATM có các loại tiền với mệnh giá: a_1, \dots, a_m . Tính xem cây phải trả tiền như thế nào để số tờ tiền là ít nhất?

Định nghĩa 2.1

Cho tập hữu hạn phần tử D . Hàm mục tiêu $f(X)$ xác định trên D . Mỗi phần tử $X \in D$ có dạng $X = (x_1, x_2, \dots, x_n)$ được gọi là 1 phương án. *Tìm phương án X_0 sao cho $f(X_0)$ đạt cực đại (cực tiểu) trên D .* Phương án X_0 được gọi là phương án tối ưu.

Ví dụ 2.1

Bài toán xếp ba lô 1 (mỗi đồ vật chọn không quá 1 lần):

- *Tập phương án (D):* Tập các bộ n phần tử (x_1, \dots, x_n) trong đó $x_i = 0$ nếu không chọn vật thứ i và bằng 1 nếu chọn.
- *Hàm mục tiêu (f):* Tổng giá trị các đồ vật xếp được $f(x) = \sum_{i=1}^n x_i c_i$.
- *Điều kiện:* Tổng khối lượng không quá b : $\sum_{i=1}^n x_i a_i \leq b$.
- *Yêu cầu:* Tìm phương án thỏa mãn điều kiện và làm hàm mục tiêu đạt cực đại

Ví dụ 2.2

Tìm nghiệm nguyên không âm (x, y, z) của phương trình $x + y + z = 9$ sao cho xyz là lớn nhất?

Mỗi nghiệm của phương trình hay một phương án của bài toán đã cho có dạng (x, y, z) . Tập phương án của bài toán là $D = \{(0, 0, 9), (0, 1, 8), \dots, (8, 1, 0), (9, 0, 0)\}$, hàm mục tiêu của bài toán là $f(x, y, z) = xyz$. Theo bất đẳng thức Cauchy, phương án tối ưu của bài toán là $X_{max} = (3, 3, 3)$ và giá trị tối ưu là $f(X_{max}) = 27$.

Phương pháp dễ nhận biết để giải một bài toán tối ưu tổ hợp là trên cơ sở thuật toán quay lui liệt kê các cấu hình tổ hợp, với mỗi phương án tìm được

- tính giá trị hàm mục tiêu tại đó
- so sánh các giá trị hàm mục tiêu của các phương án đã tìm được để chọn ra phương án tối ưu

Thực tế, phương pháp duyệt toàn bộ hay phương pháp **vết cạn toàn bộ các cấu hình tổ hợp** là khó để thực hiện bởi số lượng cấu hình tổ hợp thường khá lớn. Để lựa chọn phương pháp giải các bài toán tối ưu tổ hợp theo yêu cầu và phù hợp với thực tế, sử dụng lý thuyết **NP-C** (Non-Deterministic Polynomial-Complete)

Trong lý thuyết **NP-C**, các bài toán tối ưu tổ hợp được chia thành hai lớp, đó là lớp các **bài toán dễ** và lớp các **bài toán khó**:

- ➊ **dễ** nếu phương án tối ưu của bài toán được xác định bởi một *thuật toán đa thức* có **độ phức tạp thời gian tính toán** $O(n^k)$, n là kích thước của dữ liệu đầu vào, $k \in N^+$.
- ➋ **khó** nếu phương án tối ưu của bài toán trong trường hợp tổng quát được xác định bởi *thuật toán mũ* có **độ phức tạp thời gian tính toán** $O(k^n)$ hoặc $O(n!)$. Lớp các bài toán tối ưu tổ hợp khó được ký hiệu bởi **NP-Hard** (Non-Deterministic Polynomial-Hard).

Phương pháp quy hoạch động

Phương pháp quy hoạch động giải quyết các bài toán tối ưu tổ hợp bằng cách tổ hợp lời giải của các bài toán con có **cùng cấu trúc** để từ đó suy ra lời giải của bài toán đã cho. Thường được sử dụng để giải các bài toán tối ưu tổ hợp thuộc lớp NP-Hard.

Phương pháp nhánh-cận

Là một sự **cải tiến** của thuật toán quay lui dựa trên cơ sở xác định cận dưới/trên **tương ứng** với bài toán **min/max** cho từng phương án bộ phận. Thay vì duyệt tất cả các trường hợp, nếu ta đến một vị trí mà **giá trị** của hàm mục tiêu tại đó và các **điểm** **về sau** **chắc chắn** không tốt nhất thì quay lại.

Định nghĩa 2.2 (Phương pháp nhánh-cận)

Xét bài toán: tìm min $\{f(x) | x \in D\}$,

$D = \{x = (x_1, \dots, x_n) \in A_1 \times \dots \times A_n\}$. Chúng ta sử dụng thuật toán quay lui để xây dựng các phương án x của bài toán. Một bộ (a_1, \dots, a_k) là phương án bộ phận cấp k , tập các phương án bộ phận cấp k được ký hiệu là D_k . Giả sử tồn tại hàm g thỏa mãn:

$$g(a_1, \dots, a_k) \leq \min\{f(x) | x \in D, x_i = a_i, i = \overline{1, k}\}$$

Khi đó giá trị hàm g tại phương án bộ phận không vượt quá giá trị nhỏ nhất của hàm mục tiêu trên nhánh đó và $g(a_1, \dots, a_k)$ là cận dưới của nhánh chứa phương án bộ phận (a_1, \dots, a_k)

Phương pháp nhánh-cận

Giả sử đã có hàm g . Trong quá trình thực hiện thuật toán quay lui, ta gọi:

- \bar{x} là phương án tốt nhất đã tìm được
- $\bar{f} = f(\bar{x})$ là kỷ lục
- Nếu tại bước k , ta có phương án bộ phận (a_1, \dots, a_k) mà $g(a_1, \dots, a_k) > \bar{f} \implies$ tập con của D chứa các phương án mở rộng của (a_1, \dots, a_k) **sẽ không phải là kết quả tốt nhất** \implies quay lui.

Vấn đề phát sinh với hàm g

- Tính giá trị của g phải đơn giản hơn việc tìm phương án tối ưu trong nhánh
- Kết quả của hàm g phải gần với kết quả tối ưu của nhánh

Bài toán người du lịch - TSP (Travelling Salesman Problem)

Bài toán phát biểu như sau: Có n thành phố $1, 2, \dots, n$. Xuất phát từ một thành phố nào đó, người du lịch muốn đi tham quan tất cả các thành phố, mỗi thành phố một lần, rồi quay trở về thành phố xuất phát. Cho biết c_{ij} là chi phí đi từ thành phố i đến thành phố j ($i, j = 1, 2, \dots, n$). Xác định một hành trình du lịch sao cho tổng chi phí là nhỏ nhất.

Bài toán lập lịch gia công chi tiết - SP (Scheduling Problem)

Có n chi tiết D_1, D_2, \dots, D_n lần lượt được gia công trên m máy M_1, M_2, \dots, M_m . Cho biết thời gian gia công chi tiết D_i trên máy M_j là t_{ij} , hãy xác định lịch trình (thứ tự) gia công các chi tiết trên các máy sao cho hoàn thành công việc sớm nhất.

Sinh viên viết báo cáo tổng quan về các kỹ thuật quy hoạch động, nhánh cận, so sánh ưu và nhược điểm của hai kỹ thuật. Chọn 3 bài toán trong các bài toán dưới đây để viết báo cáo.

- ① Bài toán Tháp Hà Nội, N-Queens
- ② Bài toán người du lịch - TSP (Travelling Salesman Problem)
- ③ Bài toán người bán hàng
- ④ Bài toán xếp ba lô 1, 2
- ⑤ Bài toán hình lục giác thần bí, bài toán trả tiền ATM
- ⑥ Bài toán lập lịch gia công chi tiết- SP (Scheduling Problem)
- ⑦ Phương pháp người láng giềng gần nhất cho bài toán TSP.

Yêu cầu

Sinh viên cần:

- viết rõ lý thuyết của các kỹ thuật giải, giải thích chi tiết code.
- nghiêm cấm sao chép. Các bài có độ tương đồng $\geq 90\%$ sẽ được 0 điểm cho tất cả.

Tài liệu tham khảo



D.N. An

Giáo Trình Toán Rời Rạc. *Trường DH Nha Trang, (2021)*.



Giáo trình Toán rời rạc

Giáo trình Toán Rời Rạc. *Trường DHSP Huế. (2003), 22-35.*



N.T. Nhựt

Bài giảng Toán Rời Rạc. *Trường DH KHTN Tp.HCM. (2011).*