THUYẾT TRÌNH – QT2

Môn: Nhập môn Trí tuệ Nhân tạo

Thời gian làm bài: 05 tuần

I. Hình thức

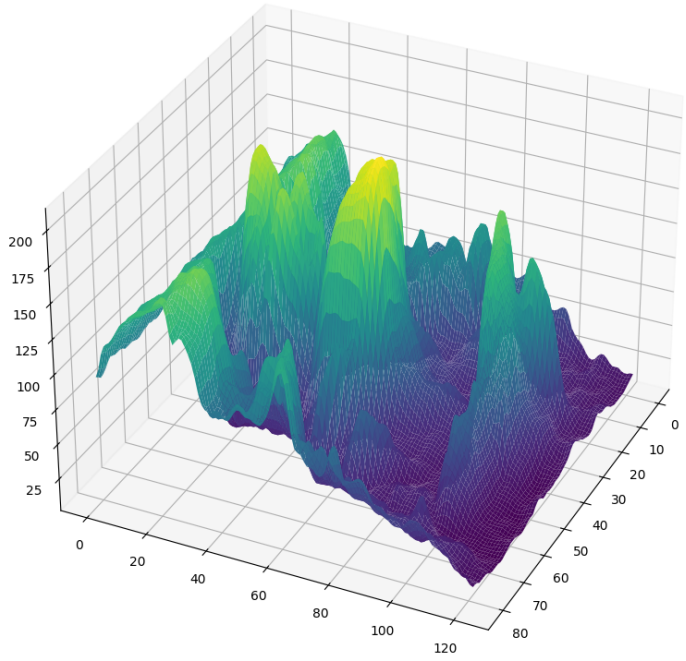
- Đề tài được thực hiện theo nhóm **04** – **05** sinh viên.
- Nhóm sinh viên thực hiện các yêu cầu và nộp bài theo hướng dẫn chi tiết bên dưới.

II. Yêu cầu

Cho thư mục **LocalSearch** chứa bên trong

- 'monalisa.jpg': ảnh dùng để phát sinh không gian trạng thái
- 'viz3d.py': tệp mã nguồn chứa các ví dụ về load không gian trạng thái, trực quan hoá không gian và vẽ đường với thư viện matplotlib.

Trong bài toán này, 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 $z \in [0, 255]$. Giá trị z càng lớn thì trạng thái càng tốt. Trực quan hoá tất cả các bộ số (x, y, z), ta có mặt cong như hình sau.



a) Câu 1 (2.0 điểm): Problem

Nhóm sinh viên tiến hành mô hình hoá bài toán tìm kiếm, xác định các thông tin cần thiết để xây dựng thành lớp đối tượng Problem trong tệp problem.py. Các thuộc tính và hàm có liên quan tuân theo nội dung lý thuyết đã học.

Lưu ý, mỗi trạng thái có hai thuộc tính ứng với giá trị x và y. Giá trị evaluation của trạng thái ứng với giá trị z. Sinh viên sử dụng mã nguồn được cho trong viz3d.py để hỗ trợ thao tác đọc dữ liệu.

Lớp Problem có hàm show() để trực quan hoá tất cả các bộ (x, y, z) và hàm draw_path(path) để vẽ đường đi trên mặt cong. Sinh viên tham khảo mã nguồn trong viz3d.py để cài đặt.

b) Câu 2 (2.0 điểm): Random Restart Hill-Climbing

Cài đặt lớp **LocalSearchStrategy** trong search.py và xây dựng phương thức

random_restart_hill_climbing(problem, num_trial) → path

- Tham số:
  - **problem**: một object của lớp Problem (câu 1) cung cấp thông tin
  - **num_trial**: số nguyên dương, số lần chạy của thuật toán.
- Trả về:
  - **path**: danh sách (list) chứa các bộ (x, y, z) từ trạng thái bắt đầu đến trạng thái kết quả mà thuật toán tìm ra.
- Xử lý: thực hiện tìm kiếm theo thuật toán Random Restart Hill Climbing đã học.
- Vận hành: viết đoạn chương trình để chạy hàm vừa cài đặt trong test.py, sau đó trực quan hoá kết quả bằng biểu đồ chứa mặt cong và đường đi.

c) Câu 3 (2.0 điểm): Simulated Annealing Search

Cài đặt lớp **LocalSearchStrategy** trong search.py và xây dựng phương thức

simulated_annealing_search(problem, schedule) → path

- Tham số:
  - **problem**: một object của lớp Problem (câu 1) cung cấp thông tin
  - **schedule**: một hàm số nhận vào điểm thời gian ***t*** và trả về giá trị không âm tượng trưng cho nhiệt độ/năng lượng/động năng còn lại.
- Trả về:

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

- Xử lý: thực hiện tìm kiếm theo thuật toán Simulated Annealing Search đã học.
- Vận hành: viết đoạn chương trình để chạy hàm vừa cài đặt trong test.py, sau đó trực quan hoá kết quả bằng biểu đồ chứa mặt cong và đường đi.

d) Câu 4 (2.0 điểm): Local Beam Search

Cài đặt lớp **LocalSearchStrategy** trong search.py và xây dựng phương thức

local_beam_search(problem, k) → path

- Tham số:
  - **problem**: một object của lớp Problem (câu 1) cung cấp thông tin
  - **k**: số nguyên dương, số lượng trạng thái tối đa được duy trì tại mỗi bước thuật toán.
- Trả về:
  - **path**: danh sách (list) chứa các bộ (x, y, z) từ trạng thái bắt đầu đến trạng thái kết quả mà thuật toán tìm ra. Lưu ý chỉ giữ lại một đường đi duy nhất tới trạng thái kết quả.
- Xử lý: thực hiện tìm kiếm theo thuật toán Local Beam Search đã học.
- Vận hành: viết đoạn chương trình để chạy hàm vừa cài đặt trong test.py, sau đó trực quan hoá kết quả bằng biểu đồ chứa mặt cong và đường đi.

e) Thuyết trình (2.0 điểm)

- Sinh viên viết báo cáo kết quả đồ và quay video thuyết trình. **KHÔNG CÓ MẪU THUYẾT TRÌNH, NHÓM SINH VIÊN TỰ TỔ CHỨC NỘI DUNG**.
- Các thông tin tối thiểu cần có.
  - Danh sách sinh viên: MSSV, Họ tên, Email, Phân công công việc, Mức độ hoàn thành.
  - Tóm tắt cách xử lý từng yêu cầu, nên diễn đạt bằng mã giả/sơ đồ.
  - HẠN CHẾ TỐI ĐA NHÚNG MÃ NGUỒN THÔ VÀO BÀI THUYẾT TRÌNH.
  - Các nội dụng tìm hiểu cần trình bày cô đọng, có ví dụ trực quan.

  - Thuận lợi và khó khăn trong đề tài.
  - Bảng tự đánh giá mức độ hoàn thành các yêu cầu.
  - Tài liệu trích dẫn ghi theo định dạng IEEE.
- Yêu cầu về định dạng: tỷ lệ slide 4x3, hạn chế dùng nền tối/màu sắc vì máy chiếu mờ, đảm bảo khi in bài thuyết trình dạng trắng đen thì các nội dung vẫn rõ ràng.
- Thời lượng tối đa cho phần thuyết trình là **10 phút**.

III. Hướng dẫn nộp bài

- Tạo thư mục với tên theo cú pháp

  QT2_<Mã nhóm>

  trong đó gồm:
  - **source:** thư mục mã nguồn chứa tệp .ipynb hoặc .py
  - **presentation.pdf:** bài thuyết trình.
- Nén thư mục thành tệp zip và nộp theo deadline.

IV. Quy định

- **Nhóm sinh viên nộp trễ hạn bị 0.0 điểm toàn nhóm.**
- **Sai sót mã số sinh viên nào thì sinh viên tương ứng bị 0.0 điểm.**
- **Thiếu sót các tài liệu được yêu cầu trong tệp nộp bài sẽ bị trừ tối thiểu 50% điểm phần thuyết trình.**
- **Mọi hành vi sao chép code trên mạng, chép bài bạn hoặc cho bạn chép bài nếu bị phát hiện đều sẽ bị điểm 0.0.**
- **Nếu bài làm của sinh viên có dấu hiệu sao chép trên mạng hoặc sao chép nhau, sinh viên sẽ được gọi lên phỏng vấn code riêng để chứng minh bài làm là của mình.**

-- HẾT --