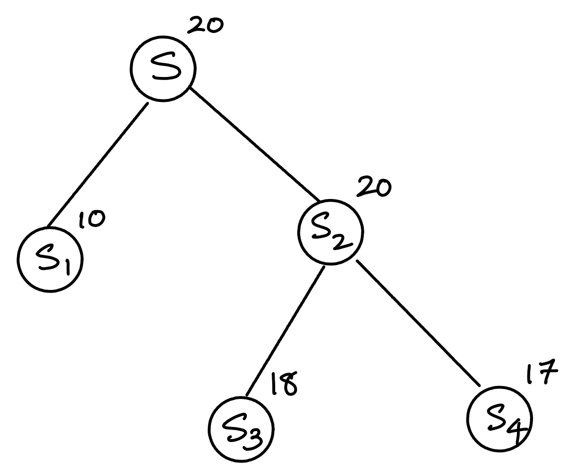
MATH 567: Lecture 15 (02/27/2025)

Today:
* B&B strategies
* reduced cost fixing in B&B
* types of branching

Node selection Strategies

How do we select a subproblem from $\mathcal{L}$?

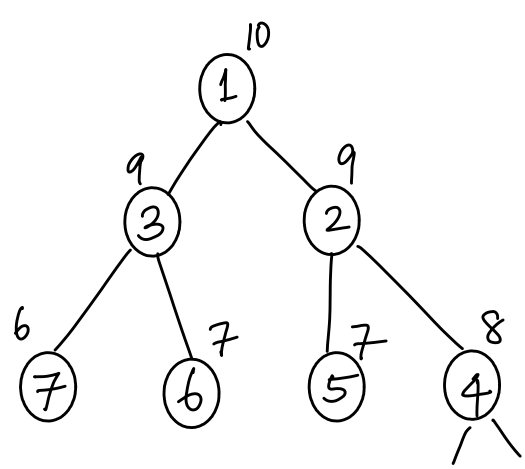
Consider this example:



If we subdivide S_1, z_u will remain at 20. If $z^*=15$ (optimal objective function value) and we knew it, we would not subdivide S_1. On the other hand, subdividing S_2 decreases z_u to 18.

This example seems to suggest that choosing a node with a high z_u may be a good idea. This strategy is called **best node first** (BNF) strategy. (pick subproblem from $\mathcal{L}$ with largest z_u).

Another typical BNF B&B tree



nodes are numbered in the order they are examined here

Advantages of BNF Strategy

global upper bound

* Rapidly decreases z_u.

optimal z-value

* Never subdivides a node T_k with $z_u(T_k) < z^*$, can prune many nodes, and hence the # nodes to prove optimality is relatively small.
  ↳ assuming you already identified the optimal solution — you still have to prove it is indeed optimal.

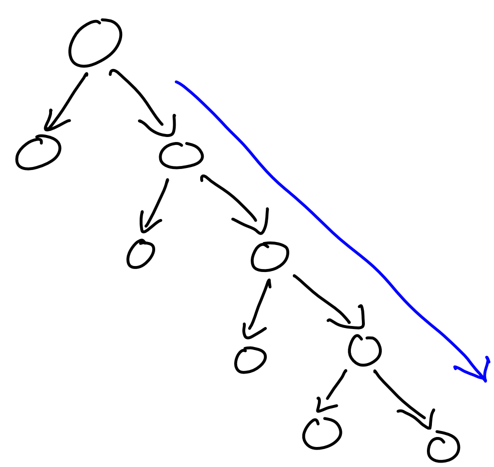
Disadvantages of BNF strategy

* The B&B tree is widespread, and memory needed to store the list of subproblems $\mathcal{L}$ may be huge.
* May take a long time to find an integer feasible solution (i.e., a node T_k with $z_u(T_k) = z_\ell(T_k)$).

Depth-First Search (DFS) B&B Strategy

Exact opposite to BNF $\longrightarrow$ always select the problem that was added to $\mathcal{L}$ the last (LIFO order).

A typical DFS B&B tree:



Advantages of DFS B&B Strategy

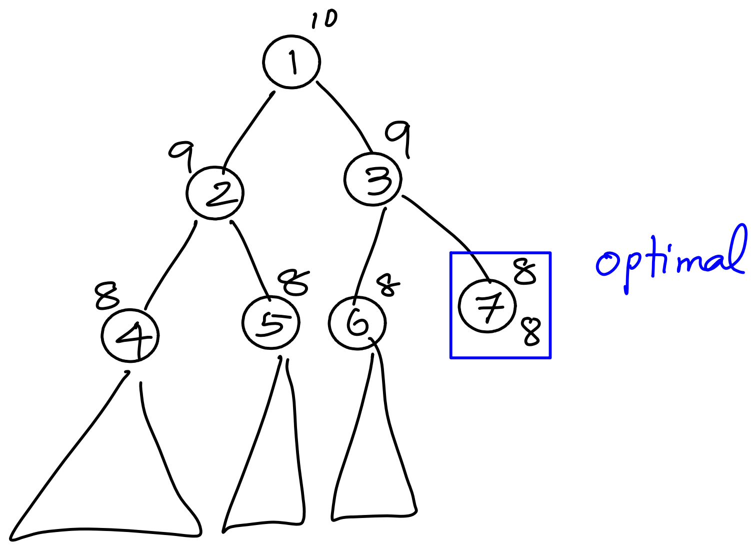
* Maximum depth of B&B tree is n; at any point, DFS stores at most $2n$ subproblems in $\mathcal{L}$.
* Since it gets down deep quickly in the B&B tree, DFS find integer feasible solutions relatively quickly.

Disadvantages of DFS

* If it hits a "wrong" subtree, it may not find a feasible solution, or even change the bounds for a long time.
* It may take a long time to prove optimality.

Let's consider a sample B&B tree, and how ~~both~~ strategies (BNF and DFS) perform ~~on~~ the same. We will consider both "good" and "bad" extremes for their performances — "lucky" or "unlucky"!

An Example



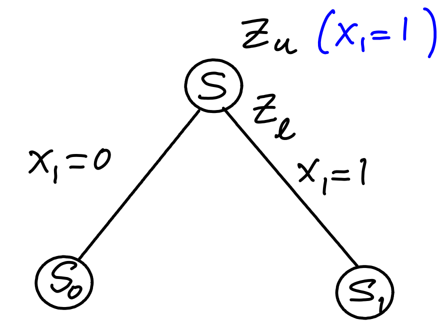
Strategy	# nodes until finding optimal solution	# nodes until finishing (proving optimality)
BNF lucky	4 (1-2-3-7)	7 (1-2-3-7-4-5-6)
BNF unlucky	7 (1-2-3-4-5-6-7)	7 (1-2-3-4-5-6-7)
DFS lucky	3 (1-3-7)	7 (1-3-7-6-2-5-4)
DFS unlucky	M	M

So, DFS is a gambler, while BNF is conservative.

In practice, we combine the two strategies, along with other "intelligent" strategies specific to the problem in hand.

Reduced Cost fixing in B&B

Consider a 0-1 IP.



Say we solve the LP relaxation at S (to get $z_u(S)$), and in the optimal solution $\bar{x}$, we have $x_1=1$. Can we conclude that

$z_u(S_0) =$ LP optimum at $S_0 \le z_\ell$?

If yes, we can fix $x_1=1$ (in the optimal solution of IP).

LP relaxation at S — Dual

$$z_u(S) = \left\{ \begin{array}{ll} \max & \bar{c}^T\bar{x} \\ \text{s.t.} & A\bar{x} \le \bar{b} \rightarrow \bar{y} \ge \bar{0} \\ & \bar{x} \le \bar{1} \rightarrow \bar{u} \ge \bar{0} \\ & -\bar{x} \le \bar{0} \rightarrow \bar{v} \ge \bar{0} \end{array} \right. \quad (P) \uparrow \text{primal LP}$$

$$=$$

$$\left. \begin{array}{ll} \min & \bar{b}^T\bar{y} + \bar{1}^T\bar{u} \\ \text{s.t.} & A^T\bar{y} + \bar{u} - \bar{v} = \bar{c} \\ & \bar{y}, \bar{u}, \bar{v} \ge \bar{0} \end{array} \right\} (D) \uparrow \text{dual LP}$$

Theorem 13 Suppose the optimal solution to (P) and (D) be $(\bar{x}, \bar{y}, \bar{u}, \bar{v})$ with

1. $x_1 = 1$, and
2. $u_1 \ge z_u(S) - z_\ell$

$z_u(S)$ also $=$ opt(D), the optimal obj. fn of dual (D)

Then $z_u(S_0) \le z_\ell$. So we can fix $x_1 = 1$.

Recall complementary slackness conditions — if a constraint in (P) is satisfied as a strict inequality, i.e., it's nonbinding, the corresponding dual variable will be zero in the optimal solution. So, $v_1 = 0$ here (as $-x_1 \le 0$ is not binding).

Proof

Consider the dual LP at S_0. $(D) \wedge (x_1 = 0)$ is the same as (D), but with v_1 free (urs). v_1 appears in (D) only in $(A^T\bar{y})_1 + u_1 - v_1 = c_1$. Hence a feasible solution to $(D) \wedge (x_1=0)$ (i.e., (D) at S_0) is given by $(\bar{y}', \bar{u}', \bar{v}')$, where

$$\bar{y}' = \bar{y}, \quad \bar{u}' = \bar{u}, \quad \bar{v}' = \bar{v} \quad \text{except for } u_1' = 0,\ v_1' = -u_1.$$

$\Rightarrow$ The optimal obj. fn. value at $(D) \wedge (x_1 = 0) \leq z_u(S) - u_1$.

$$\leq z_\ell \text{ by (2)}.$$

Hence we can prune S_0, i.e., fix $x_1 = 1$. $\square$

In practice, reduced cost fixing and other similar strategies are all implemented as part of B&B (for example, in CPLEX). In fact, packages such as CPLEX do much more than simple B&B. Still, there are some pathological instances of certain IPs, which are bad for CPLEX even at moderate dimensions (≤ 100!).

Example

$$(P)\ \begin{cases} \max & 2x_1 + x_2 \\ \text{s.t.} & x_1 + 2x_2 \leq 2 \quad y \\ & x_1 \leq 1 \quad u_1 \\ & x_2 \leq 1 \quad u_2 \\ & -x_1 \leq 0 \quad v_1 \\ & -x_2 \leq 0 \quad v_2 \end{cases}$$

$$\begin{array}{ll} \min & 2y + u_1 + u_2 \\ \text{s.t.} & y + u_1 - v_1 = 2 \\ & 2y + u_2 - v_2 = 1 \\ & y, u_1, u_2, v_1, v_2 \geq 0 \end{array} \quad (D)$$

$z_\ell = z^* = 2$ with $\bar{x}^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

$z_u = 5/2$ with $\bar{x} = \begin{bmatrix} 1 \\ 1/2 \end{bmatrix}$.

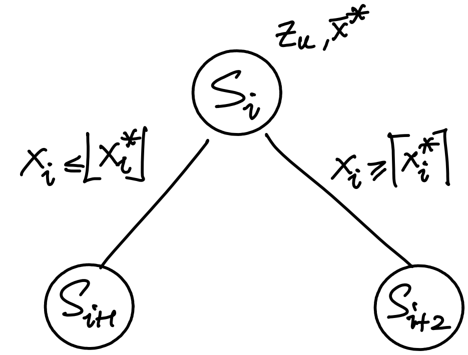
For (D), opt. solution is $y = \frac{1}{2}$, $u_1 = 3/2$.

$u_1 \geq z_u - z_\ell = \frac{5}{2} - 2 = \frac{1}{2}$.

So, fix $x_1 = 1$ by Theorem 13.

Types of Branching

① Binary branching

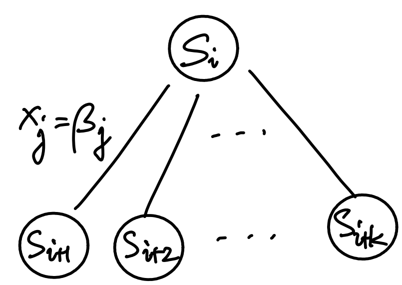


Solve LP relaxation at S_i.
Let the optimal solution to this LP relaxation be $\bar{x}^*$ with x_i^* non-integral, where $x_i \in \mathbb{Z}$ is required. Create two branches by adding $x_i \le \lfloor x_i^* \rfloor$ and $x_i \ge \lceil x_i^* \rceil$.

Example: $x_5 = 13.6$ (in $\bar{x}^*$). Create the branches $x_5 \le 13$ and $x_5 \ge 14$.

Binary variables are indeed covered in this case.

② Integer branching



$k = \lfloor \gamma_{ij} \rfloor - \lceil \delta_{ij} \rceil + 1$

Choose a variable x_j that needs to be integral, find

$$\delta_{ij} = \min\{x_j \mid \bar{x} \in \text{LP relaxation at } S_i\},$$
$$\gamma_{ij} = \max\{x_j \mid \bar{x} \in \text{LP relaxation at } S_i\}.$$

Create branches by adding constraints

$$x_j = \beta_j \quad \text{where} \quad \beta_j \in \{\lceil \delta_{ij} \rceil, \lceil \delta_{ij} \rceil + 1, \ldots, \lfloor \gamma_{ij} \rfloor\}.$$

So, create $\lfloor \gamma_{ij} \rfloor - \lceil \delta_{ij} \rceil + 1$ nodes.

e.g., $\delta_{ij} = 13.64$, $\gamma_{ij} = 16.39$ for $\delta_{ij} \le x_j \le \gamma_{ij}$; we create 3 branches with $x_j = 14, 15, 16$.