



Học viện Công nghệ Bưu chính Viễn thông
Khoa Công nghệ thông tin 1

Nhập môn trí tuệ nhân tạo

Tìm kiếm cục bộ
(local search)

Nội dung

- ❑ Giới thiệu tìm kiếm cục bộ
- ❑ Thuật toán leo đồi (Hill climbing)
- ❑ Thuật toán tô thép (Simulated Annealing)

Tìm kiếm cục bộ

- ❑ Các thuật toán tìm kiếm đã học (mù hoặc có thông tin) khảo sát không gian tìm kiếm một cách hệ thống theo một số quy tắc nhất định
 - ❑ Cần **lưu lại thông tin** về trạng thái và đường đi đã khảo sát
 - ❑ Không thích hợp cho bài toán có không gian trạng thái lớn
- ❑ Các thuật toán cục bộ tại một thời điểm chỉ xem xét trạng thái hiện thời và các trạng thái lân cận
 - ❑ **Không lưu lại thông tin** về trạng thái và đường đi đã khảo sát
 - ❑ Tiết kiệm thời gian và bộ nhớ
 - ❑ Có thể áp dụng cho các bài toán có không gian trạng thái lớn
 - ❑ Không cho lời giải tối ưu



Bài toán tối ưu hóa tổ hợp (rời rạc)

- ▶ Tìm trạng thái tối ưu hoặc tổ hợp tối ưu trong không gian rời rạc các trạng thái
 - Không quan tâm tới đường đi
- ▶ Không gian trạng thái rất lớn
 - Không thể sử dụng các phương pháp tìm kiếm đã học để duyệt tất cả các trạng thái
- ▶ Không tồn tại thuật toán cho phép tìm lời giải tốt nhất với độ phức tạp tính toán nhỏ
 - Có thể chấp nhận lời giải **tương đối tốt**
- ▶ Ví dụ: Bài toán lập kế hoạch, lập thời khóa biểu, bài toán một triệu con hậu, ...

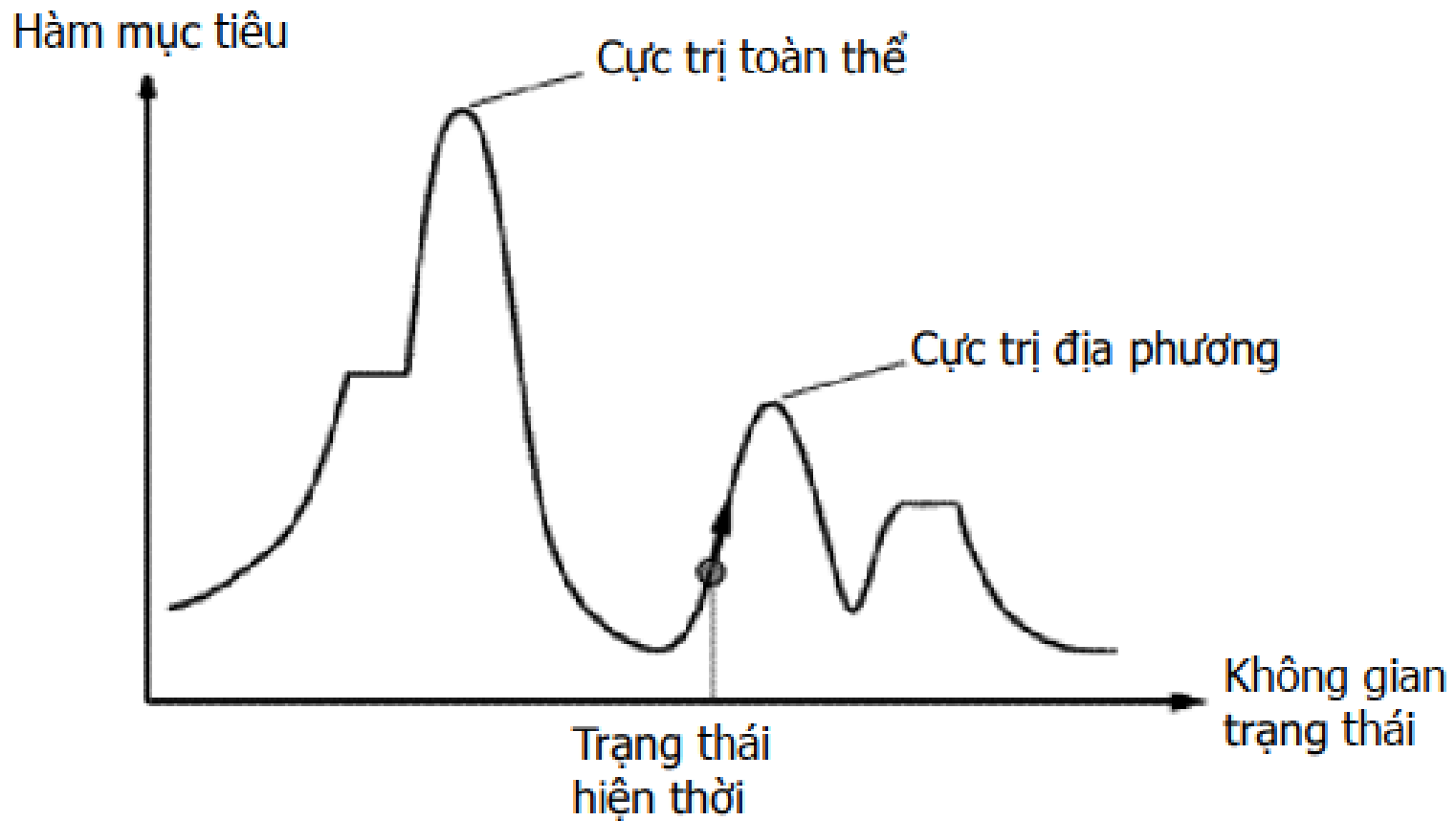
Tìm kiếm cục bộ: tư tưởng

- ▶ Khác với bài toán tìm kiếm thông thường, tìm kiếm cục bộ **chỉ quan trọng trạng thái đích** (trạng thái tốt nhất), **không quan trọng đường đi**
 - Mỗi trạng thái tương ứng với một lời giải (chưa tối ưu)
- ▶ **Cải thiện dần** (iterative improvement) lời giải bằng cách xuất phát từ một trạng thái, sau đó thay đổi để chuyển sang trạng thái có hàm mục tiêu tốt hơn
- ▶ Thay đổi trạng thái bằng cách thực hiện các **chuyển động**
 - Trạng thái nhận được từ một trạng thái n bằng cách thực hiện các chuyển động gọi là hàng xóm của n

Phát biểu bài toán tìm kiếm cục bộ

- ▶ Không gian trạng thái X
- ▶ Hàm mục tiêu $Obj: X \rightarrow R$
- ▶ Tập chuyển động để sinh ra hàng xóm
 - $N(x)$ là tập các hàng xóm của x
- ▶ Yêu cầu: Tìm trạng thái x^* sao cho $Obj(x^*)$ là lớn nhất hoặc nhỏ nhất

Minh họa tìm kiếm cục bộ



Nội dung

- ❑ Giới thiệu tìm kiếm cục bộ
- ❑ Thuật toán leo đồi (Hill climbing)
- ❑ Thuật toán tô thép (Simulated Annealing)

Thuật toán leo đồi: tư tưởng

- ▶ **Leo đồi:** là tên chung của một họ thuật toán cùng nguyên lý
- ▶ **Cách thức:** Từ trạng thái hiện tại, xem xét tập hàng xóm, di chuyển sang trạng thái tốt hơn
 - Chọn trạng thái hàng xóm để di chuyển thế nào?
- ▶ **Trạng thái đích:** Thuật toán dừng lại khi không có trạng thái hàng xóm nào tốt hơn
 - Thuật toán có thể tìm được cực trị hoặc cực trị địa phương

Di chuyển sang trạng thái tốt nhất

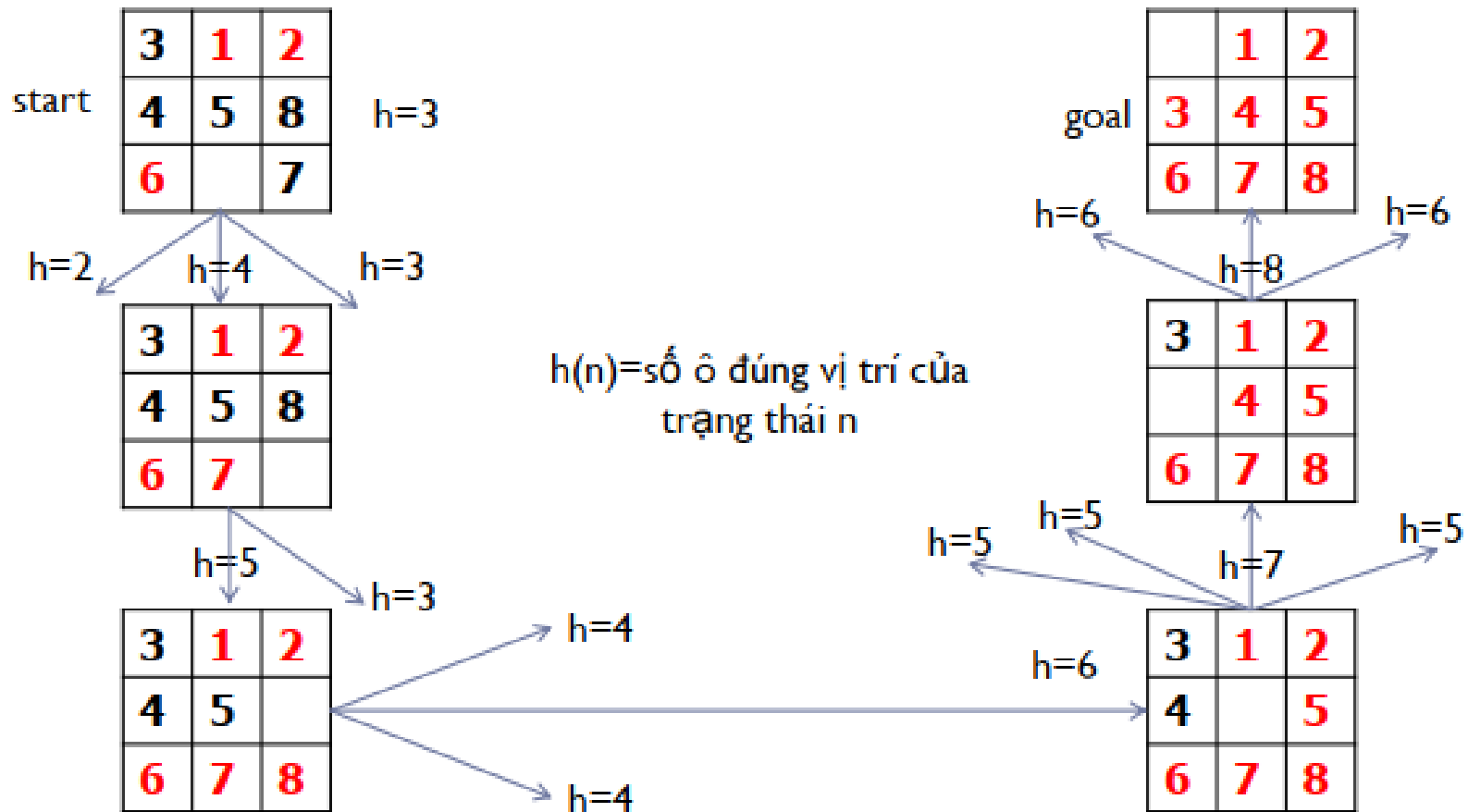
Đầu vào: bài toán tối ưu tổ hợp

Đầu ra: trạng thái với hàm mục tiêu lớn nhất (hoặc cực đại địa phương)

1. Chọn ngẫu nhiên trạng thái x
2. Gọi Y là tập các trạng thái hàng xóm của x
3. **if** $\forall y_i \in Y: Obj(y_i) < Obj(x)$
 - **return** x
4. $x \leftarrow y_i$ trong đó $i = \operatorname{argmax}_i (Obj(y_i))$
5. **Go to** 2

Di chuyển
sang trạng
thái tốt nhất

Ví dụ leo đồi



Tính chất thuật toán leo đồi

- ▶ Đơn giản, dễ lập trình
- ▶ Không tốn bộ nhớ (không phải ghi nhớ các trạng thái)
- ▶ Dễ bị lời giải **tối ưu cục bộ** (**cực trị địa phương**)
- ▶ Việc lựa chọn chuyển động rất quan trọng, không có quy tắc chung
 - Nếu có quá nhiều chuyển động
 - Sinh ra quá nhiều hàng xóm
 - Mất nhiều thời gian lựa chọn phương án tốt nhất
 - Nếu quá ít chuyển động
 - Rất dễ bị cực trị địa phương

Leo đồi ngẫu nhiên: tư tưởng

- ▶ Là một phiên bản khác của thuật toán leo đồi
- ▶ Lựa chọn ngẫu nhiên một trạng thái hàng xóm
 - Chuyển sang trạng thái hàng xóm nếu trạng thái này tốt hơn
 - Nếu không tốt hơn lại chọn ngẫu nhiên một hàng xóm khác
- ▶ Kết thúc khi nào hết **kiên nhẫn**
 - Số hàng xóm mà thuật toán xem xét trong mỗi bước lặp hoặc trong toàn bộ thuật toán

Leo đồi ngẫu nhiên

1. Chọn ngẫu nhiên trạng thái x
2. Gọi Y là tập các trạng thái hàng xóm của x
3. Chọn ngẫu nhiên $y_i \in Y$
4. **if** $Obj(y_i) > Obj(x)$
 - $x \leftarrow y_i$
5. **Go to** 2 nếu chưa hết kiên nhẫn

Vấn đề: Chọn tiêu chuẩn kết thúc thế nào?

Một số tính chất

- ▶ Trường hợp mỗi trạng thái có nhiều láng giềng
 - Leo đồi ngẫu nhiên thường cho kết quả nhanh hơn, và ít gặp cực trị địa phương hơn

- ▶ Với những không gian trạng thái có ít cực trị địa phương
 - Các thuật toán leo đồi thường tìm được lời giải khá nhanh

- ▶ Với những không gian phức tạp
 - Các thuật toán leo đồi thường chỉ tìm được cực trị địa phương
 - Bằng cách thực hiện nhiều lần với trạng thái xuất phát ngẫu nhiên, leo đồi thường tìm được cực trị địa phương khá tốt

Nội dung

- ❑ Giới thiệu tìm kiếm cục bộ
- ❑ Thuật toán leo đồi (Hill climbing)
- ❑ Thuật toán tô thép (Simulated Annealing)

Thuật toán tô thép: tư tưởng

- ▶ Là phiên bản khái quát hoá của leo đồi ngẫu nhiên
- ▶ **Mục tiêu:** Giải quyết phần nào vấn đề cực trị địa phương trong các thuật toán leo đồi
- ▶ **Nguyên tắc chung:** chấp nhận những trạng thái kém hơn trạng thái hiện thời với một xác suất p
 - Chọn xác suất p thế nào?

Lựa chọn p

- ▶ **Nguyên tắc:** không chọn p cố định, giá trị p phụ thuộc hai yếu tố
 - Nếu trạng thái mới kém hơn nhiều so với trạng thái hiện thời, thì p phải giảm đi
 - Xác suất chấp nhận trạng thái tỉ lệ nghịch với độ kém của trạng thái
 - Theo thời gian, giá trị của p phải giảm dần
 - Khi mới bắt đầu, thuật toán chưa ở vùng trạng thái tốt, do vậy chấp nhận thay đổi lớn
 - Theo thời gian, thuật toán chuyển sang vùng trạng thái tốt, do vậy cần hạn chế thay đổi

Thuật toán tô thép

$SA(X, Obj, N, m, x, C)$

Đầu vào: số bước lặp m

trạng thái bắt đầu x (chọn ngẫu nhiên)

sơ đồ làm lạnh C

Đầu ra: trạng thái tốt nhất x^* (cực đại hàm mục tiêu)

Khởi tạo: $x^* = x$

for $i = 1$ **to** m

1. chọn ngẫu nhiên $y \in N(x)$

a) tính $\Delta(x, y) = Obj(x) - Obj(y)$

b) **if** $\Delta(x, y) < 0$ **then** $p = 1$

c) **else** $p = e^{-\Delta(x, y)/T}$

d) **if** $rand[0,1] < p$ **then** $x \leftarrow y$

if $Obj(x) > Obj(x^*)$ **then** $x^* \leftarrow x$

2. giảm T theo sơ đồ C

return x^*

Sơ đồ làm lạnh C

► $T_t = T_0 * \alpha^{t*k}$

- $T_0 > 0$
- $\alpha \in (0,1)$
- $1 \leq t \leq m$
- $1 < k < m$

► Ý nghĩa

- t càng tăng thì T càng nhỏ, p càng nhỏ
- T lớn: chấp nhận bất cứ trạng thái nào
 - chuyển động ngẫu nhiên (random walk)
- T nhỏ: không chấp nhận trạng thái kém
 - leo đồi ngẫu nhiên



- ▶ Không có cơ sở lý thuyết rõ ràng
- ▶ Thường cho kết quả tốt hơn leo đồi
 - Ít bị cực trị địa phương
- ▶ Việc lựa chọn tham số phụ thuộc vào bài toán cụ thể