Artificial Intelligence Methods (COMP2051 or AE2AIM)

Dr. Xinan Chen Spring 2024

Labs 2: Heuristics and Local Search

In this session, you shall practice in using very simple heuristics and a local search to solve the knapsack problem .

Knapsack problem

Given a knapsack of capacity *C* and a set of *n* items, each item *i* has a volume v_i and a value p_i. Determine the list of items to be packed so that the total value is maximized.

$$\max \sum_{i=1}^{n} p_i x_i$$

Subject to:

$$\sum_{i=1}^{n} v_i x_i \leq C$$

$$x_i = \{0,1\}$$

1. Generate a solution heuristically

In this task you are asked to generate a feasible solution based on some simple heuristics that you can think of. Discussions with your peers are encouraged before you decide what heuristics to use. Here are two possible ways to generate hints:

- **Generating solution from LP rounding.** Given the results of x_i returned from the LP solver (Lab 1), you can round non-integer variables to their nearest integers (binary in this case). If this leads to an infeasible solution (because of the violation of the capacity constraint caused by rounding), you can adjust (increase) the default rounding threshold (0.5) until you are able to obtain a feasible solution.
- **Generating solution by a simple heuristic of intuition.** For this problem, a good greedy heuristic will be trying to include items with the highest possible profit-volume ratio (p_i/v_i). To do this, you need to sort all items decreasingly according to p_i/v_i and then iteratively consider packing the item from the top of the sorted list until either no item can be packed in the knapsack anymore (e.g. zero capacity left) or all items have been considered. (our example code used this method)

2. Local search - discussions

Based on the initial solution from Section 1, discuss with your peers how a local search method could improve the initial solution further. Please work out all the components of Local Search method listed in the lecture slides. In particular, you should consider carefully the data structure for storing candidate solutions. With this method, would you get the optimal solution? Why?

3. Local search – Best Descent Method

In this lab, you will work with the knapsack problem and explore different local search techniques to find an optimal solution. The initial solution is generated using the Greedy method, which selects items based on their value-to-weight ratio, and then the First Decent method is applied to improve the solution. The task is to modify the code to implement the Best Decent method, which refines the solution by exploring all neighboring solutions and selecting the one with the best improvement. After modifying the code, you will need to answer a set of questions on Moodle, reflecting on the differences between the methods and their performance.

The program can be compiled and run using the following command:

```
gcc lab02.c; ./a.out
```