Multiple Knapsacks and Logic

Junaid Hasan

Lecture 6 Discrete Math Modelling

Multiple Knapsacks

- Last Friday we talked about the Knapsack problem.
- Let us consider an extension to the Knapsack Problem:
- We start with a collection of items with varying weights and values.
- However, now we can pack them in N equal knapsacks (say 5 knapsacks).
- Again, we want to pack so that the total volume is maximum.

Contd..

- Suppose we have 5 knapsacks(bins) each of maximum allowed weight 100.
- The weights are `weights = {48, 30, 42, 36, 36, 48, 42, 42, 36, 24, 30, 30, 42, 36, 36}`
- The values are `values = {10, 30, 25, 50, 35, 30, 15, 40, 30, 35, 45, 10, 20, 30, 25}`
- How to pack?

Model: Multiple Knapsack

- Let us consider `data[i,j]` as a binary variable which is 1 when item i is packed in bin j.
- Constraints:
- Each item goes in exactly one bin for each i

$$\sum_{j} data[i,j] \leq 1$$

- Bin capacity for each bin j

$$\sum_{i} data[i,j] \cdot weights[i] \leq 100$$

- Maximize value

$$\sum_{i} data[i,j] \cdot values[i]$$

Bin Packing

- Let us ask a new question.
- Suppose we have an infinite number of knapsacks(bins) of a common size say 100.
- However we want to pack all items with the fewest number of bins.
- An application of this problem is when a logistics agent wants to pack items in boxes of a fixed size and wants to use the fewest number of boxes.
- Note that the value of the items is irrelevant now.

Model

- `x[i,j]` is a binary variable which records if object i is placed in bin j.
- `y[j]` is a binary variable which records if bin j is used.
- Contraints:
- Each item is packed in exactly one bin, for all i

$$\sum_j x[i,j] = 1$$

- Bin capacity for each j

$$\sum_i x[i,j] \cdot weights[i] \leq 100 \cdot y[j]$$

- The multiplication by y[j] makes sure that the capcity is 0 if not used.
- Minimize

$$\sum_j y[j]$$

.

LPs and Logic

- Let us change course and discuss how to put logical constraints in LPs.
- Suppose we have two binary variables and we require that x_1 or x_2 is 1. How?
- We can do this by
$$x_1 + x_2 \geq 1.$$
- If we want exclusive or, then?
-
$$x_1 + x_2 = 1$$

More logical operations

- If we want x_1 to imply x_2
- If $x_1 = 1$, then $x_2 = 1$, how to enforce this with linear constraints?
- Answer:

$$x_2 \geq x_1.$$

AND operator

- If we want $b = 1$ if $x_1 = 1$ **and** $x_2 = 1$, and $b = 0$ otherwise. Then?
- $$\begin{aligned} b &\geq x_1 + x_2 - 1 \\ b &\leq x_1 \\ b &\leq x_2 \\ b &\in \{0, 1\} \end{aligned}$$
- When both are 1, the first condition forces b to be 1 and
- When either is 0, it forces b to be less than them making b equal to 0.

OR Operator

- If we want $b = 1$ if $x_1 = 1$ **or** $x_2 = 1$, then
- $$b \leq x_1 + x_2$$
$$b \geq x_1$$
$$b \geq x_2$$
$$b \in \{0, 1\}$$
- When both are 0, the first statement makes b 0 as well and
- When either is 1, then the greater than equal makes b equal to 1 as well.

XOR Operator

- If we want $b = 1$ if $x_1 = 1$ **xor** $x_2 = 1$. This means that b must be 0 when both are 1, and b is 1 only when exactly on of x_1, x_2 is 1.
- $$b \leq x_1 + x_2$$
  $$b \geq x_1 - x_2$$
  $$b \geq x_2 - x_1$$
  $$b \leq 2 - x_1 - x_2$$
- The first condition ensures b is 0, when both are 0.
- The last condition ensures b is 0 when both are 1.
- The remaining two ensure b is 1, when exactly one of them is 1.

More complex contructions

- If we want either $2x_1 + x_2 \geq 5$ or $2x_3 - x_4 \leq 2$ or both.
- Then how to do?
- We add a binary variable y and
- a large value M. How large M is depends on the largest possible value taken by $2x_3 - x_4$ and $2x_1 + x_2$.
- We make M larger than the largest these can take so that they are always less than M.
- $$2x_1 + x_2 \geq 5 - M \cdot y$$
  $$2x_3 - x_4 \leq 2 + M \cdot (1 - y)$$

contd..

- $$2x_1 + x_2 \geq 5 - M \cdot y$$
  $$2x_3 - x_4 \leq 2 + M \cdot (1 - y)$$
- If $y = 0$ then this means we only check if $2x_1 + x_2 \geq 5$ because the second is always true.
- If $y = 1$, then we only check the second condition as the first is always true.

Indicator variable

- Suppose x is an integer variable (say $x \geq 0$).
- We want a binary variable y such that if \$ x >0\$ then $y = 1$, else $y = 0$.
- Again we choose a postive M such that $x < M$ always holds then
- $$x \leq M \cdot y$$
  $$y \leq M \cdot x$$
- Why?