COL 702 Lecture 11 Sept 4

Optimization Problems.

- Knapsack problem.

A Knapsack with capacity B and n elements $e_1, e_2 \ldots e_n$ with volumes $v_1\ v_2\ \ldots\ v_n$

profits $p_1\ p_2\ \cdots\ p_n$

$$\max \sum x_i p_i \qquad x_i = \begin{cases} 1 & \text{if } e_i \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$$

s.t. $$\sum x_i v_i \le B$$

$$x_i \in \{0,1\}$$

SH case of a more general class of probs called Linear prog

$$0 \le x_i \le 1$$

Constraints in the form of linear inequalities

$$\left[\begin{array}{lll} x_1 + x_3 + x_5 & \le & 5 \\ 2x_2 + 4x_3 & \le & 8 \\ & & \end{array}\right.$$

$x_i \geq 0$

Maximize some linear function in x

Max $c_1 x_1 + c_2 x_2 + \cdots c_n x_n$

Matrix form Max $\bar{c} \cdot \bar{x}$

s.t. $A\bar{x} \leq \bar{b}$

A : $m \times n$ matrix

maximal

~~Minimal~~ Spanning Tree (MST)

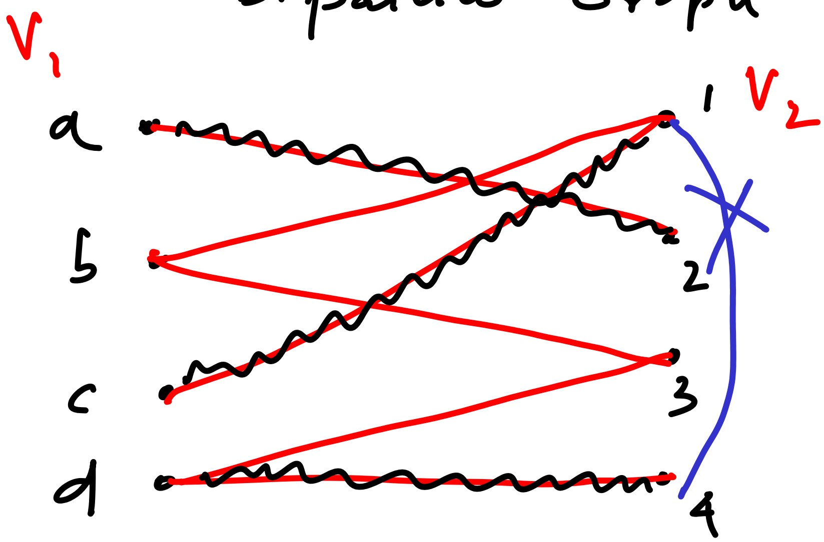
Graph $G = (V, E)$ with weights on edges $\omega : E \to \mathbb{R}^+$

Max $\sum x_i \, \omega(e_i)$ $\quad x_i = \begin{cases} 1 & \text{if } e_i \text{ is chosen} \\ 0 & \end{cases}$

s.t. No cycles are induced by the edges chosen

Maximum Matching (Assignment Problem)

Bipartite Graph



$(b,3)$ can be added to preserve 1–1

Choosing a subset of edges such that no vertex is incident to more than 1 edges is a feasible matching

Max matching : Maximum Cardinality

Maximum Weight (weights on every edge)

Write the appropriate math formulation

Common Framework for Optimization

We have a ground set

$$S : \{e_1, e_2\ e_3 \ldots e_n\}$$

weights $w : S \to \mathbb{R}^+$

The set of subsets 2^S consists of all possible subsets of S

However some of them are 'feasible' and others are 'infeasible'

The set of feasible subsets is often called "Independent"

<u>Goal</u>: Choose a subset $T \in 2^S$ s.t. the weight $w(T)$ is maximum

$$w(T) = \sum_{x \in T} w(x)$$

<u>Some common properties</u>. ϕ is independent

If $S_1 \in \mathcal{I}$ and $S_2 \subset S_1$ then $S_2 \in \mathcal{I}$

<u>subset-property</u>

Greedy : never backtracks

Backtracking may try all possible subsets in $\mathcal{I}$

Generic Greedy

Initialize $T = \phi$

Consider the set of elements in reverse sorted order of their weights

highest wt $\rightarrow$ $\tilde{e}_1, \tilde{e}_2, \tilde{e}_3 \ldots \tilde{e}_n$ $\leftarrow$ least wt

Repeat

Add the next object e_i to T if $T \cup \{e_i\}$ is feasible

until no more edges remain

(Kruskals algorithm follows this)

Running time: ① Presorting of elements

② For the iterative part, we repeatedly "test" if by adding the next object it remains feasible

Suppose the time is T_i for the i^{th} obj

$$\sum_{i=1}^{n} T_i$$

How about "correctness"

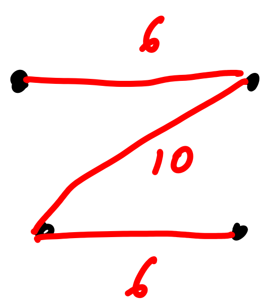
Is the subset T the best soln for all possible instances of the problem?

<u>Knapsack</u> $B = 10$

	w_i	p_i
e_1	10	10
e_2	5	6
e_3	5	6

<u>MST</u> : It works!

<u>Matching</u>



The problems for which Greedy succeeds, the underlying set system is called a "Matroid"
(weights have no role in this framework)

Knapsack :

Consider the profit per volume as the metric for being greedy. Show that the final soln is at least 50% of the optimal