$\boxed{\text{LECTURE \#6}}$ Oct 16 2011

1) Class: Discussions on OptMap and SAT.
2) B&B & IP
3) Architecture Team for B&B.

DIVIDE & CONQUER:

Consider the following Optimization Problem:

$$x^* = \underset{x}{\operatorname{argmin}} \{ f(x) : x \in S \}$$

$\boxed{\text{Proposition 1:}}$ (Branching: b = Branching Factor)

Let $S = S_1 \cup S_2 \cup \cdots \cup S_b$ be a decomposition of S into smaller sets $(\forall_k\ |S_k| < |S|) \wedge \underset{S_k \subsetneq S}{S_k \neq \phi}$

Let $x_k^* = \underset{x}{\operatorname{argmin}} \{ f(x) : x \in S_k \}$, $\forall k \in \{1, .., b\}$

Then

$$f(x^*) = \min \{ f(x_1^*), \ldots, f(x_b^*) \}.$$

<u>Example</u>:

$$S = \{0, 1\}^n$$

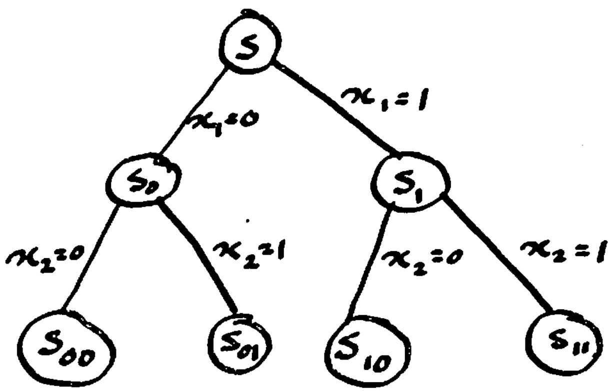
$\Rightarrow S_0 = \{ x \in S \mid x_1 = 0 \}$ and $S_1 = \{ x \in S \mid x_1 = 1 \}$

$\Rightarrow S_{00} = \{ x \in S \mid x_1 = 0, x_2 = 0 \} = \{ x \in S_0 \mid x_2 = 0 \}$

$S_{01} = \{ x \in S \mid x_1 = 0, x_2 = 1 \} = \{ x \in S_0 \mid x_2 = 1 \}$

$S_{10} = \{ x \in S \mid x_1 = 1, x_2 = 0 \} = \{ x \in S_1 \mid x_2 = 0 \}$

$S_{11} = \{ x \in S \mid x_1 = 1, x_2 = 1 \} = \{ x \in S_1 \mid x_2 = 1 \}$



Binary Enumeration Tree.

$\boxed{\text{Proposition 2:}}$ (Bounding & Pruning [Fathoming]: Upper and Lower Bounds.)

Let $S = S_1 \cup S_2 \cup \dots \cup S_b$ be a decomposition of S into smaller sets:

$$\forall_k \quad S_k \neq \phi \quad \text{and} \quad S_k \subsetneq S$$

Let $x_k^* = \underset{x}{\operatorname{argmin}} \{ f(x) \mid x \in S_k \} \quad \forall_{k \in \{1,..,b\}}$

Let $f_k^{low} \leq f(x_k^*) \leq f_k^{up}$ be lower and upper bounds on x_k^*.

Let

$$f^{low} = \min_k f_k^{low} \quad \text{and} \quad f^{up} = \min_k f_k^{up}$$

Then.

1a) $$f^{low} \leq f(x^*) \leq f^{up}$$

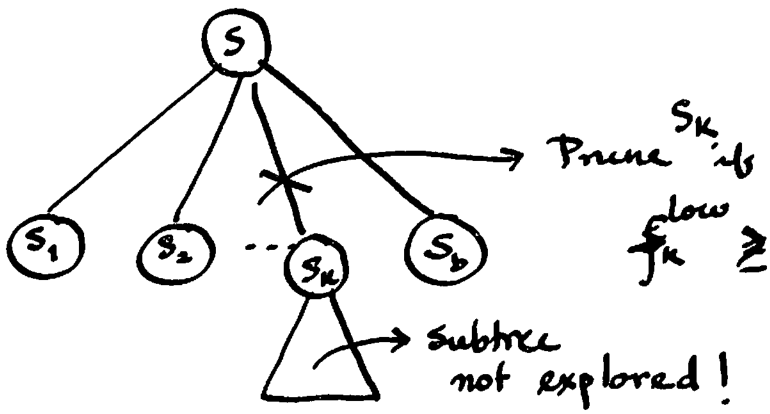
1b) $$f^{low} = f^{up} \Rightarrow f^{low} = f^{up} = f(x^*)$$

2) If $S_t = \{x_t\}$ be a singleton set, then

$$f(x_t) = f(x_t^*) = f_t^{low} = f_t^{up}$$

3) PRUNE/FATHOM

$$f(x^*) = \min \{ f(x_t^*) \mid x_t^* \in S_t,\ f_t^{low} \leq f^{up} \}$$



$$f_k^{low} \geq f^{up} = \min_k f_k^{up}.$$

Divide & Conquer:

OptDC (f, S):

If $S = \{x\}$ = singleton **then** return $\langle x, f(x) \rangle$;

else

```
    Let  S = S1 ∪ S2 ∪ ... ∪ Sb;      [ Si ⊊ S ∧ Si ≠ φ ]
    Select  x ∈ S, arbitrarily;
    temp := <x, f(x)>;

    for  i = 1, .. b  loop
       if  Si ≠ φ  then
          temp := MIN (temp, OptDC (f, Si));

    return temp;
```

Branch & Bound:

OptBB (f, S);

if $S = \{x\}$ = singleton then return $\langle x, f(x)\rangle$

else

$S = S_1 \cup S_2 \cup \cdots \cup S_b$; $\quad [S_i \subsetneq S \wedge S_i \neq \phi]$

$f^{up} = \min(f_1^{up}, f_2^{up}, \cdots, f_b^{up})$;

$f^{low} = \min(f_1^{low}, f_2^{low}, \cdots, f_b^{low})$;

Select $x \in S$, arbitrarily;

temp := $\langle x, f(x)\rangle$

for $i := 1, .. b$ loop

if $S_i \neq \phi$ and $f_i^{low} \leq f^{up}$ then

temp := MIN (temp, OptBB(f, S_i));

return temp.

How to compute Upper and Lower Bounds?

a) If $S = \{x\}$ = singleton $f^{up} = f(x) = f^{low}$ or $|S| = k$, small $= \{x_1, \ldots, x_k\}$ $f^{up} = \min(f(x_1), \ldots, f(x_k)) = f^{low}$.

b) If $|S| > k$ then

$$f^{up} \leq f(x), \forall x \in S.$$

Randomly sample S, k times; $\{x_1', \ldots, x_k'$

Set $f^{up} = \min(f(x_1'), \ldots, f(x_k'))$

c) How to compute lower Bounds.

TSP (Travelling Salesman Problem)

$$f^{low}(G, \omega) = \frac{1}{2} \sum_{\substack{i \in V \\ j,k \neq i}} \underbrace{\omega_{ij} + \omega_{ik}}$$

Graph $G = (V, E)$

$\omega : E \to \mathbb{R}$

$: \langle i, j \rangle \mapsto \omega_{ij}$

$\omega_{ij} = \min(W_{i,\cdot})$
$\omega_{ik} = \min(W_{i,\cdot} \setminus \omega_{ij})$

Sum of the costs of the two least cost edges adjacent to i.

Note: Cost of any tour

$$= \frac{1}{2} \sum_{i \in V} \underbrace{\omega_{i,i-1} + \omega_{i,i+1}}$$

Sum of the costs of the two tour edges adjacent to i.

0-1 Integer Programming:

$$\max \sum_{j=1}^{n} v_j x_j$$

$$\sum_{j=1}^{n} \omega_j x_j \leq b$$

$$x_j \in \{0, 1\} \quad j = 1, \ldots, n.$$

Relaxing x_j's will result in

$$0 \leq x_j \leq 1 \quad j = 1, \ldots, n.$$

A polytime linear programming problem.

The Knapsack Problem: (Binary KP).

1) A set of objects to select from.

   n: Number of objects, indexed by
   $1, 2, \dots, j, \dots, n$

2) Each object has a weight w_j, and a value v_j. $\Big\}$ $j \in \{1, \dots, n\}$

3) You have a knapsack to carry the objects, but you are not able to carry more than a weight of b.

   $$x_j = \text{Indicator variable} = \begin{cases} 1 & \text{if object } j \text{ is selected} \\ 0 & \text{otherwise.} \end{cases}$$

4) Your goal is to maximize the total value you can carry.

$$\Rightarrow \quad \max \sum_{j=1}^{n} v_j x_j \quad \text{subject to} \quad \sum_{j=1}^{n} w_j x_j \le b$$

$$x_j \in \{0, 1\} \quad j = 1, \dots n.$$

~

Integer Linear Programming:

$$\max \; c^T x \qquad\qquad c \in \mathbb{Z}^n$$

$$\text{subject to } Ax \le b \qquad\qquad A \in \mathbb{Z}^{m \times n}$$

$$x \ge 0 \qquad\qquad b \in \mathbb{Z}^m$$

$$\text{and } x \in \mathbb{Z} \qquad\qquad \vdots$$

0-1 Case. LP relaxation

Solve the LP version of the problem:

$x \in \{0,1\}$ is replaced by $0 \le x \le 1$.

Let x^* be a solution to the relaxed problem.

$x^* \in \mathbb{R}^n \equiv [0,1]^n$, though the soln desired must be in $\{0,1\}^n$.

Upper bound.

a) Round each x_i^* to x_i in 0 or 1.
b) Generate random $x_i \in \{0,1\}$ with $\text{Prob}(x_i = 1) = x_i^*$

→ Check that the guess generated $\hat{x}$ is feasible.

Suppose you have $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_k$ as feasible guesses.

$\boxed{A\hat{x}_i \le b.}$ ← feasibility

Upper bound $= \min(+\infty, c^T \hat{x}_i)$

Lower bound

$c^T x^*$ is a lower bound on the problem

since every solution to ILP is also a solution to the relaxed LP.