

Algorithmic Operation Research

Homework 6

Theodora Panagea - 1115201400135
Anna-Aikaterini Kavvada - 1115201500050

23-1-2020

Exercise 1

Define the 0-1 knapsack problem.

To begin with, in 0–1 knapsack problem we have the primal restriction of the number x_i of copies of each item to zero or one, since we cannot intersect the items available and get just part of them.

Now, consider a set of n items, numbered from 1 up to n . Each item has a weight w_i and a value v_i . Also, for the problem, consider our knapsack has a maximum weight capacity W . Our goal, is to maximize the the sum of the item values in the knapsack, so that the weights is less than or equal to the maximum capacity W . Thus, we get the following problem expression:

$$\begin{aligned} & \text{maximize } \sum_{i=1}^n v_i x_i \\ & \text{subject to } \sum_{i=1}^n w_i x_i \leq W \quad \text{and } 0 \leq x_i \leq 1 \end{aligned}$$

Exercise 2

Give a dynamic programming solution to the 0-1 knapsack problem.

Exercise 3

Give real world applications of knapsack problem.

The most simple and obvious real life application of the knapsack problem is the following. Consider being a traveller/nomad in todays world. This means that the total weight of all your wordly possesions must fall under airline cabin baggage weight limits - usually 10kg. On some smaller airlines, however, this weight limit drops to 7kg. Occasionally, you have to decide not to bring something with you to adjust to the smaller weight limit.

So you have to decide what to leave behind (or get rid of altogether), which entails laying out all your belongings and choosing which ones to keep. That decision is based on the item's usefulness to you (/cost) and its weight, so that i can pack as more items as possible in the limited space you get.

The knapsack problem is an example of a combinational optimization problem, a topic in mathematics and computer science about finding the optimal object among a set of objects. This is a problem that has been studied for more than a century and is a commonly used example problem in combinatorial optimization, where there is a need for an optimal object or finite solution where an exhaustive search is not possible. The problem can be found real-world scenarios like resource allocation in financial constraints or even in selecting investments and portfolios. It also can be found in fields such as applied mathematics, complexity theory, cryptography, combinatorics and computer science. It is easily the most important problem in logistics.

In the knapsack problem, the given items have two attributes at minimum - an item's value, which affects its importance, and an item's weight or volume, which is its limitation aspect. Since an exhaustive search is not possible, one can break the problems into smaller sub-problems and run it recursively. This is called an optimal sub-structure. This deals with only one item at a time and the current weight still available in the knapsack. The problem solver only needs to decide whether to take the item or not based on the weight that can still be accepted. However, if it is a program, re-computation is not independent and would cause problems. This is where dynamic programming techniques can be applied. Solutions to each sub-problem are stored so that the computation would only need to happen once. Some other applications of the knapsack algorithm can be found in

1. Home Energy Management

2. Cognitive Radio Networks
3. Resource management in software
4. Large-scale multi-period precedence constrained knapsack problem: A mining application
5. Relay selection in secure cooperative wireless communication
6. Power allocation management
7. Selecting adverts garden city radio
8. Solve the production planning problem
9. Cognitive radio networks
10. 5G mobile edge computing
11. Selection of renovation actions
12. Solve the production planning problem
13. Sensor Selection in Distributed Multiple-Radar
14. Architectures for Localization
15. Appliance Scheduling Optimization for Demand Response
16. Adaptive Variable Density Sampling
17. Secure Cooperative Wireless Communication
18. Optimizing Power Allocation to Electrical Appliances
19. Computation Offloading in Wireless Multi-Hop Networks
20. Formulation and Solution Method of Tour Conducting
21. Plastic Bags Waste Management Using the Knapsack Model
22. Workflow mapping using CUDA
23. Optimization of Content Delivery Networks

24. Network Selection for Mobile Nodes

Another application is the one described below.

Mapping and running jobs on suitable resources are the core tasks in grid computing. In the algorithm to map light communication Grid-based workflow within the SLA context, there is an operation of resolving the conflict period which is exact a 0/1 knapsack problem. When the size of the workflow is large such as in the case of mapping a group of workflows, the time to solve this problem is long and thus, makes the whole mapping process long. In this paper, we describe a way to solve this problem by exploiting the parallel computing power of graphic processing unit (GPU) with compute unified device architecture (CUDA). The experiment shows that the approach is very efficient with huge problem.

Link to paper: *Light Communication SLA-Based Workflow Mapping Using CUDA*

Exercise 4

Define the Subset sum problem and give a dynamic programming solution for it. Write down the difference between the subset sum and the knapsack problem.