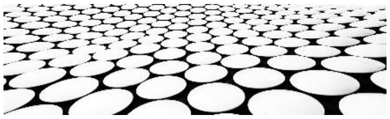
Caroline Barrière, Fall 2023

CSI4106 Introduction to Artificial Intelligence



ASSIGNMENT 1

Knapsack

Greedy Search, Simulated Annealing and Genetic algorithm

GOALS

The overall goal of this assignment is to test different search algorithms on the classic problem of knapsack. Also, the goal is to get familiar with the experimental set-up required to compare 3 search algorithms, and report comparative results.

At the end of this project, you will have:

- Reviewed your Python skills, as the project MUST be done in Python
- Explored Kaggle, an amazing resource of challenges and datasets
- Programmed 3 search algorithms: Greedy Search, Simulated Annealing, Genetic algorithm
- Performed an empirical, comparative study, using the programmed algorithms and dataset
- Documented, in a Jupyter Notebook, everything about your empirical study (view the Specific Requirements section), in a way to make your experiment understandable and reproducible.

SUBMISSION INFORMATION

- Deadline:
  - Submission of link to your notebook: **Sunday, October 1rst, midnight**
  - Your notebook MUST NOT be modified following your submission
- Groups:
  - You are expected to form groups of 2 and do a single submission per group. You first need to register your group in Brightspace to later be able to do a group submission.
  - If you prefer to work alone, that is fine, but the requirements are not changed.
- Where to submit:
  - Your submission must be done in Brightspace in Assignment section (Assignment 1)
- Submission format:
  - No files accepted.
  - Your submission MUST be a **link** to a Colab Jupyter Notebook that the corrector will be able to go through (and run the code cells). If you prefer a different platform than Colab, that is fine, but the corrector MUST be able to access your notebook without having to install anything or copy any data.

PLEASE NOTE:

If the corrector cannot access your notebook, or cannot run your code, the mark will be zero. It is your responsibility to test if your submission link works from a computer different than yours, as well as test that the cells in your Notebook are executable. The most frequent problem is that the dataset is not readable (file not found). Make sure you follow the instructions provide in the notebook template to put your data at the right place. You CANNOT submit a notebook file in Brightspace that the corrector would need to download, you must submit a link to a web-accessible notebook, ready to run.

TUTORIALS/TECHNOLOGIES

To achieve this assignment, you need to explore different environments. As this is a 4th year course, and you all have programming experience, you are mature enough to perform the exploration by yourself. If you “get stuck”… don’t hesitate to post your questions on the Brightspace forum in the Assignments section.

To help you, Baharin (your TA) has suggested the many links below.

What you need to know:

1. **Python**: The project MUST be done using the Python programming language. There are many tutorials that you can use to familiarize yourself with Python.

    - Python Tutorial for Beginners https://www.youtube.com/watch?v=t8pPdKYpowI
    - Python Full Course for Beginners https://www.youtube.com/watch?v=_uQrJ0TkZlc

2. **Jupyter Notebook**: Your project will have to submitted as a Jupyter Notebook. You can create/run such notebooks using Colab (see point 4).

    - Jupyter Notebook tutorial (Windows) https://www.youtube.com/watch?v=2WL-XTl2QYI
    - Jupyter Notebook tutorial (Mac) https://www.youtube.com/watch?v=HW29067qVWk

3. **Colab** https://colab.research.google.com for the creation of your final report (as a Jupyter Notebook) and to access machines that may be faster than yours for training classifiers. You do not have to train your models on Colab, you can work on your local machine, but... the training might be long.

    - Google Colab Tutorial https://www.tutorialspoint.com/google_colab/index.htm

REQUIREMENTS

1. Use The Provided Template

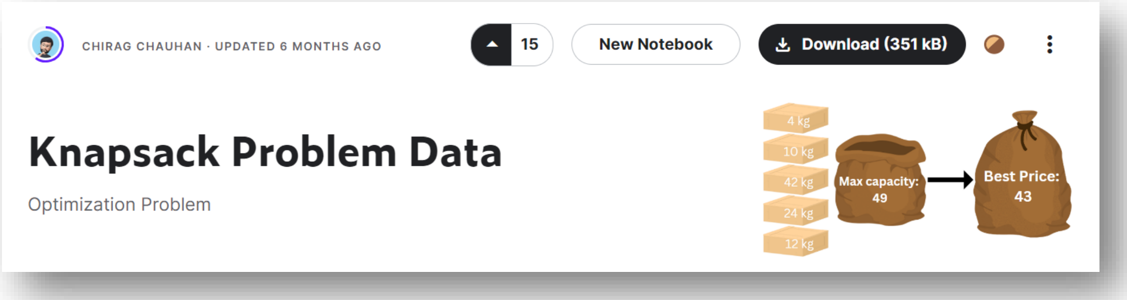
To help you to get started with the first assignment, your TA has prepared a template of a Jupyter Notebook. You can find this template on Brightspace under the assignment 1 description. Open the template on https://colab.research.google.com/ (click on open notebook from file tab).

2. Download Kaggle Knapsack Dataset

The Kaggle site is a very interesting site to explore as it contains datasets for many tasks in Artificial Intelligence.

There is one dataset for the Knapsack problem that you must use. It is a zip file and comes with two datasets:

https://www.kaggle.com/datasets/warcoder/knapsack-problem?resource=download



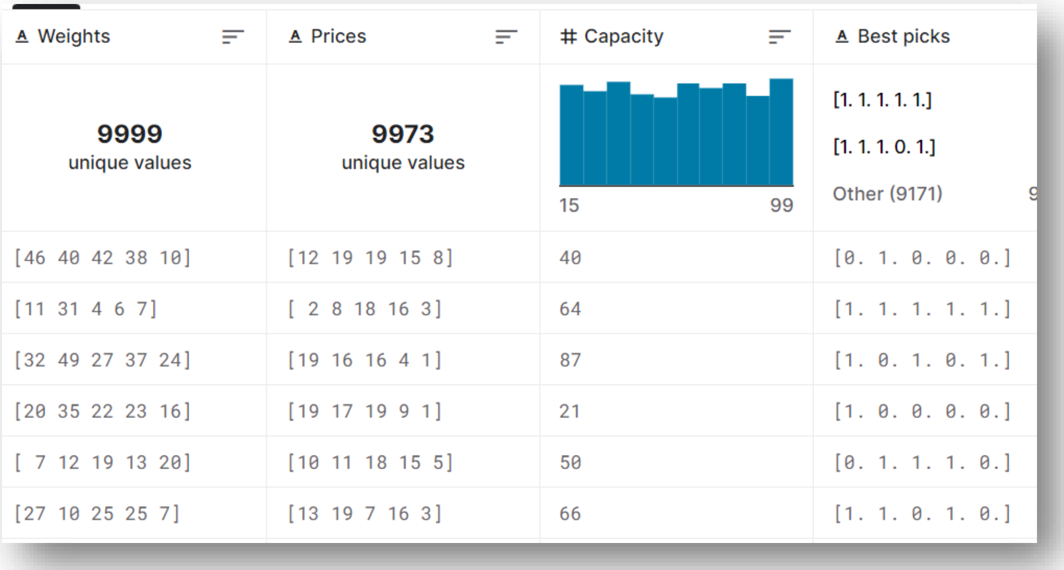
CHIRAG CHAUHAN · UPDATED 6 MONTHS AGO

Knapsack Problem Data

Optimization Problem

Use the “knapsack_5_items.csv” dataset which contains many instances of the Knapsack problem, providing weights, costs, capacity and best picks (expected good answer) for all the instances. Knowing the expected answers will allow you to evaluate your algorithms.

Note: Please note that the preprocessing step in the provided notebook template assumes that you downloaded this dataset (knapsack_5_items.csv).



Weights	Prices	Capacity	Best picks
9999 unique values	9973 unique values	15 … 99	[1. 1. 1. 1. 1.] [1. 1. 1. 0. 1.] Other (9171)
[46 40 42 38 10]	[12 19 19 15 8]	40	[0. 1. 0. 0. 0.]
[11 31 4 6 7]	[2 8 18 16 3]	64	[1. 1. 1. 1. 1.]
[32 49 27 37 24]	[19 16 16 4 1]	87	[1. 0. 1. 0. 1.]
[20 35 22 23 16]	[19 17 19 9 1]	21	[1. 0. 0. 0. 0.]
[7 12 19 13 20]	[10 11 18 15 5]	50	[0. 1. 1. 1. 0.]
[27 10 25 25 7]	[13 19 7 16 3]	66	[1. 1. 0. 1. 0.]

3. Implement Generate and Test

Often, for large problems, it is impossible to program a “generate and test” approach which exhaustively explores the full solution space to find the optimal solution.

In our current case, although the dataset contains a lot of knapsack problem instances, each instance is with only 5 objects, so a maximum of 2^5 (32) possibilities for each.

- **CODING:** Program the “generate and test” approach.
- **RESULTS / ANALYSIS:** Use it to find the optimal solution for each problem instance. Your results of optimal solutions for each knapsack instance will allow you to validate if the provided dataset on Kaggle is trustable. Does the optimal solution you find matches what is provided as “best pick” by the authors?

4. Implement 3 search algorithms

We have explored different types of search algorithms in class. In this assignment, you will get hands-on experience with these algorithms by implementing them and testing them.

You must implement 3 algorithms:

a) Greedy Search

- **CODING:** Define a greedy strategy and code it.
- **RESULTS / ANALYSIS:** Run your search on the dataset and find the recall (number of optimal solutions found). Is the greedy search providing interesting results??

b) Simulated Annealing (Random Modification Algorithm)

- **CODING:** Program a simulated annealing algorithm. We saw the algorithm in class. The modification, for a knapsack, is simply a change from 0 to 1 (or 1 to 0) for an object being included or not in the bag. And the fitness function is the sum of the values of the objects included in the bag. If a change makes the solution a non-satisfiable solution (exceeds the bag capacity), its fitness can be considered zero.
- **RESULTS / ANALYSIS:** Run your search on the dataset and find the recall (number of optimal solutions found). The simulated annealing requires a Temperature change schedule with a cooling rate. Explore at least two other values for cooling rate than the one provided in the notebook template. What is the impact of it on results? You can also explore changing other parameters.

c) Genetic Algorithm (Population-Based algorithm)

- **CODING:** Program a genetic algorithm. Make sure to include crossovers and mutations**.** Also use Tournament Selection to select the subset of the populations on which the crossovers and mutations will be applied. Mutations can be defined the same way as modifications in the Simulated Annealing algorithm. Or you can define other types of mutations. For crossover, you can use one-point crossover. The point can be fixed or variable. You can also explore other types of crossovers. Explore different rates (probabilities) for crossover and mutation. Typically, the default rate of crossover is 0.7 and the rate of mutation is 0.1.
- **RESULTS / ANALYSIS:** Run your search on the dataset and find the recall (number of optimal solutions found). Explore at least two other rates (one lower and one greater than the default rates) for crossover and mutation. What is the impact of changing rates of crossover and mutation?

5. <u>Do a comparative study of the 3 algorithms with their parameters</u>

Using the dataset, which contains the actual best picks for each knapsack problem instance, you can evaluate how well each of your algorithms is doing.

For each one, measure the recall (number of problems for which the algorithm get the correct answer). Show a comparison on a chart (or other way to visualize easily).

For both the simulated annealing and the population-based algorithm, there are various parameters to be set (e.g. Temperature, Stopping Conditions). Show the impact of varying these parameters.

6. Document your empirical study in a Jupyter Notebook

The purpose of the report is to illustrate the whole process followed during this assignment. Your Jupyter Notebook should include:

1. Group number, names and student numbers of group members
2. A description of the problem tackled.
3. A description of the dataset used with references.
4. A description + python code of the "Generate and Test" with an analysis of the results compared to the dataset provided.
5. A description + python code of the Greedy algorithm. Describe your strategy.
6. A description + python code of the Simulated annealing algorithm.
7. A description + python code of the Genetic algorithm.
8. A comparative result section, with figures (or tables) + explanations.
9. Conclusion
10. References

Note: Although some fields/sections are provided by your TA in the template, it is your responsibility to make sure that you cover all the requested sections in your notebook.

So, overall, if a section requires a Python code, add the Python code to a cell. If it requires an explanation or results, add them to a cell as well. Therefore, for each section, there will be either a python code (if it is a programming section), or an explanation/result cell.

☆☆☆

EVALUATION

- Overall effort in the report (20%)
    - Writing in a clear and descriptive style that will allow the corrector to easily read/understand what was done, how and why
    - Good cell separation (text, code, results, etc)
    - Tests on various examples easy to perform by the corrector
    - Comparison between the approaches easy to understand (visualization using tables and/or graphs)
    - Well commented code overall
    - Report detailed enough for reproducibility
- Generate and Test (10%)
    - Code + result analysis.
- Greedy Search (10%)
    - Code + explanation of strategy
- Simulated Annealing (20%)
    - Code well implemented with comments to show understanding
- Genetic algorithm (20%)
    - Code well implemented with comments to show understanding
- Result analysis (20%)
    - Presentation of comparative results with a good analysis
    - Inclusion of variations of different parameters and analysis of impact
- References (should be present, -10% if not)
    - For any part of your code taken from a web site (even a tutorial site or stackoverflow), you must provide the reference to it.
    - Any theory/algorithms found in books, slides, tutorials that you used should be referenced.

QUESTIONS

- You can ask your questions within the assignment topic of the discussion forum on Brightspace.
- You can also send an email to Baharin (balia034@uottawa.ca), but using the forum is a much preferred way as fellow students will benefit from your questions and Baharin's answers.