MAT3007 Optimization
Lecture 12 Integer Optimization
Branch-and-Bound Method

Yuang Chen

School of Data Scienc
The Chinese University of Hong Kong, Shenzhen

July 2, 2025

Outline

1. Min Cost Network Flow and Totally Unimodular

2. Branch-and-Bound Method

Outline

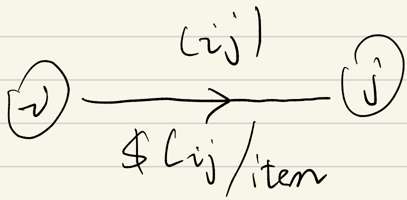
Minimum Cost Network Flow Model

- We are given a directed network $G = (N, A)$ with a set of nodes N and a set of arcs A.
- Each node $i \in N$ has an associated "supply" b_i. If $b_i > 0$ the node is a supply node, if $b_i < 0$ it is a demand node. We will assume that the network is balanced, i.e. $\sum_{i \in N} b_i = 0$.
- Each arc $(i, j) \in A$ has an associated cost c_{ij} and capacity u_{ij}.
- A flow on this network is a set of values on the arcs that obey capacities and satisfy flow conservation at the nodes.
- The goal is to find flows on each arc with minimum cost.

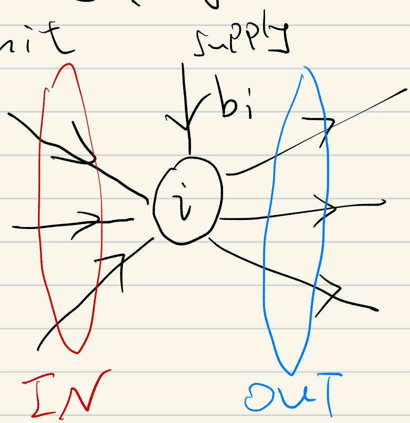
b_i : supply

$$b_i \begin{cases} > 0 & \text{supply} \\ < 0 & \text{demand} \end{cases}$$

balance : $\sum_{i \in N} b_i = 0$



Capacity : u_{ij} items
limit



IN + Supply = OUT

$$\sum_{ji \in A} x_{ji} + b_i = \sum_{ij \in A} x_{ij}, \quad \forall i \in N$$

Minimum Cost Network Flow Problem

\# items delivered from i to j

Let x_{ij} denote the flow from i to j

flow conservation constraint

$$
\begin{aligned}
\min \quad & \sum_{(ij)\in A} c_{ij}x_{ij} \\
\text{s.t.} \quad & \sum_{j:(ij)\in A} x_{ij} - \sum_{j:(ji)\in A} x_{ji} = b_i \quad \forall i \in N \\
& 0 \le x_{ij} \le u_{ij} \quad \forall (ij) \in A
\end{aligned}
$$

$Ax = b$

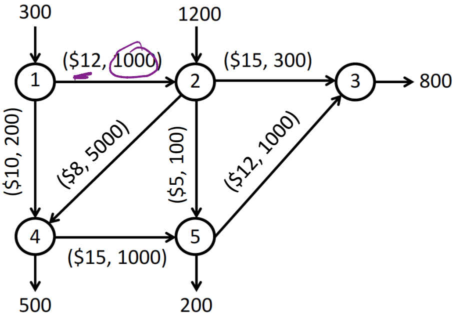
$x \geq 0$

~~$x_{ij} \in \mathbb{Z}$~~

No Need

Example

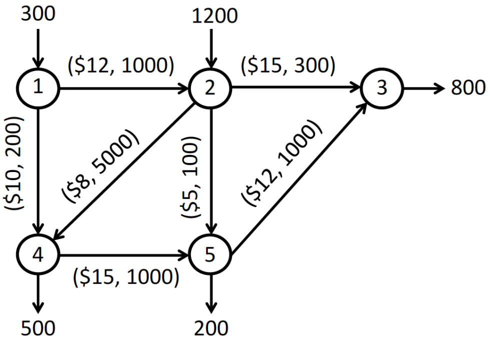
- A company has two manufacturing plants and 3 distribution centers (nodes)
- The plants and DCs are connected by transportation channels (arcs)
- The production amount and demand at each node is shown
- The capacity and cost per unit for each channel (arc) is shown
- Find the cheapest way to move units from plants to DCs



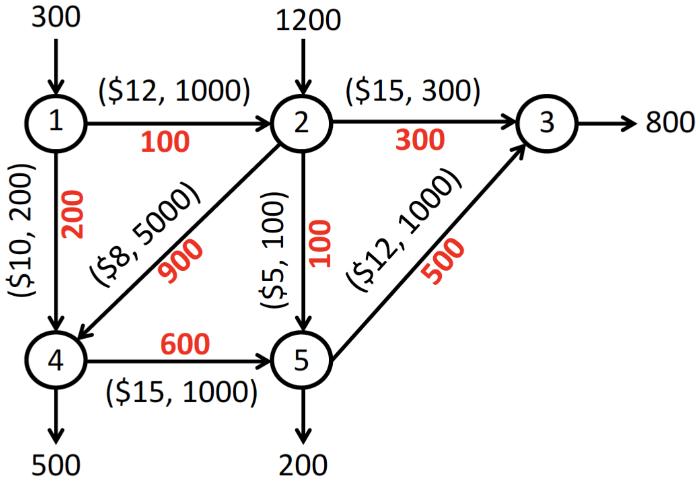
Example Formulation

LP Not IP

$$
\begin{aligned}
\min \quad & 12x_{12} + 10x_{14} + 15x_{23} + 8x_{24} \\
& +5x_{25} + 15x_{45} + 12x_{53} \\
\text{s.t.} \quad & x_{12} + x_{14} = 300 \\
& x_{23} + x_{24} + x_{25} - x_{12} = 1200 \\
& -x_{23} - x_{53} = -800 \\
& x_{45} - x_{14} - x_{24} = -500 \\
& x_{53} - x_{25} - x_{45} = -200 \\
& 0 \leq x_{12} \leq 1000,\ \ 0 \leq x_{14} \leq 200, \\
& 0 \leq x_{23} \leq 300, 0 \leq x_{24} \leq 5000 \\
& 0 \leq x_{25} \leq 100,\ \ 0 \leq x_{45} \leq 1000 \\
& 0 \leq x_{53} \leq 1000
\end{aligned}
$$



Example Solution



Integrality of Min Cost Network Flow

Theorem

For any minimum cost network flow problem (LP), if all supplies (b_i) and capacities (u_{ij}) are integers, then the problem has an optimal solution with integer flow on each arc.

Totally Unimodular Matrix

Definition

A matrix A is said to be *totally unimodular* (TU) if the determinant of each square submatrix of A is either 0, 1, or -1.

Examples:

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Observation: If A is TU, then each element $a_{ij} \in \{-1, 0, 1\}$.

Theorem

In the network flow problem, the matrix of flow conservation constraints is totally unimodular.

balance constraint:

$$\sum_{ji\in A} x_{ji} + b_i = \sum_{ij\in A} x_{ij}, \quad \forall i \in A$$

$$Ax = b$$

$$\underbrace{\sum_{ij\in A} x_{ij}}_{\text{OUT}} - \underbrace{\sum_{ji\in A} x_{ji}}_{\text{IN}} = b_i, \quad \forall i \in A$$

Claim: x_{ij} only appears two times in balance constraints

- For node i, x_{ij} is in the flows out of node i, it has a coefficient $+1$.
- For node j, x_{ij} is in the flows into node i, it has a coefficient -1.
- For any other nodes (except i or j), x_{ij} is not in the balance constraints, it has a coefficient 0.

One Sufficient Condition for TU

Theorem

Let A be an $m \times n$ matrix. Then the following conditions together are sufficient for A to be totally unimodular:

1. *Every column of A contains at most two non-zero entries;*
2. *Every entry in A is 0, $+1$, or -1;*
3. *The rows of A can be partitioned into two disjoint sets B and C such that*

   $(1,1)$ $(-1,-1)$

   (a) *If two non-zero entries in a column of A have the same sign, then the row of one entry is in B, and the row of the other in C;*
   (b) *If two non-zero entries in a column of A have opposite signs, then the rows are both in B, or both in C.*

   $(1,-1)$

Example

$$A = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

B

C

- $B = \{1, 2\}$ and $C = \{3, 4\}$

$$A = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 \end{bmatrix} \end{array}$$

- $B = \{1, 2, 3, 4\}$ and $C = \phi$

Cramer's Rule

$x = B^{-1}b$

Theorem

Let B be a nonsingular $m \times m$ matrix. Let x be a solution to $Bx = b$, then

$$x_j = \frac{\det(B^j)}{\det(B)} \quad \forall j = 1, \dots, m$$

where B^j is B with the j-th column replaced by b.

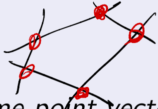
$$B^j = \begin{array}{c} \\ \left[\begin{array}{cccccc} B_1 & B_2 & \cdots & b & \cdots & B_m \end{array} \right] \\ B \end{array}$$

Properties Totally Unimodular Matrix

Theorem

If the matrix A is TU and the vector b has integer entries, then the polyhedron

$$X = \{x \in \mathbb{R}^n : Ax = b,\ x \geq 0\}$$



(if nonempty) has integral extreme points (i.e., each extreme point vector has integral entries).

Corollary

If the matrix A is TU and the vectors a, b, d, f have integer entries, then the following two polyhedra

$$\begin{aligned} X &= \{x \in \mathbb{R}^n : Ax \leq b\} \\ Y &= \{x \in \mathbb{R}^n : a \leq Ax \leq b, d \leq x \leq f\} \end{aligned}$$

(if nonempty) have integral extreme points.

Proof of Thm:

$X = \{x : Ax = b, x \geq 0\}$

The extreme Point of X has the following format:

$$x = \begin{bmatrix} x_B \\ x_N \end{bmatrix} \qquad A = [B \vdots N]$$

extreme pt

$$\begin{cases} x_B = B^{-1}b \\ x_N = 0 \rightarrow \text{integer} \end{cases}$$

$$B x_B = b$$

By Cramer's Rule, any element in x_B is

$$x_{B_j} = \frac{\det(B^j)}{\det(B)} \quad \begin{matrix} \rightarrow \text{integer} \\ \rightarrow 1, -1 \end{matrix}$$

- B is a square submatrix of A, and A is TU, then $\det(B) = 1, -1$

- B is a matrix with elements $1, -1, 0$.
  b, is a vector with integer elements.

So B^j is a matrix with integer elements.

$\Rightarrow \det(B^j)$ is integer

Then $(X_B)_j = \dfrac{\det(B_j)}{\det(B)} \in \mathbb{Z}, \quad \forall j$

$$X_N = 0 \in \mathbb{Z}$$

So the extreme point $X = \begin{bmatrix} X_B \\ X_N \end{bmatrix} \in \mathbb{Z}^n$

Problems Equivalent to Min Cost Network Flow

IPs → Solve in LP

- Transportation
- Transshipment
- Assignment
- Shortest Path
- Max Flow

$\longrightarrow$ Min Cost Network Flow

Solve IP

TU property is uncommon in practice - if one solves the LP relaxation and find the optimal BFS is not integral, it means that TU doesn't hold

- We need to have methods to solve IP systematically (and efficiently)
- The most commonly applied technique, as we will discuss in the following, is the *branch-and-bound* algorithm.

Outline

The Idea

Consider the following example

$$\begin{aligned} \text{maximize} \quad & 8x_1 + 5x_2 \\ \text{subject to} \quad & 9x_1 + 5x_2 \leq 45 \\ & x_1 + x_2 \leq 6 \\ & x_1, x_2 \geq 0, x_1, x_2 \in \mathbb{Z} \end{aligned}$$

LP relaxation

- We solve the LP relaxation and get $\boldsymbol{x}^* = (15/4, 9/4)$
- x_1^* is not feasible (we can't allow 15/4 in the solution)

Solution: Consider two subproblems:

- One with an additional constraint $x_1 \leq 3$.
- One with an additional constraint $x_1 \geq 4$.

The optimal solution to the IP must still be in one of the subproblems, but solutions with $3 < x_1 < 4$ are eliminated

Generally

First, we solve the LP relaxation of the IP:

- If the solution is already integral, then it is optimal to the IP

If the optimal solution to the LP relaxation is $\boldsymbol{x}^*$ and x_i^* is fractional, then branch the problem into the following two:

1. One with an added constraint that $x_i \leq \lfloor x_i^* \rfloor$, we call this (S1)
2. One with an added constraint that $x_i \geq \lceil x_i^* \rceil$, we call this (S2)

Here $\lfloor \cdot \rfloor$ means rounding down, and $\lceil \cdot \rceil$ means rounding up.

- $\lfloor 3.75 \rfloor = 3$, $\lceil 3.75 \rceil = 4$

(IP) $V_{IP} = \max.\ c^T x$

s.t. $Ax = b$

$x \geq 0$

$x \in \mathbb{Z}$

x^*_{IP}

(S$_0$) $V_{S_0} = \max.\ c^T x$

s.t. $Ax = b$

$x \geq 0$

LP relaxation

$\Downarrow$

$x^*_{S_0}$ x^* optimal and x_i^* is fractional

$x_i \leq \lfloor x_i^* \rfloor$ $\qquad$ $x_i \geq \lceil x_i^* \rceil$

LP $\qquad$ LP

(S$_1$) $V_{S_1} = \max.\ c^T x$

s.t. $Ax = b$

$x \geq 0$

$x_i \leq \lfloor x_i^* \rfloor$

$x^*_{S_1}$

(S$_2$) $V_{S_2} = \max.\ c^T x$

s.t. $Ax = b$

$x \geq 0$

$x \geq \lceil x_i^* \rceil$

$x^*_{S_2}$

Compare $\underline{V_{IP},\ V_{S_0}}$, V_{S_1}, V_{S_2}

Pos 1 $\qquad$ Pos 2

———— V_{S_0} $\qquad$ ———— V_{S_0}

———— V_{S_1} $\qquad$ ———— V_{S_1}

———— V_{S_2} $\qquad$ ———— V_{IP}

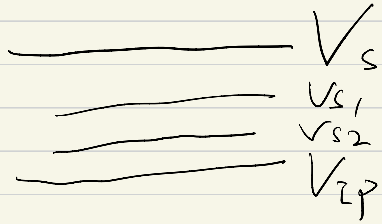
———— V_{IP} $\qquad$ ———— V_{S_2}

claim 1. $V_{S_0} \geq V_{IP}$

$V_{S_0} \geq V_{S_1}$

$V_{S_0} \geq V_{S_2}$

Claim 2: If $x_{s_1}^* \in Z$ and $x_{s_2}^* \in Z$



$\rightarrow$ NOT Possible

IP: $V_{IP} =$ max. $c^T x$

s.t. $Ax = b$

$x \geq 0$

$x \in Z$

X^{IP}

S_1 : $V_{s_1} =$ max. $c^T x$

s.t. $Ax = b$

$x \geq 0$

$x_i \leq \lfloor x_i^* \rfloor$

$$\begin{cases} V_{IP} \geq V_{s_1} \\ V_{IP} \geq V_{s_2} \end{cases} \quad \text{if} \quad \begin{matrix} x_{s_1}^* \in Z \\ x_{s_2}^* \in Z \end{matrix}$$

Proof: $x_{s_1}^* \in Z \Rightarrow x_{s_1}^* \in X^{IP}$

$x_{s_1}^*$ is just a feasible soln to (IP)

Claim 3: If $x_{s_1}^* \in Z$, $x_{s_2}^* \in Z$

$$V^{IP} = \max\{V_{s_1}, V_{s_2}\}$$

- If $V_{s_1} > V_{s_2}$, then $x_{s_1}^*$ is optimal for (IP)
- If $V_{s_2} > V_{s_1}$, then $x_{s_2}^*$ is optimal for (IP)

What Happens after Branching

We get two IP's after branching.

- We then solve (S1) and (S2) and assume we can get the optimal solutions $\boldsymbol{x}_1^*$ and $\boldsymbol{x}_2^*$ with optimal values v_1^* and v_2^* (both (S1) and (S2) are still integer programs).
- If $v_2^* \leq v_1^*$, then $\boldsymbol{x}_1^*$ is the optimal solution to the original IP
- If $v_1^* \leq v_2^*$, then $\boldsymbol{x}_2^*$ is the optimal solution to the original IP

The claim is true because the union of the feasible regions in each branch equals the feasible region of the original problem

How to solve (S1) and (S2)?

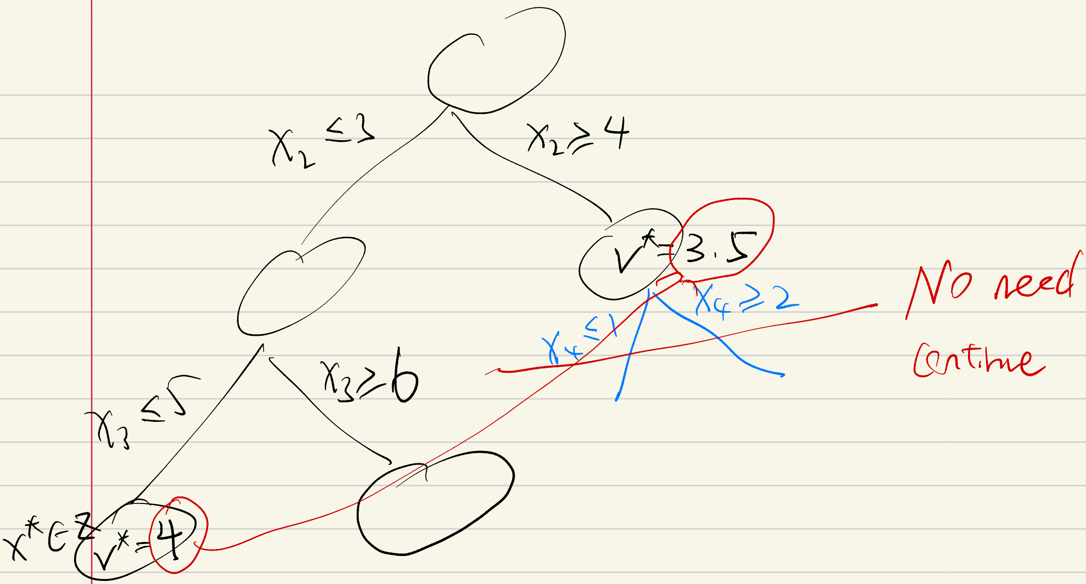
- Use the same idea (solve LP relaxation and further branch)

Bounding

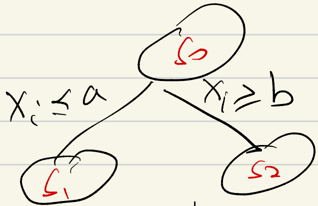
For each branch, we can construct an upper bound and a lower bound for the problem (assume we are solving a maximization problem):

- Upper bound: The LP relaxation solution will be an upper bound — V_{S_0} The objective value of any integral solution from this node must be lower than the optimal value of the LP relaxation
- Lower bound: The objective value of any feasible (integral) from Sub Problem solution is a lower bound for the optimal value — The optimal solution of the IP must be no less than the objective value achieved by any feasible solution
- When at a certain node, the optimal value of the LP relaxation of this branch is even less than the current lower bound. Then we should abandon this branch
- These results will be the opposite if we are minimizing

We will use the bounding steps to *prune* unnecessary computations (or in other words, remove unnecessary branches)



claim:



In branch-and-bound, the next nodes solve problems with new constraints added to above node problem, so subproblem's optimal val is getting worse compared to above problems.

Example

Consider the earlier example

$$\begin{aligned} \text{maximize} \quad & 8x_1 + 5x_2 \\ \text{subject to} \quad & 9x_1 + 5x_2 \le 45 \\ & x_1 + x_2 \le 6 \\ & x_1, x_2 \ge 0, x_1, x_2 \in \mathbb{Z} \end{aligned}$$

- We solve the LP relaxation and get $\boldsymbol{x}^* = (15/4, 9/4)$
- We branch for x_1

We solve two subproblems:

- One with an additional constraint $x_1 \le 3$. We call it (S1)
- One with an additional constraint $x_1 \ge 4$. We call it (S2)

S_0

$$\begin{aligned} \max.\ & 8x_1 + 5x_2 \\ \text{s.t.}\ & 9x_1 + 5x_2 \le 45 \\ & x_1 + x_2 \le 6 \\ & x_1, x_2 \ge 0 \end{aligned}$$

$x^* = \left(\frac{15}{4}, \frac{9}{4}\right)$

$x_1 \le 3$

$x_1 \ge 4$

S_1

$$\begin{aligned} \max.\ & 8x_1 + 5x_2 \\ \text{s.t.}\ & 9x_1 + 5x_2 \le 45 \\ & x_1 + x_2 \le 6 \\ & x_1, x_2 \ge 0 \\ & x_1 \le 3 \end{aligned}$$

S_2

$$\begin{aligned} \max.\ & 8x_1 + 5x_2 \\ \text{s.t.}\ & 9x_1 + 5x_2 \le 45 \\ & x_1 + x_2 \le 6 \\ & x_1, x_2 \ge 0 \\ & x_1 \ge 4 \end{aligned}$$

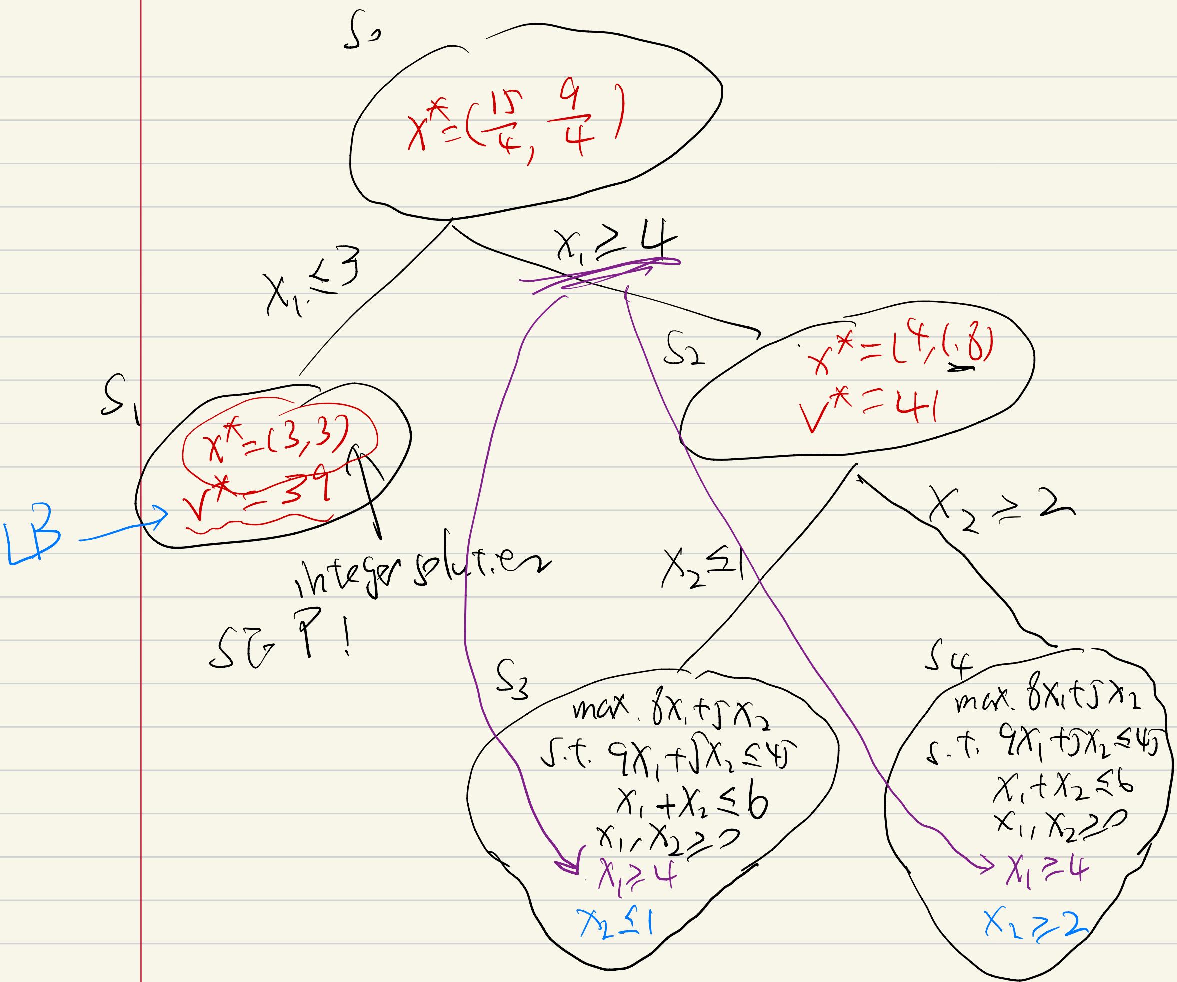
Example Continued

Now we solve (S1) and (S2) respectively

- We solve the LP relaxation of (S1), and get the optimal solution is $(3, 3)$. This is an integral solution, so we are done with this branch (the optimal value is 39)
- We solve the LP relaxation of (S2), the optimal solution is $(4, 1.8)$ (the optimal value is 41).

The solution to (S2) is not integral. We have to do further branching:

- We add a constraint that $x_2 \leq 1$ (S3)
- We add a constraint that $x_2 \geq 2$ (S4)



S0
x* = (15/4, 9/4)
x1 ≤ 3
x1 ≥ 4
S1
x* = (3, 3)
v* = 39
LB
integer solution
SGP!
S2
x* = (4, 1.8)
v* = 41
x2 ≤ 1
x2 ≥ 2
S3
max. 8x1 + 5x2
s.t. 9x1 + 5x2 ≤ 45
x1 + x2 ≤ 6
x1, x2 ≥ 0
x1 ≥ 4
x2 ≤ 1
S4
max. 8x1 + 5x2
s.t. 9x1 + 5x2 ≤ 45
x1 + x2 ≤ 6
x1, x2 ≥ 0
x1 ≥ 4
x2 ≥ 2

Example Continued

S3:

$$\begin{aligned} \text{maximize} \quad & 8x_1 + 5x_2 \\ \text{subject to} \quad & 9x_1 + 5x_2 \leq 45 \\ & x_1 + x_2 \leq 6 \\ & x_1 \geq 4, x_2 \leq 1 \quad x_1, x_2 \geq 0, x_1, x_2 \in \mathbb{Z} \end{aligned}$$

S4:

$$\begin{aligned} \text{maximize} \quad & 8x_1 + 5x_2 \\ \text{subject to} \quad & 9x_1 + 5x_2 \leq 45 \\ & x_1 + x_2 \leq 6 \\ & x_1 \geq 4, x_2 \geq 2 \quad x_1, x_2 \geq 0, x_1, x_2 \in \mathbb{Z} \end{aligned}$$

One can easily see that (S4) is not feasible. Thus we don't need to further consider that.

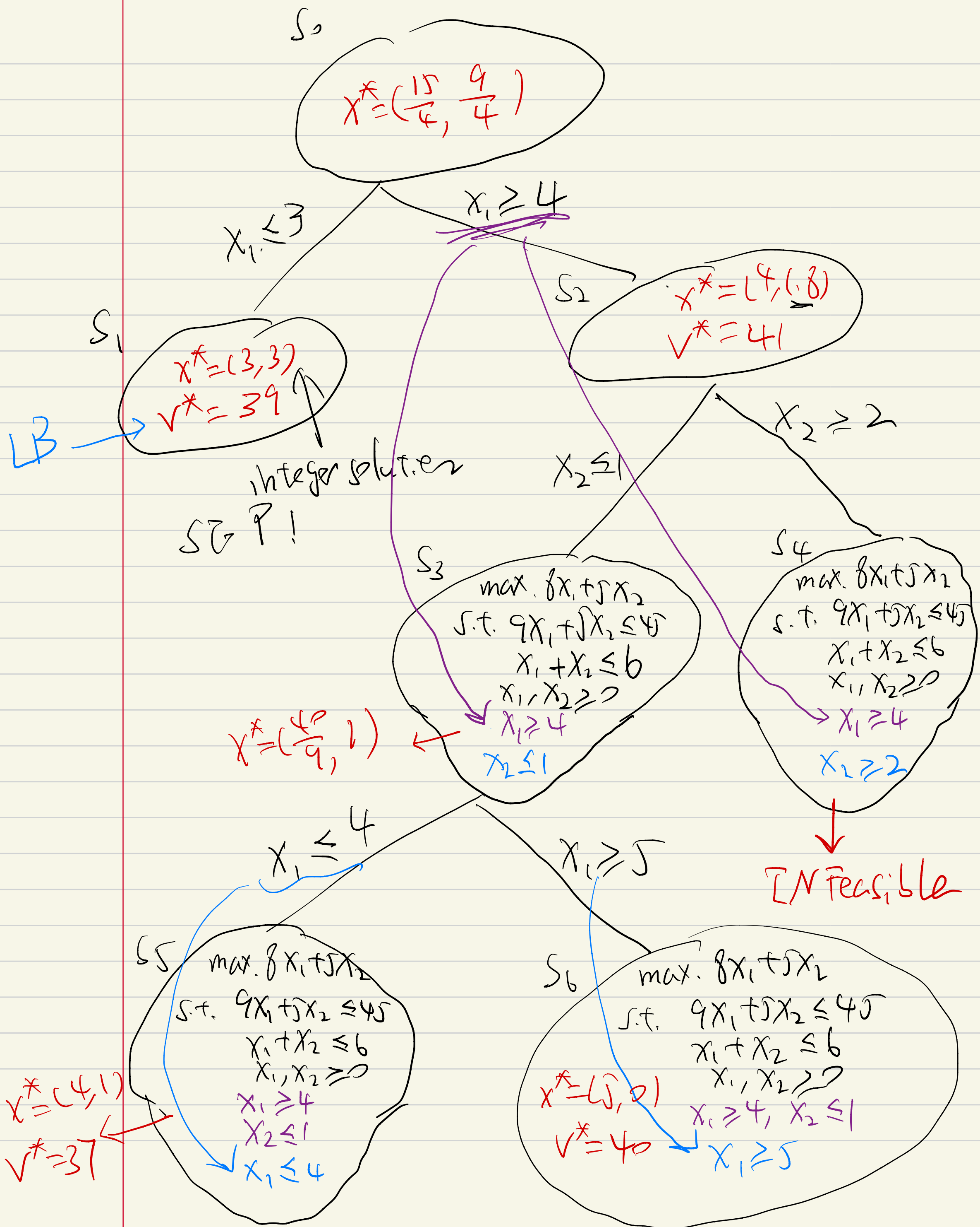
Solve (S3)

We solve (S3) and the optimal solution is $(40/9, 1)$, we have to do further branching:

- Add a constraint $x_1 \leq 4$ (S5)
- Add a constraint $x_1 \geq 5$ (S6)

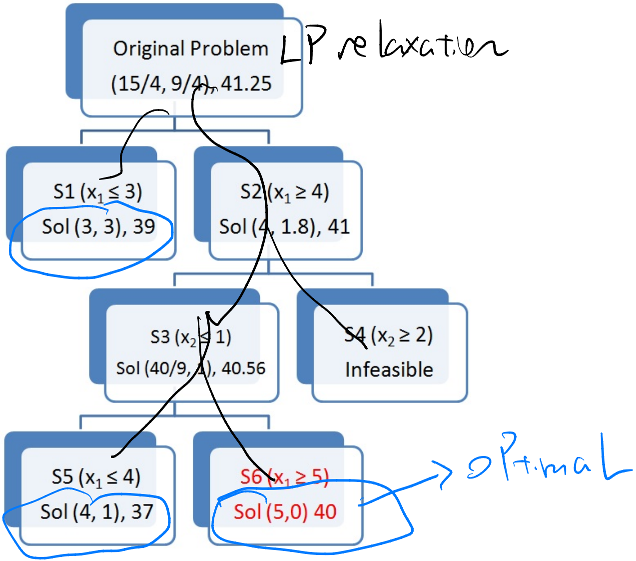
For (S5), x_1 has to be 4 and the optimal solution is $(4, 1)$ with objective value 37 (already integer, so don't need to further branching).

For (S6), the optimal solution is $(5, 0)$, the objective value is 40.



S_0
$x^* = (\frac{15}{4}, \frac{9}{4})$
$x_1 \le 3$
$x_1 \ge 4$
S_1
$x^* = (3,3)$
$v^* = 39$
LB
integer solution
SGP!
S_2
$x^* = (4, 1.8)$
$v^* = 41$
$x_2 \le 1$
$x_2 \ge 2$
S_3
max. $8x_1 + 5x_2$
s.t. $9x_1 + 5x_2 \le 45$
$x_1 + x_2 \le 6$
$x_1, x_2 \ge 0$
$x_1 \ge 4$
$x_2 \le 1$
$x^* = (\frac{40}{9}, 1)$
S_4
max. $8x_1 + 5x_2$
s.t. $9x_1 + 5x_2 \le 45$
$x_1 + x_2 \le 6$
$x_1, x_2 \ge 0$
$x_1 \ge 4$
$x_2 \ge 2$
INFeasible
$x_1 \le 4$
$x_1 \ge 5$
S_5
max. $8x_1 + 5x_2$
s.t. $9x_1 + 5x_2 \le 45$
$x_1 + x_2 \le 6$
$x_1, x_2 \ge 0$
$x_1 \ge 4$
$x_2 \le 1$
$x_1 \le 4$
$x^* = (4,1)$
$v^* = 37$
S_6
max. $8x_1 + 5x_2$
s.t. $9x_1 + 5x_2 \le 45$
$x_1 + x_2 \le 6$
$x_1, x_2 \ge 0$
$x_1 \ge 4,\ x_2 \le 1$
$x_1 \ge 5$
$x^* = (5,0)$
$v^* = 40$

Example


Original Problem
(15/4, 9/4), 41.25
LP relaxation
S1 ($x_1 \le 3$)
Sol (3, 3), 39
S2 ($x_1 \ge 4$)
Sol (4, 1.8), 41
S3 ($x_2 \le 1$)
Sol (40/9, 1), 40.56
S4 ($x_2 \ge 2$)
Infeasible
S5 ($x_1 \le 4$)
Sol (4, 1), 37
S6 ($x_1 \ge 5$)
Sol (5,0) 40
optimal

Branch-and-Bound Method

High-level idea:

- Branching: Divide the feasible region into smaller ones, solve each of them and combine them to find the optimal solution
- Bounding: Use bounds (LP optimal value and feasible solutions obtained) to reduce the number of branches we need to consider

Branch-and-Bound Method

Branching Procedures:

1. Solve the LP relaxation
   - If the optimal solution is integral, then it is optimal to IP
   - Otherwise go to step 2
2. If the optimal solution to the LP relaxation is $\boldsymbol{x}^*$ and x_i^* is fractional, then branch the problem into the following two:
   1. One with an added constraint that $x_i \leq \lfloor x_i^* \rfloor$
   2. One with an added constraint that $x_i \geq \lceil x_i^* \rceil$
3. For each of the two problems, use the same method to solve them, and get optimal solution $\boldsymbol{x}_1^*$ and $\boldsymbol{x}_2^*$ with optimal value v_1^* and v_2^*
   - Compare and get the optimal solution

Branch-and-Bound Method

Bounding procedures (for maximization):

- Any LP relaxation solution can provide an upper bound for each node in the branching process
- Any feasible solution to the IP can provide a lower bound for the entire problem $x^* \in \mathbb{Z}$
- We call the best integer solution obtained for the problem the "incumbent solution"

When at a certain node, the optimal value of the LP relaxation of this branch is even less than the current lower bound (the objective value of the incumbent solution). Then we should abandon this branch (also called prune or fathom that branch)

- No better solution can be obtained from exploring this branch

Bounding is very important for branch-and-bound, it is the key to make it efficient (and practical)

Order of Branching

When we perform branch-and-bound, we may have two choices at each step:

- In the example, if we compute (S2) first, then we get a non-integer solution and thus two branches.
- Then we need to decide if we want to continue with one of the new branches or try (S1) next.

Basically, we need to decide if we want to **go deep** into one branch first or **go wide** to solve all problems on a given level.

Deep or Wide?

In the branch-and-bound algorithm, the best approach is to go deep into the tree, not to go wide:

- Most integer solutions lie deep in the tree. It is good to have integer feasible solutions early, so we can use it in the bounding procedure.
- It is also memory-efficient, since each LP is obtained from its parent by merely adding one constraint.
- It is also easier to code (recursion).

Complexity of Branch-and-Bound

Branch-and-bound is essentially an enumeration method.

- In the worst case, branch-and-bound may need to go through each feasible integer solutions in the region, which is exponential in the problem size
- Remember there is no polynomial-time algorithm for IP

However, branch-and-bound does enumeration in a smart way and typically it only needs to visit a tiny fraction of all solutions

- Much more efficient than explicitly enumerating the solution
- It is one of the most useful practical methods

Branch-and-Bound for Binary Problem

Branch-and-bound can also be used to solve binary linear programs

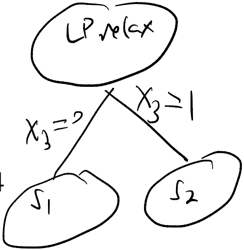
Consider the following knapsack problem:

$$
\begin{aligned}
\text{maximize} \quad & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\
\text{subject to} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14 \\
& x_j \in \{0, 1\}, \quad j = 1, ..., 4.
\end{aligned}
$$

$0 \le x_j \le 1 \quad \forall j$ LP relaxation

Example Continued

The LP relaxation for this IP is:



$$\begin{array}{ll} \text{maximize} & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{subject to} & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & 0 \leq x_j \leq 1, \quad j = 1, \ldots, 4. \end{array}$$

In this case, the optimal solution is $x_1 = 1,\ x_2 = 1,\ x_3 = 0.5,\ x_4 = 0.$ The optimal value is 22.

- We need to do branching for x_3
- Consider two subproblems, one with $x_3 = 1$ (called (S1)), the other with $x_3 = 0$ (called (S2))

binary $\quad x_3 \leq 0 \Rightarrow x_3 = 0$

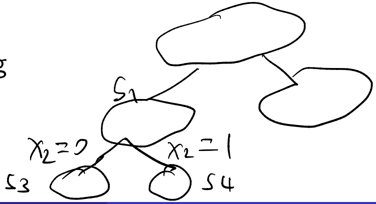
$x_3 \geq 1 \Rightarrow x_3 = 1$

Example Continued

Go deep first, no need solve S_2 after S_1.

Solve the LP relaxation (S1), we get the optimal solution is $x_1 = 1$, $x_2 = 0.714$, $x_3 = 1$ and $x_4 = 0$. And the optimal value of the LP relaxation is 21.85.

- Still fractional. We need to do further branching.
  - $x_2 = 0$ (S3)
  - $x_2 = 1$ (S4)
- However, we obtained one important information: The optimal value of (S1) can't be better than 21.85.
- In fact, since the optimal value of (S1) must be an integer, therefore, it is at most 21
- This is a trick often used in bounding



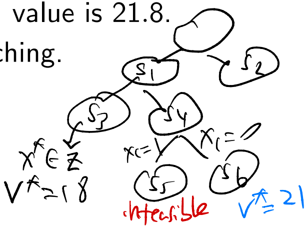
Example Continued

Consider (S3), we solve the LP relaxation, and get the optimal solution is $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, $x_4 = 1$. This is an integral solution with optimal value 18. Therefore, we are done with this branch

- A lower bound of 18 is obtained
- The optimal value of the original IP is at least 18
- If we solve a later LP relaxation and get the optimal value less than 18, then we don't need to further consider that branch

Consider (S4), we solve the LP relaxation, and get the optimal solution $x_1 = 0.6$, $x_2 = 1$, $x_3 = 1$, $x_4 = 0$ and the optimal value is 21.8.

- Still fractional. We need to do further branching.
    - $x_1 = 1$ (S5)
    - $x_1 = 0$ (S6)



Example Continued

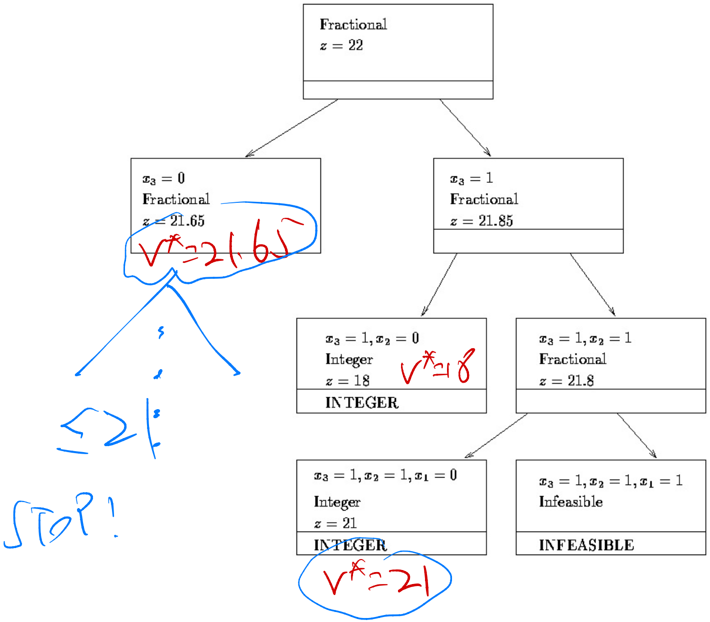
Consider (S5), it is easy to see that (S5) is infeasible. So we don't need to further consider it

Consider (S6), the optimal solution is $x_1 = 0$, $x_2 = 1$, $x_3 = 1$ and $x_4 = 1$. The optimal value is 21.

- Therefore, we get 21 as the optimal value for the first branch (S1) (optimal solution (0,1,1,1)).

Now consider (S2), we compute the LP relaxation and the optimal value is 21.65

- 21.65 is an upper bound on this branch
- Since the optimal value of (S2) must be integer, it means it can't be larger than 21.
- Therefore, no better solution can be obtained in this branch. We don't need to consider it


Fractional
$z = 22$
$x_3 = 0$
Fractional
$z = 21.65$
$v^* = 21.65$
≤ 21
STOP!
$x_3 = 1$
Fractional
$z = 21.85$
$x_3 = 1, x_2 = 0$
Integer
$z = 18$
$v^* = 18$
INTEGER
$x_3 = 1, x_2 = 1$
Fractional
$z = 21.8$
$x_3 = 1, x_2 = 1, x_1 = 0$
Integer
$z = 21$
INTEGER
$v^* = 21$
$x_3 = 1, x_2 = 1, x_1 = 1$
Infeasible
INFEASIBLE

Example Summary

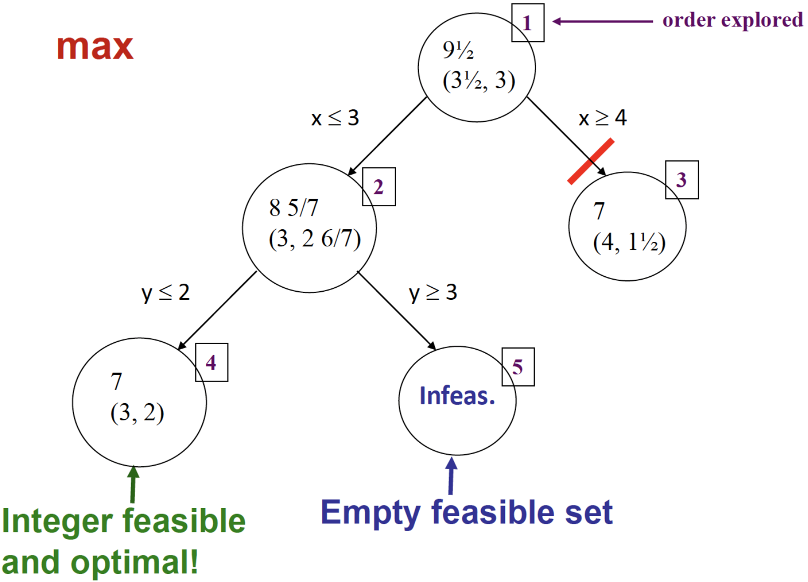
There are 16 (2^4) possible combinations in total, but we don't need to visit all of them

- Bounding is very important, it can greatly reduce the search space
- In the above example, we don't need to consider the $x_3 = 0$ branch because of bounding

Another Example

$$
\begin{aligned}
\text{Maximize} \quad & x + 2y \\
\text{subject to} \quad & -2x + 7y \leq 14 \\
& 6x + 2y \leq 27 \\
& x, y \geq 0 \\
& x, y \in \mathbb{Z}
\end{aligned}
$$

Another Example Solution



The End of Linear Part

- We have completed the linear portion of the course.
- Starting next lecture, we will transition to nonlinear (general) optimization.
- Some people argue that studying linear optimization is outdated, especially since many real-world and machine learning problems are nonlinear. However, this view overlooks the foundational value and broad applicability of linear methods. There is still huge room for research in linear optimization.