Week 9 Computer Omptimization

MSc Artificial Intelligence and Data Analytics, UoM

Construction of simple heuristic for an initial solution in 0-1 Knapsack Problem.

Idea of the heuristic:

Our goal is to put as many items inside the sack by depending only on the capacity, we keep the items that satisfy this condition (*total_weight* <= *capacity*) and we try to **maximize** the number of items.

The Optimal solution would be to put all items inside the sack, that means ***solution = [1,..,1]***, where ***len(solution)=len(weights_list)***.

Then we check constantly the total sum with the capacity, and if the total sum is greater, *we find the max weight in the list and remove it*, by turning the index of that item inside solution list to 0. And we repeat this untill the condition (*total_weight* <= *capacity*) is true.

This idea is implemented inside :

> **def** *init_heuristic_solutionW(capacity,weights)*

Then we calculate the total profit from the items that are inside the sack.

> **def** *calculate_total_profit(capacity,weights)*

Testing for different knapsack problem data:

See lines: **106-108** inside --**knap_heuristic_init.py**--, comment in/out the preferred line to load the data you want.

Results:

Test for the problem1 showed that we find the best solution, so we have a success.

Test for problem2 showed that our solution gives a **tota_profit = 1249** while the optimal solution, according the input data, would give **tota_profit = 1458**, meaning that we are close.

We can test even more, and see that this heuristic is a good approach as a starting solution for the 0-1 knapsack problem.

Execution:

Use cmd like this,

> python3 knap_heuristic_init.py

Author:

Tassos Karageorgiadis

| *December 2020* |