

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

Thuật toán di truyền (Genetic Algorithm)

Đoàn Duy Tùng

CTTN - Khoa học máy tính - K67
SOICT, HUST

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

Cho không gian tìm kiếm, rời rạc X , và một hàm số $f : X \rightarrow \mathbb{R}$. Ta cần giải bài toán tối ưu:

$$\arg \min_{x \in X} f(x)$$

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

Cho không gian tìm kiếm, rời rạc X , và một hàm số $f : X \rightarrow \mathbb{R}$. Ta cần giải bài toán tối ưu:

$$\arg \min_{x \in X} f(x)$$

Bài toán tối ưu tổ hợp (combinatorial optimization problem).

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

- 1 Với thuật toán Brute Force, tức là duyệt qua toàn bộ các phần tử trong không gian tìm kiếm, thời gian dẫn tới lời giải sẽ rất lâu.

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

- 1 Với thuật toán Brute Force, tức là duyệt qua toàn bộ các phần tử trong không gian tìm kiếm, thời gian dẫn tới lời giải sẽ rất lâu.
- 2 Một trong những hướng tiếp cận giải bài toán này là giải thuật di truyền.

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

Thuật toán GA sẽ hoạt động theo những nguyên lý của thuyết tiến hóa Darwin:

- 1 Di truyền: Phải có một cơ chế cho phép các sinh vật "cha mẹ" trong một thế hệ truyền lại các đặc điểm của họ cho các sinh vật "con cái" trong thế hệ tiếp theo.

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

Thuật toán GA sẽ hoạt động theo những nguyên lý của thuyết tiến hóa Darwin:

- ❶ Di truyền: Phải có một cơ chế cho phép các sinh vật "cha mẹ" trong một thể hệ truyền lại các đặc điểm của họ cho các sinh vật "con cái" trong thế hệ tiếp theo.
- ❷ Sự đa dạng: Phải có sự đa dạng về các đặc điểm có mặt trong quần thể của các sinh vật, hoặc thêm sự đa dạng từ các yếu tố bên ngoài vào quần thể.

Thuật toán GA sẽ hoạt động theo những nguyên lý của thuyết tiến hóa Darwin:

- ❶ Di truyền: Phải có một cơ chế cho phép các sinh vật "cha mẹ" trong một thể hệ truyền lại các đặc điểm của họ cho các sinh vật "con cái" trong thế hệ tiếp theo.
- ❷ Sự đa dạng: Phải có sự đa dạng về các đặc điểm có mặt trong quần thể của các sinh vật, hoặc thêm sự đa dạng từ các yếu tố bên ngoài vào quần thể.
- ❸ Lựa chọn: Phải có một cơ chế để một số sinh vật có cơ hội trở thành cha mẹ và truyền lại thông tin di truyền của họ, trong khi những sinh vật khác không có cơ hội đó.

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

Mỗi vector x trong X được mã hóa thành các "kiểu gen", là một chuỗi ký tự s , độ dài l , các ký tự được chọn từ một alphabet A , và ngược lại mỗi chuỗi ký tự s thể hiện một vector x duy nhất.

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

Mỗi vector x trong X được mã hóa thành các "kiểu gen", là một chuỗi ký tự s , độ dài l , các ký tự được chọn từ một alphabet A , và ngược lại mỗi chuỗi ký tự s thể hiện một vector x duy nhất.

Nói cách khác ta sử dụng một song ánh $c : A^l \rightarrow X$.

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

Mỗi vector x trong X được mã hóa thành các "kiểu gen", là một xâu ký tự s , độ dài l , các ký tự được chọn từ một alphabet A , và ngược lại mỗi xâu ký tự s thể hiện một vector x duy nhất.

Nói cách khác ta sử dụng một song ánh $c : A^l \rightarrow X$.

Bài toán tối ưu lúc này trở thành:

$$\arg \min_{s \in S} f(c(s)), S \subseteq A^l.$$

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

- 1 Mã hóa nhị phân: Phổ biến nhất, $A = \{0, 1\}$. Ví dụ:

1	0	1	0	0	1
---	---	---	---	---	---

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

- 1 Mã hóa nhị phân: Phổ biến nhất, $A = \{0, 1\}$. Ví dụ:

1	0	1	0	0	1
---	---	---	---	---	---

- 2 Mã hóa hoán vị: Phù hợp với các bài toán liên quan đến hoán vị. Chẳng hạn như trong bài toán người du lịch, ta sử dụng cách mã hóa lời giải thành một chuỗi số, thể hiện các thành phố mà người này đi tới.

1	5	3	4	2	6
---	---	---	---	---	---

- ❶ Mã hóa nhị phân: Phổ biến nhất, $A = \{0, 1\}$. Ví dụ:

1	0	1	0	0	1
---	---	---	---	---	---

- ❷ Mã hóa hoán vị: Phù hợp với các bài toán liên quan đến hoán vị. Chẳng hạn như trong bài toán người du lịch, ta sử dụng cách mã hóa lời giải thành một chuỗi số, thể hiện các thành phố mà người này đi tới.

1	5	3	4	2	6
---	---	---	---	---	---

- ❸ Mã hóa đa giá trị: Sử dụng để mã hóa những giá trị phức tạp. Ví dụ:

5.16	3.78	9.12	5.64
------	------	------	------

Giới thiệu thuật toán

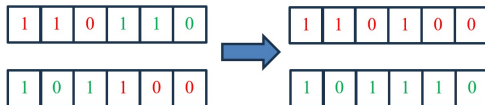
Các thành phần của thuật toán

Thuật toán

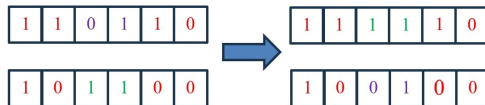
Crossover rate: Khả năng mà hai cá thể sẽ xảy ra trao đổi chéo.

Crossover rate: Khả năng mà hai cá thể sẽ xảy ra trao đổi chéo.

❶ Một điểm cắt:



❷ Hai điểm cắt:

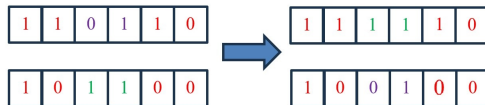


Crossover rate: Khả năng mà hai cá thể sẽ xảy ra trao đổi chéo.

❶ Một điểm cắt:



❷ Hai điểm cắt:



❸ Có thể tổng quát với N điểm cắt.

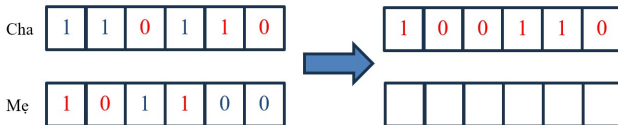
Mỗi vị trí của cá thể con được kế thừa từ cha và mẹ với xác suất bằng nhau, được thể hiện thông qua crossover mask:
Crossover mask là một dãy các bit 0-1 , trong đó 0 thể hiện cho gen từ cha, 1 thể hiện gen từ mẹ.

Mỗi vị trí của cá thể con được kế thừa từ cha và mẹ với xác suất bằng nhau, được thể hiện thông qua crossover mask:

Crossover mask là một dãy các bit 0-1, trong đó 0 thể hiện cho gen từ cha, 1 thể hiện gen từ mẹ.

Ví dụ:

Crossover mask: **1 1 0 1 0 0**

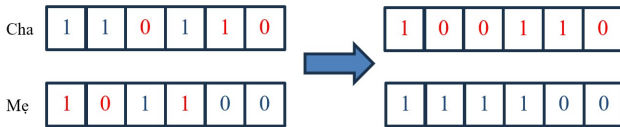


Mỗi vị trí của cá thể con được kế thừa từ cha và mẹ với xác suất bằng nhau, được thể hiện thông qua crossover mask:

Crossover mask là một dãy các bit 0-1, trong đó 0 thể hiện cho gen từ cha, 1 thể hiện gen từ mẹ.

Ví dụ:

Crossover mask: **1 1 0 1 0 0**

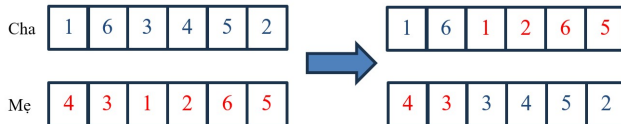


Giới thiệu thuật toán

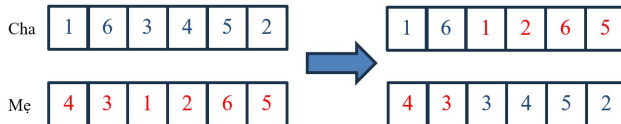
Các thành phần của thuật toán

Thuật toán

Đặt vấn đề: Bài toán TSP



Đặt vấn đề: Bài toán TSP



Đối với non-linear encoding, cần các non-linear crossover.

❶ Bước 1: Chọn ngẫu nhiên một crossover range:

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

- ❶ Bước 1: Chọn ngẫu nhiên một crossover range:

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

- ❷ Bước 2: Thực hiện tạo đời con bằng cách hoán đổi các gen của cha mẹ trong crossover range đó:

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

❶ Bước 1: Chọn ngẫu nhiên một crossover range:

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

❷ Bước 2: Thực hiện tạo đời con bằng cách hoán đổi các gen của cha mẹ trong crossover range đó:

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

❸ Bước 3: Rút ra quy tắc hoán đổi trong bước 2:

$$1 \leftrightarrow 6 \leftrightarrow 3$$

$$2 \leftrightarrow 5$$

$$9 \leftrightarrow 4$$

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

- ❶ Bước 4: Dựa trên quy tắc hoán đổi từ bước 3, xác định vị trí còn lại của đời con:

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

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

Ví dụ:

P ₁	1	2	3	4	5	6	7	8	9
P ₂	4	5	2	1	8	7	6	9	3

Ví dụ:

P ₁	1	2	3	4	5	6	7	8	9
P ₂	4	5	2	1	8	7	6	9	3

- 1) Bước 1: Chọn một crossover range, sau đó bước đầu hình thành đời con dựa trên hoán đổi các gen trong range đó:

O ₁	*	*	*	1	8	7	6	*	*
O ₂	*	*	*	4	5	6	7	*	*

O_1	*	*	*	1	8	7	6	*	*
O_2	*	*	*	4	5	6	7	*	*

- 2) Bước 2: Viết các phần tử của P_2 theo thứ tự từ điểm cắt thứ hai, ta được dãy:

9	3	4	5	2	1	8	7	6
---	---	---	---	---	---	---	---	---

O_1	*	*	*	1	8	7	6	*	*
O_2	*	*	*	4	5	6	7	*	*

- 2) Bước 2: Viết các phần tử của P_2 theo thứ tự từ điểm cắt thứ hai, ta được dãy:

9	3	4	5	2	1	8	7	6
---	---	---	---	---	---	---	---	---

- 3) Bước 3: Bỏ đi các phần tử đã xuất hiện trong O_2 là 4,5,6,7; ta chỉ còn dãy:

9	3	2	1	8
---	---	---	---	---

- 4) Bước 4: Chèn dãy trên vào O_2 , theo thứ tự đó từ điểm cắt thứ hai, ta hoàn thành O_2 :

O_2	2	1	8	4	5	6	7	9	3
-------	---	---	---	---	---	---	---	---	---

Tương tự xác định O_1 .

Mutation rate(per genes): tần số đột biến, thể hiện cho số lượng cá thể bị đột biến trong một thế hệ.

Mutation rate (per bit): tần số đột biến, thể hiện cho số vị trí bị đột biến trong một cá thể.

Mutation rate(per genes): tần số đột biến, thể hiện cho số lượng cá thể bị đột biến trong một thế hệ.

Mutation rate (per bit): tần số đột biến, thể hiện cho số vị trí bị đột biến trong một cá thể.

Các phương pháp đột biến:

1. Đổi biến đảo bit: Chọn ngẫu nhiên một (hay nhiều) , đảo bit đó từ 0 thành 1 và ngược lại.

Trước

1	0	1	1	0	1
---	---	---	---	---	---

Sau

1	1	1	0	0	1
---	---	---	---	---	---

2. Đổi chỗ: Chọn hai vị trí trong xâu, đổi chỗ chúng.

Trước

1	2	3	4	5	6
---	---	---	---	---	---

Sau

1	4	3	2	5	6
---	---	---	---	---	---

3. Đảo đoạn: Chọn hai vị trí ngẫu nhiên, đảo lại đoạn nằm giữa hai vị trí đó:

Trước

1	2	3	4	5	6
---	---	---	---	---	---

Sau

1	5	4	3	2	6
---	---	---	---	---	---

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

Fitness: Đánh giá độ thích nghi, phù hợp của cá thể.

Một cách tổng quát, fitness bằng $h(f(c(s)))$, $h: \mathbb{R} \rightarrow \mathbb{R}^+$ là một hàm số nhận giá trị dương, đơn điệu nào đó.

Fitness: Đánh giá độ thích nghi, phù hợp của cá thể.

Một cách tổng quát, fitness bằng $h(f(c(s)))$, $h: \mathbb{R} \rightarrow \mathbb{R}^+$ là một hàm số nhận giá trị dương, đơn điệu nào đó.

Bánh xe Roulette:

- ❶ Tính tổng độ thích nghi của quần thể: $F = \sum_{i=1}^N f_i, i = 1, \dots, n.$
- ❷ Xác suất lựa chọn cá thể thứ i là: $p_i = \frac{f_i}{F}, i = 1, \dots, n.$
- ❸ Xác suất quỹ tích của cá thể thứ i là: $q_i = \sum_{j=1}^i p_j, i = 1, \dots, n.$
- ❹ Chọn ngẫu nhiên một số thực $r \in [0, 1)$. Cá thể thứ i ($i = 1, \dots, n$) được lựa chọn nếu $q_{i-1} \leq r < q_i$ (coi $q_0 = 0$).

Ranking Selection: Đầu tiên, ta xếp hạng các cá thể theo fitness của chúng trước, sau đó tính lại đại lượng Fitness theo xếp hạng này:

Cá thể yếu nhất có fitness 1, cá thể yếu nhì có fitness 2, ..., cá thể mạnh nhất có fitness là N .

Sau đó thực hiện giống như phương pháp vòng quay Roulette.

Ranking Selection: Đầu tiên, ta xếp hạng các cá thể theo fitness của chúng trước, sau đó tính lại đại lượng Fitness theo xếp hạng này:

Cá thể yếu nhất có fitness 1, cá thể yếu nhì có fitness 2, ..., cá thể mạnh nhất có fitness là N .

Sau đó thực hiện giống như phương pháp vòng quay Roulette.

Tournament Selection:

- 1 Chọn ngẫu nhiên k cá thể (tournament size) vào một tournament.
- 2 Lấy cá thể tốt nhất (dựa trên fitness) trong tournament đó vào thế hệ tiếp theo.
- 3 Để thế hệ tiếp theo có p cá thể, ta thực hiện p tournament.

Ranking Selection: Đầu tiên, ta xếp hạng các cá thể theo fitness của chúng trước, sau đó tính lại đại lượng Fitness theo xếp hạng này:

Cá thể yếu nhất có fitness 1, cá thể yếu nhì có fitness 2, ..., cá thể mạnh nhất có fitness là N .

Sau đó thực hiện giống như phương pháp vòng quay Roulette.

Tournament Selection:

- 1 Chọn ngẫu nhiên k cá thể (tournament size) vào một tournament.
- 2 Lấy cá thể tốt nhất (dựa trên fitness) trong tournament đó vào thế hệ tiếp theo.
- 3 Để thế hệ tiếp theo có p cá thể, ta thực hiện p tournament.

Elitism: Đảm bảo rằng k trong số cá thể tốt nhất sẽ chắc chắn có mặt trong thế hệ tiếp theo, còn lại được chọn lọc theo những phương pháp trên.

Giới thiệu thuật toán

Các thành phần của thuật toán

Thuật toán

