



Process 2 Presentation

**Subject: 503043 – Introduction to
Artificial Intelligence**

Team 9

Instructor: Msc Nguyễn Thành An

Group's member

MSSV	Full name	Email
52200168	Lý Tiểu Long	52200168@student.tdtu.edu.vn
52200151	Huỳnh Hoài Nam	52200151@student.tdtu.edu.vn
52200156	Lê Hồng Quang	52200156@student.tdtu.edu.vn
52200147	Giản Hoàng Huy	52200147@student.tdtu.edu.vn
52000620	Lý Tuấn An	52000620@student.tdtu.edu.vn

- Problem
- Random Restart Hill-Climbing
- Simulated Annealing Search
- Local Beam Search

Problem

Initial state

- Mỗi trạng thái của không gian chứa hai thuộc tính dạng số nguyên không âm là x và y .
- Ứng với mỗi trạng thái (x, y) có một giá trị evaluation function là $z \in [0, 255]$

Actions

- Chuyển động của trạng thái hiện tại bằng một trong các hành động đi qua Trái, Phải, Lên hoặc Xuống.
- Phụ thuộc vào vị trí của trạng thái hiện tại.

Transition Model

- Trả về trạng thái tiếp theo sau khi thực hiện hành động.

Goal state

- Tọa độ mang giá trị evaluation cao nhất mà trạng thái bắt đầu có thể di chuyển đến.

Path

- Danh sách chứa các bộ (x, y, z) từ trạng thái bắt đầu đến trạng thái kết quả mà thuật toán tìm ra.

Class Problem

<code>__init__(filename)</code>	→	Hàm khởi tạo Problem
<code>load_file(filename)</code>	→	Đọc dữ liệu cần thiết từ file hình ảnh
<code>get_evaluation(state)</code>	→	Giá trị Z tương ứng với trạng thái input
<code>make_random_state()</code>	<u>tuple (x, y)</u> →	Trạng thái ngẫu nhiên trong giới hạn kích thước của hình ảnh.
<code>global_maximum_test(state)</code>	<u>true/false</u> →	Kiểm tra trạng thái input có đạt được giá trị z cực bộ hay không
<code>get_neighbors(state)</code>	<u>list (neighbors)</u> →	Danh sách trạng thái hàng xóm của trạng thái input
<code>get_previous_state(state, direction)</code>	<u>tuple (x, y)</u> →	Trạng thái trước đó của trạng thái trạng thái input ứng với hành động L, U, R, D
<code>show()</code>	→	Trực quan hoá tất cả các bộ (x, y, z).
<code>draw_path(path)</code>	→	Vẽ đường đi trên mặt cong.

Class LocalSearchStrategy

random_restart_hill_climbing(problem, num_trial)

path

Algorithm
Random Restart Hill Climbing

schedule(t)

Giá trị không âm tượng trưng cho
nhiệt độ/năng lượng/động năng
còn lại

simulated_annealing_search(problem, schedule)

path

Algorithm
Simulated Annealing Search

local_beam_search(problem, k)

path

Algorithm
Local Beam Search

Pseudocode Random Restart Hill Climbing

Function random_restart_hill_climbing (problem, num_trial) **returns** best_path

best_evaluation \leftarrow MIN_VALUE

best_path \leftarrow an empty list

for $t = 1$ **to** num_trial **do**

 current_state \leftarrow random state

 path \leftarrow append (current_state with evaluation(current_state))

loop do

 neighbor \leftarrow get_neighbors(current_state)

 next_state \leftarrow max(neighbor)

if evaluation(current_state) **better** evaluation(next_state) **then**

break

 current_state \leftarrow next_state

 path \leftarrow INSERT(current_state with evaluation(current_state))

if evaluation(current_state) **better** best_evaluation **then**

 best_evaluation \leftarrow evaluation(current_state)

 best_path \leftarrow path

Pseudocode Simulated Annealing Search

Function simulated_annealing_search (problem, schedule) **returns** path

```
current_state ← problem.RANDOM_STATE()
path ← INSERT(current_state, evaluation(current_state))
t ← 1
loop do
    T ← SCHEDULE(t)
    if T < 0 then return path
    neighbors ← get_neighbors(current_state)
    next_state ← RANDOM(neighbors)
    ΔE ← next.VALUE – current.VALUE
    if ΔE > 0 then current ← next
    else current ← next with probability  $e^{\Delta E/T}$ 
    t ← t + 1
```


Pseudocode Local Beam Search

Function local_beam_search(problem, k) **returns** best_path

k_states \leftarrow INSERT(k random state)

iteration \leftarrow 0

litmit \leftarrow 500

loop do

next_successors \leftarrow an empty list

for each state **in** k_states

next_successors \leftarrow INSERT(get_neighbors(state))

k_states \leftarrow k best state in next_successors

if global_maximum_test(k_states[0]) or iteration = litmit **then** break

iteration \leftarrow iteration + 1

best_path \leftarrow the best state in k_states

1. Lý Tuấn An

- Email: 52000602@student.tdtu.edu.vn
- Công việc: Câu 1, câu 4, làm slide thuyết trình.
- Mức độ hoàn thành: Hoàn thành.

2. Lý Tiểu Long

- Email: 52200168@student.tdtu.edu.vn
- Công việc: Câu 1, câu 2, làm slide thuyết trình.
- Mức độ hoàn thành: Hoàn thành.

3. Lê Hồng Quang

- Email: 52200156@student.tdtu.edu.vn
- Công việc: Câu 1, câu 2, làm slide thuyết trình.
- Mức độ hoàn thành: Hoàn thành.

4. Huỳnh Hoài Nam

- Email: 52200151@student.tdtu.edu.vn
- Công việc: Câu 3, câu 4, làm slide thuyết trình.
- Mức độ hoàn thành: Hoàn thành.

5. Giản Hoàng Huy

- Email: 52200147@student.tdtu.edu.vn
- Công việc: Câu 3, câu 4 quay video thuyết trình.
- Mức độ hoàn thành: Hoàn thành.

Thuận lợi và khó khăn

- **Thuận lợi:**

- + Có nhiều nguồn tài liệu tham khảo về các thuật toán Hill-Climbing, Simulated-Annealing và Local Beam.
- + Được cung cấp đầy đủ kiến thức để hoàn thiện báo cáo và hoàn thành công việc. Cùng với đó là sự hỗ trợ nhiệt tình từ thầy và các bạn.

- **Khó khăn:**

- + Khó khăn trong việc tìm hiểu và giải thích các thuật toán phức tạp, cụ thể là Local Beam
- + Bất lợi trong việc triển khai và tinh chỉnh các thuật toán trong môi trường thực tế.

Đánh giá mức độ hoàn thành các yêu cầu

Câu 1: Problem	Đã hoàn thành
Câu 2: Random Restart Hill-Climbing	Đã hoàn thành
Câu 3: Simulated Annealing Search	Đã hoàn thành
Câu 4: Local Beam Search	Đã hoàn thành

[1] Từ Minh Phương, Giải quyết vấn đề bằng tìm kiếm, *Giáo trình Nhập môn trí tuệ nhân tạo*, Hà Nội, 2014, 22.

[2] GS. TSKH Hoàng Kiếm, ThS. Đình Nguyễn Anh Dũng, *Giáo Trình Nhập Môn Trí Tuệ Nhân Tạo*, NXB Đại học Quốc gia 2005, Đại học Quốc gia TP.HCM, Trường Đại học Công nghệ thông tin.



**Subject: 503043 – Introduction to
Artificial Intelligence**

**Thanks for
watching and listening**
