Optimization

Guttag Chapter 14

Goals

Understand Knapsack Problem

- Values
- Weights
- Constraint
- Optimization Objective

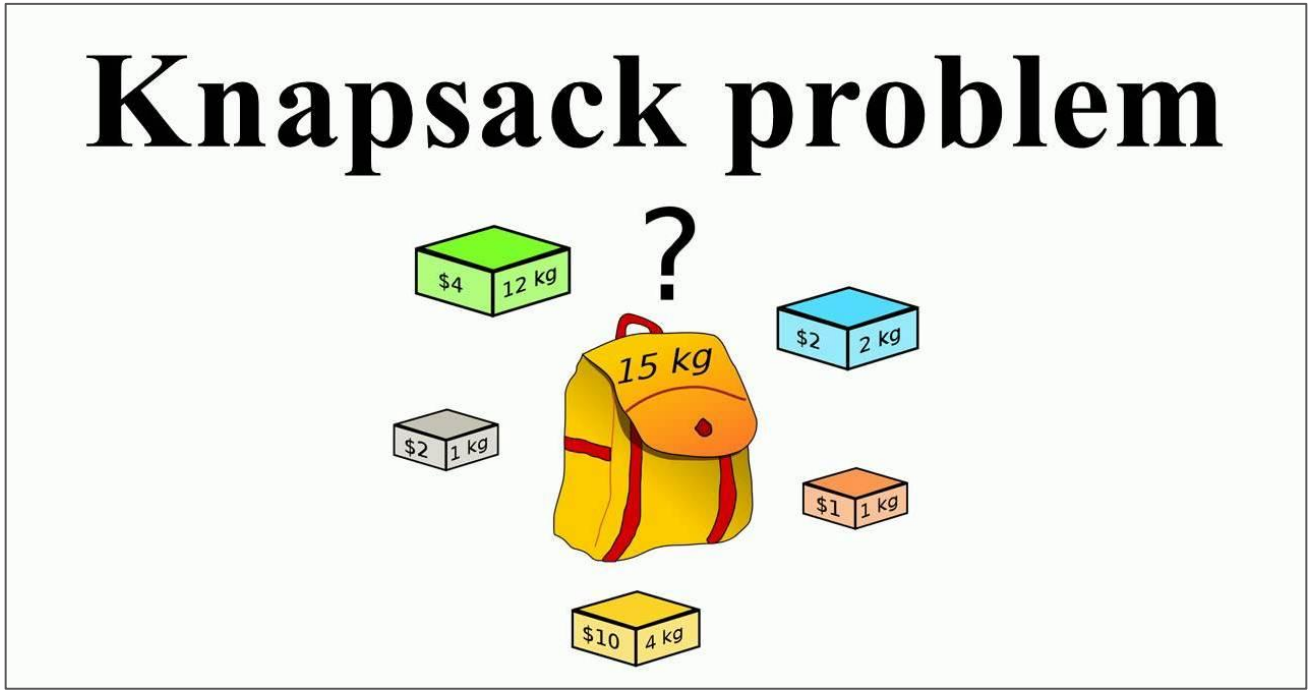
Powerset Solution

Greedy Algorithm Solution

Code Example

Understanding the Knapsack

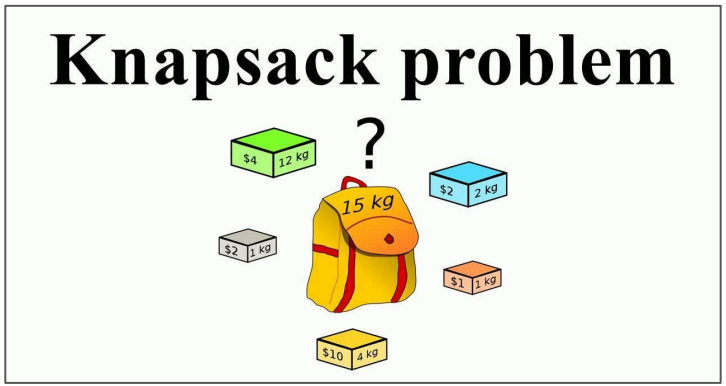
Knapsack Problem



Knapsack Problem

Definition:

- a knapsack has limited room for items inside.
- The packed knapsack can only contain a subset of items
- The problem is finding the **optimal** subset

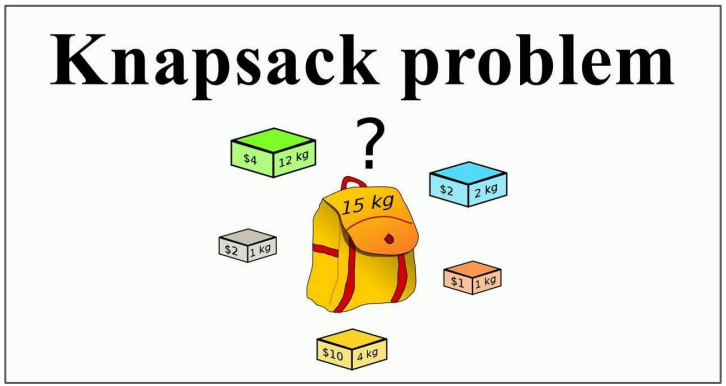


Knapsack Problem

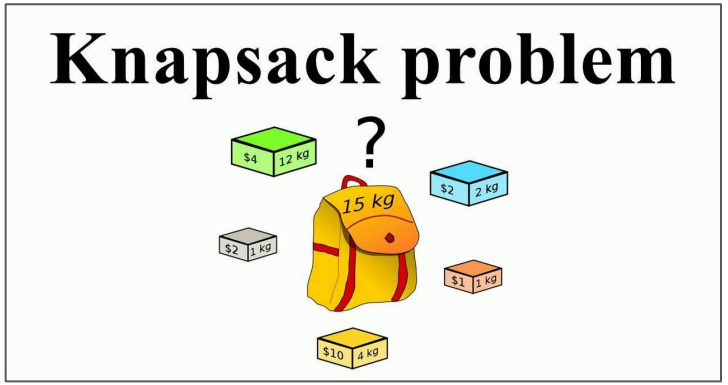
Example:

- the knapsack can only contain 15 kg
- which subset of items should fill the sack?

Lime: $4, 12 kg
Gray: $2, 1 kg
Blue: $2, 2 kg
Orange: $1, 1 kg
Yellow: $10, 4 kg



Knapsack Problem



Example:

- the knapsack can only contain 15 kg
- which subset of items should fill the sack?

Lime: $4, 12 kg
Gray: $2, 1 kg
Blue: $2, 2 kg
Orange: $1, 1 kg
Yellow: $10, 4 kg

Each item has a **value** ($)

Each item has a **weight** (kg)

The knapsack is **constrained** to have a max weight

Optimization **Objective**

Definition:

- Choosing the ideal items
- Follow a goal

Example

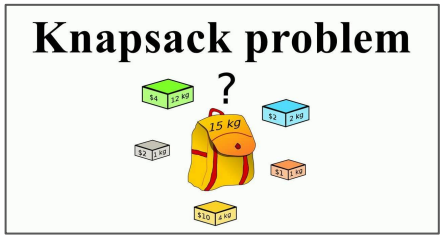
- choose items to maximize value
- choose items to minimize value (e.g. let's say calories)
- choose items to maximize weight
- choose items to maximize density (e.g. value/weight)

Powerset Solution

Powerset Solution

The objective is to maximize the value in a knapsack that can only hold 15 kg max

Lime: $4, 12 kg
Gray: $2, 1 kg
Blue: $2, 2 kg
Orange: $1, 1 kg
Yellow: $10, 4 kg



1. Generate the Powerset:

{}, {L}, {G}, {GL}, {B}, {BL}, {BG}, {BGL}, {O}, {OL}, {OG}, {OGL}, {OB}, {OBL}, {OBG}, {OBGL}, {Y}, {YL}, {YG}, {YGL}, {YB}, {YBL}, {YBG}, {YBGL}, {YO}, {YOL}, {YOG}, {YOGL}, {YOB}, {YOBL}, {YOBG}, {YOBGL}

2. Search through the powerset to find the subset that has largest $ value under 15 kg !

Critical Thinking

1. Generate the Powerset:

{}, {L}, {G}, {GL}, {B}, {BL}, {BG}, {BGL}, {O}, {OL}, {OG}, {OGL}, {OB}, {OBL}, {OBG}, {OBGL}, {Y}, {YL}, {YG}, {YGL}, {YB}, {YBL}, {YBG}, {YBGL}, {YO}, {YOL}, {YOG}, {YOGL}, {YOB}, {YOBL}, {YOBG}, {YOBGL}

2. Search through the powerset to find the subset that has largest $ value under 15 kg

What does the search entail? Be ready to answer out loud

Pseudo Code

for each subset

set up a value tracker, and weight tracker variable

loop through everything in the subset one at a time

increment the value and weight as you go

if the max weight is exceeded, stop and move on to the Next SUBSET

if no weight issue, only save the subset if it has the best value you've seen so far

Greedy Approximation

Greedy

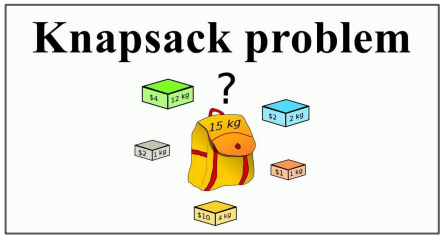
Definition

- take the best thing that fits within the **constraint**
- best is guided by the **objective**
- repeat the process until the **constraint** is hit

Greedy Solution

The objective is to maximize the value in a knapsack that can only hold 15 kg max

Lime: $4, 12 kg
Gray: $2, 1 kg
Blue: $2, 2 kg
Orange: $1, 1 kg
Yellow: $10, 4 kg



1. Take the best $ value first → Yellow
2. Check that the constraint is not violated (11 kg remain)
3. Take the best $ value again → Lime
4. Check that the constraint is not violated (-1 kg remain XXXX)
5. Discard Lime and continue greedy algorithm

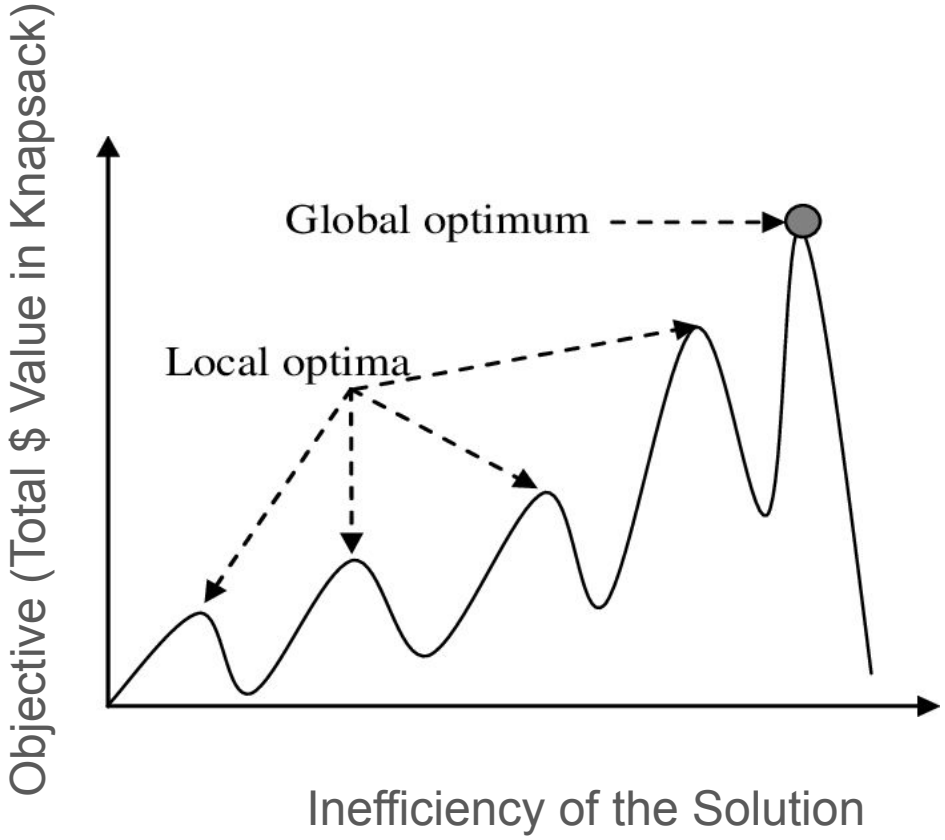
Greedy vs Powerset

The **Global Optimum** will be found with the **powerset**.

- But the search is extremely inefficient! $O(2^n)$

A **Local Optimum** will be found with the **Greedy Algorithm**

- Greedy is efficient O(n)



Code Example

For the Code Example

Think of 8 foods that you find delicious

- **Value** will be deliciousness value for you
- **Weight** will be the $ cost of the food
- **Objective** will be getting the most value while not exceeding some calorie **constraint**

Burger		Pizza
1 burger	SERVING SIZE	1 slice (1/8 of 12" pizza)
354	CALORIES	285
22g	PROTEIN	12g
27g	CARBOHYDRATES	36g
18g	FAT	10g
7g	SATURATED FAT	4g
80mg	CHOLESTEROL	22mg
2.5mg	IRON	2mg
4.2mg	ZINC	n/a
2.4mcg	VITAMIN B12	n/a