CSE 102 2-20-20

Greedy Algorithms

Ex. Knapsack (continuous)

maximize: $\sum_{i=1}^{n} x_i \cdot v_i$

subject to: $\sum_{i=1}^{n} x_i \cdot w_i \leq W$ *

where $W > 0$, $v_i > 0$, $w_i > 0$ $(1 \leq i \leq n)$

and $x = (x_1, x_2, \ldots, x_n)$ satisfies

$0 \leq x_i \leq 1$.

Recall x is feasible iff $*$ is satisfied.

Greedy strategy:

make a locally optimal choice, then solve resulting sub-problem.

must decide which object is "best".

selection function: $f(i)$

at each step, maximize $f(i)$

```
Knapsack (v, w, W)
1.) n ← length[v]
2.) for i ← 1 to n
3.)       x[i] ← 0
4.) weight ← 0
5.) while weight < W
*6.)      i ← THE "BEST" REMAINING OBJECT
7.)      if weight + w[i] ≤ W
8.)            x[i] ← 1
9.)            weight ← weight + w[i]
10.)           mark i AS INCLUDED.
11.)     else
12.)           x[i] ← (W - weight) / w[i]
13.)           weight ← W
14.) return x
```

GREEDY LOOP

THE "BEST" OBJECT IN (6) CAN BE INTERPRETED IN SEVERAL WAYS.

IN GENERAL WE DEFINE A <u>SELECTION FUNCTION</u> $f(i)$ WHICH ENCODES THE DESIRABILITY OF OBJECT i. LINE (6) THEN MAXIMIZES f OVER ALL REMAINING (UNMARKED) OBJECTS.

Possible selection fcns.

- $f(i) = v_i$
- $f(i) = \frac{1}{w_i}$
- $f(i) = \frac{v_i}{w_i}$

Ex $n=5$, $W=10$

i	1	2	3	4	5	order
v_i	2	3	6.6	4	6	3,5,4,2,1
w_i	1	2	3	4	5	1,2,3,4,5
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2	3,1,2,5,4

f(i)	x					Value
v_i	0	0	1	.5	1	14.6
$1/w_i$	1	1	1	1	0	15.6
v_i/w_i	1	1	1	0	.8	16.4

Theorem

If we maximize $\frac{v_i}{w_i}$ on line 6, then Knapsack returns an optimal solution.

Proof

Assume w.l.o.g that

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \frac{v_3}{w_3} \geq \cdots \geq \frac{v_n}{w_n}$$

let $x = (x_1, x_2, \ldots, x_n)$ be the solution returned by Knapsack.

If all $x_i = 1$, then x is optimal. otherwise let j be the j^{st} index s.t. $x_j < 1$. so we have

$$x_i = 1 \qquad (1 \le i < j)$$

$$x_j < 1$$

$$x_i = 0 \qquad (j < i \le n)$$

and

$$\sum_{i=1}^{n} x_i w_i = W.$$

let $V(x) = \sum_{i=1}^{n} x_i v_i$, the value of x.

let $y = (y_1, \ldots, y_n)$ be any other feasible solution, and let

$$V(y) = \sum_{i=1}^{n} y_i v_i$$

must show: $V(x) \geq V(y)$.

since y is feasible

$$\sum_{i=1}^{n} y_i w_i \leq W$$

hence

$$\sum_{i=1}^{n} (x_i - y_i) w_i = \sum_{i=1}^{n} x_i w_i - \sum_{i=1}^{n} y_i w_i$$

$$= W - \sum_{i=1}^{n} y_i w_i \geq 0$$

Now

$$V(x) - V(y) = \sum_{i=1}^{n} x_i v_i - \sum_{i=1}^{n} y_i v_i$$

$$= \sum_{i=1}^{n} (x_i - y_i) v_i$$

$$= \sum_{i=1}^{n} (x_i - y_i) w_i \cdot \left(\frac{v_i}{w_i}\right)$$

observe

$$i < j \Rightarrow x_i = 1 \Rightarrow x_i - y_i \geq 0 \overset{!}{\geq} \frac{v_i}{w_i} \geq \frac{v_j}{w_j}$$

$$\Rightarrow (x_i - y_i)\left(\frac{v_i}{w_i}\right) \geq (x_i - y_i)\left(\frac{v_j}{w_j}\right)$$

$$i > j \Rightarrow x_i = 0 \Rightarrow x_i - y_i \leq 0 \overset{!}{\leq} \frac{v_i}{w_i} \leq \frac{v_j}{w_j}$$

$$\Rightarrow (x_i - y_i)\left(\frac{v_i}{w_i}\right) \geq (x_i - y_i)\left(\frac{v_j}{w_j}\right)$$

$$i=j \Rightarrow (x_i-y_i)\left(\frac{v_i}{w_i}\right)=(x_i-y_i)\left(\frac{v_j}{w_j}\right)$$

In all cases $(1 \le i \le n)$:

$$(x_i-y_i)\left(\frac{v_i}{w_i}\right) \ge (x_i-y_i)\left(\frac{v_j}{w_j}\right)$$

Thus

$$V(x)-V(y) = \sum_{i=1}^{n} (x_i-y_i)\left(\frac{v_i}{w_i}\right)\cdot w_i$$

$$\ge \sum_{i=1}^{n} (x_i-y_i)\left(\frac{v_j}{w_j}\right) w_i$$

$$= \left(\frac{v_j}{w_j}\right) \cdot \sum_{i=1}^{n} (x_i-y_i)\, w_i$$

$$\ge 0$$

$\therefore V(x) \geq V(y)$

since y was an arbitrary feasible solution, x is optimal. ∎

Runtime:

- either sort on $\frac{v_i}{w_i}$ (cost $\Theta(n \log n)$) then run algorithm: $\Theta(n)$, so total cost $\Theta(n \log n)$
- or don't sort and search for "best" on each iteration:

$$\text{cost} = \Theta(n \log n)$$

What about Discrete Knapsack

Dyn. Prog had cost $\Theta(n \cdot (W+1))$

Exercise

modify greedy algorithm to break when can't steal all of obj i. Same selection fcn $f(i) = \frac{v_i}{w_i}$.

- find an Example in which greedy returns optimal.
- find an Example in which greedy is not optimal.

Coin changing again

denominations $d = (d_1, \ldots, d_n)$

amount to be paid: N

Greedy strategy:

- from among all denom. whose addition would not cause sum to exceed N, choose largest
- stop when sum is N.

We assume

$$1 = d_1 < d_2 < d_3 < \dots < d_n$$

Exercise (hard, but doable).

show that for

$$d = (1, 5, 10, 25, 100)$$

the greedy strategy yields an optimal coin disbursement. (for all N.)

Exercise (easy)

show if $d = (1, 10, 25, 100)$, greedy strategy does not produce opt. solution for some N.

Exercise (very hard)

Characterize all $d = (d_1, \ldots, d_n)$ s.t. greedy strategy yields opt. soln for all $N \geq 0$.

What features tell us a greedy solution may work?

- optimal substructure.
- Greedy choice property: some greedy strategy works.

Minimum Weight Spanning Trees (MWST)

Given a weighted connected Graph G, find a MWST in G.