Ton Duc Thang University
Faculty of Information Technology

PRESENTATION – PROCESS II

Course: Introduction to Artificial Intelligence

Duration: 05 weeks

I. Formation

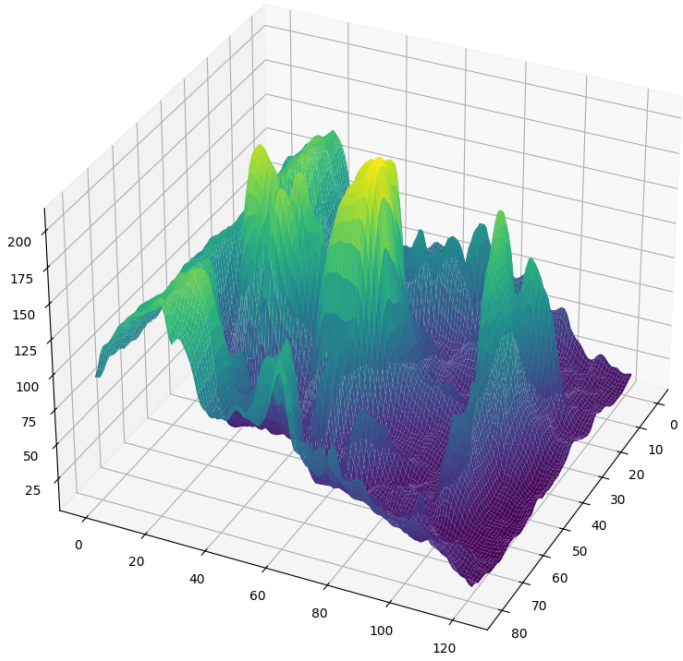
- The presentation is conducted in groups of 04 – 05 students.
- Student groups conduct required tasks and submit the project following instructions.

II. Requirements

Given the project folder **LocalSearch** consisting of

- 'monalisa.jpg': an image for generating the state space.
- 'viz3d.py': a source code file with examples for loading state spaces, visualizing spaces, and drawing moving paths using the matplotlib library.

In this problem, each state of the space contains two non-negative integer attributes, x and y. Corresponding to each state (x, y) is an evaluation function value, $z \in [0, 255]$. The larger the value of z, the better the state. Visualizing all sets of numbers (x, y, z), we have a curve as shown below.

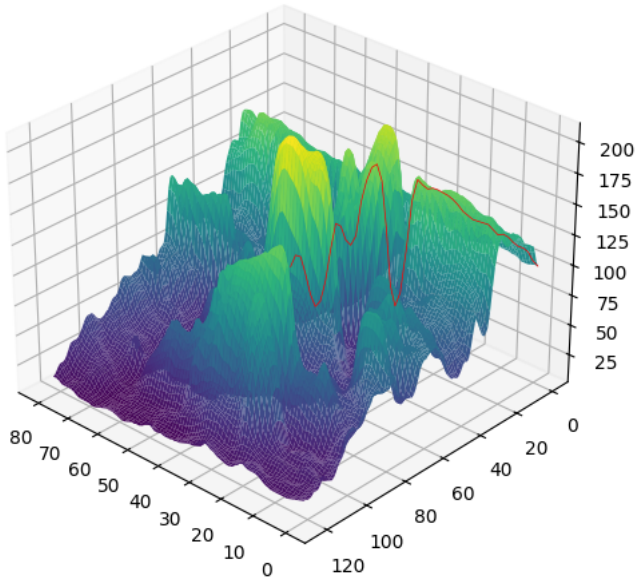


a) Task 1 (2.0 points): Problem formulation

The student group formulate the given search problem, identifying the necessary information to construct the Problem class in the problem.py file. Attributes and methods follow theorical principles in the lectures.

Note that each state has two attributes corresponding to the values of x and y. The evaluation value of the state corresponds to the value of z. Students utilize the source code provided in viz3d.py to implement data loading operations.

The Problem class includes a show() function to visualize all tuples of (x, y, z) and a draw_path(path) function to draw the path on the curved surface. Students refer to the source code in viz3d.py for implementation.



b) Task 2 (2.0 points): Random Restart Hill-Climbing

Implement the **LocalSearchStrategy** class in search.py with the method below

random_restart_hill_climbing(problem, num_trial) → path

- Params:
  - **problem**: an instance of the class Problem (task 1) for information providing
  - **num_trial**: a positive integer, the number of trial of restarting the algorithm.
- Returns:
  - **path**: a list of tuples (x, y, z) representing the path from the initial state to the resulting state.

- Operations: regarding to the Random Restart Hill Climbing algorithm.
- Execution: implement a brief program, in test.py, to execute the method. Then, visualize the results with the curved surface and the found path.

c) Task 3 (2.0 points): Simulated Annealing Search

Implement the **LocalSearchStrategy** class in search.py with the method below

simulated_annealing_search(problem, schedule) → path

- Params:
  - **problem**: an instance of the class Problem (task 1) for information providing
  - **schedule**: a function taking a time step ***t*** and returning a non-negative value corresponding to the temperature/energy at that time step.
- Returns:
  - **path**: a list of tuples (x, y, z) representing the path from the initial state to the resulting state.
- Operations: regarding to the Simulated Annealing Search algorithm.
- Execution: implement a brief program, in test.py, to execute the method. Then, visualize the results with the curved surface and the found path.

d) Task 4 (2.0 points): Local Beam Search

Implement the **LocalSearchStrategy** class in search.py with the method below

local_beam_search(problem, k) → path

- Params:
  - **problem**: an instance of the class Problem (task 1) for information providing
  - **k**: a positive integer, the maximal number of states maintained at each step of the algorithm.
- Returns:
  - **path**: a list of tuples (x, y, z) representing the path from the initial state to the resulting state. Note to maintain only the path to the resulting state.
- Operations: regarding to the Local Beam Search algorithm.
- Execution: implement a brief program, in test.py, to execute the method. Then, visualize the results with the curved surface and the found path.

e) Presentation (2.0 points)

- Student groups compose a presentation to report your work.
- **THERE IS NO PRESENTATION TEMPLATES. STUDENTS ARANGE CONTENTS IN A LOGICAL LAYOUT BY YOURSELVES.**
- The presentation must include below contents
  - Student list: Student ID, Full name, Email, Assigned tasks, Complete percentage.
  - Briefly present approaches to solve tasks, should make use of pseudo code/diagrams.
  - AVOID EMBEDDING RAW SOURCE CODE IN THE PRESENTATION.
  - Study topics are introduced briefly with practical examples.
  - Advantages versus disadvantages
  - A table of complete percentages for each task.
  - References are presented in IEEE format.
- **Format requirements:** slide ratio of 4x3, avoid using dark background/colorful shapes because of projector quality, students ensure contents are clear enough when printing the presentation in grayscale.
- Presentation duration is **10 minutes**.

III. Submission Instructions

- Create a folder whose name is as

  QT2_<Group ID>

- Content:
  - **source** → source code folder, consisting of .py/.ipynb files
  - **presentation.pdf** → presentation.
- Compress the folder to a zip file and submit by the deadline.

IV. Policy

- **Student groups submitting late get 0.0 points for each member.**
- **Missing required materials in the submission loses at least 50% points of the presentation.**

- **Copying source code on the internet/other students, sharing your work with other groups, etc. cause 0.0 points for all related groups.**
- **If there exist any signs of illegal copying or sharing of the assignment, then extra interviews are conducted to verify student groups' work.**

-- THE END --