LP-based Branch and Bound

Jesper Larsen, David Pisinger[1]

[1]Department of Management Engineering
Technical University of Denmark

DTU Management Engineering
Department of Management Engineering

Learning objectives

After todays lecture you should:

- understand the general idea behind branch and bound as a tool for solving any integer programming problem.
- be able to solve problems using LP-based branch and bound.
- understand the reason for preprocessing and understand the simple rules used in Wolsey.
- be able to perform preprocessing using the simple rules discussed in Wolsey.

What can we solve now...?

- Problems for which we know the ideal formulation
- Problems for which the matrix is TU (and rhs are all integer)
- Problems for which the LP relaxation is IP
- Problems where we have "principle of optimality"
- **... but** we cannot solve a general IP problem.

Divide and conquer

In an effort to solve

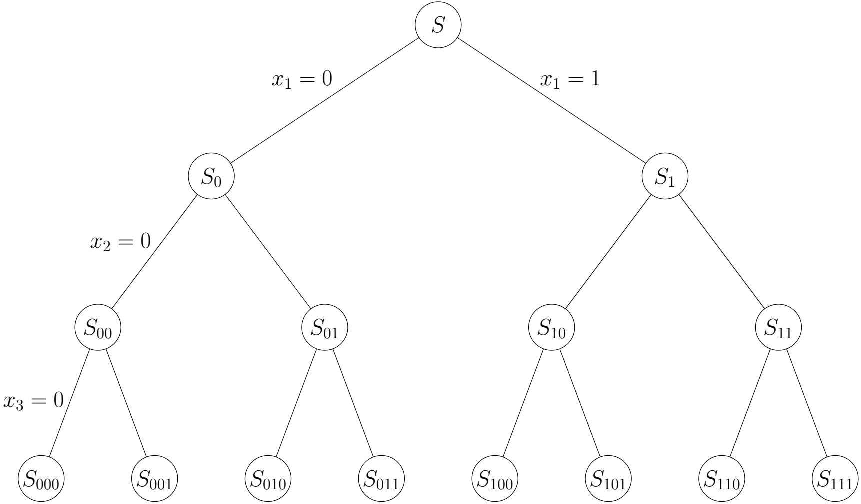
$$z = \max\{cx : x \in S\}$$

an idea would be to divide the problem into smaller problems that are easier to solve.

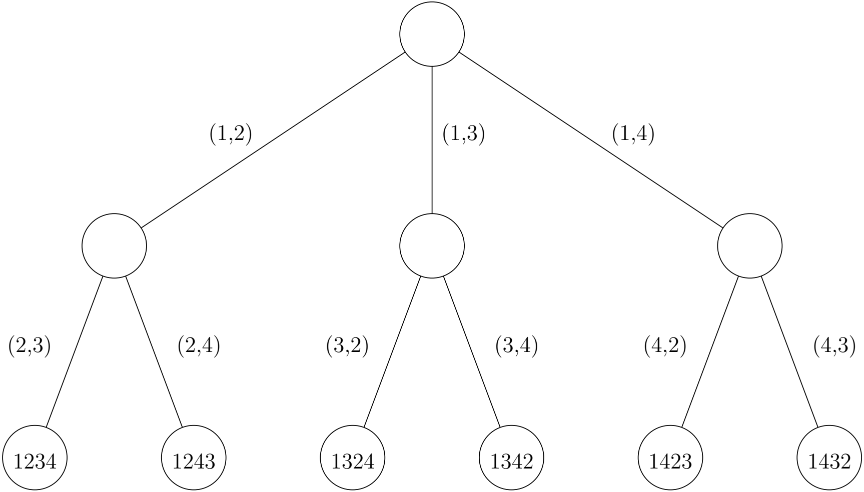
- $S = S_1 \cup S_2 \cup \ldots \cup S_K$ be a **decomposition** of S.
- $z^k = \max\{cx : x \in S_k\}$ for $k = 1, 2, \ldots, K$.
- $z = \max_k z^k$.

Enumeration tree

A way to represent the decomposition approach is via an enumeration tree.



Enumeration tree for the TSP

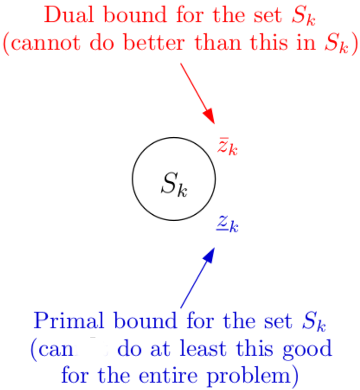


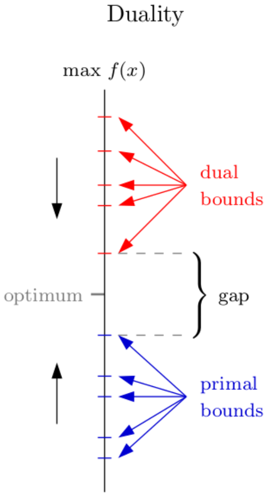
Enumeration

In order to overcome large problems we need to do more than just divide and solve leaf nodes.

- How can we put together bound information?
- How can we use some bounds on the values of $\{z^k\}$ intelligently?

Bounds (Maximization)



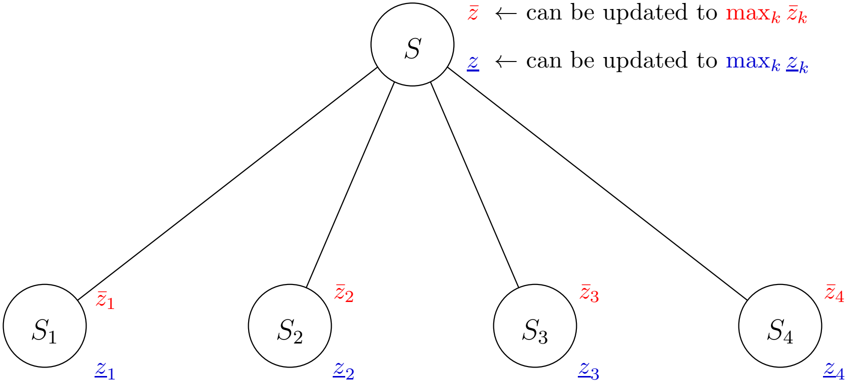


Bounds (Maximization)

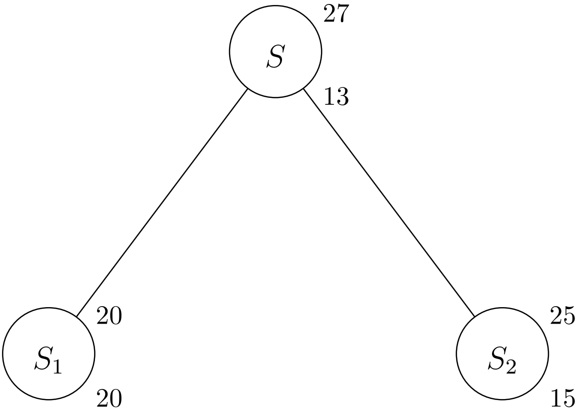
Considering a maximization problem. Let $S = S_1 \cup S_2 \cup \ldots \cup S_K$ be a **decomposition** of S, and let $\bar{z}^k$ be an upper bound on z^k and $\underline{z}^k$ be a lower bound on z^k. Then

- $\bar{z} = \max_k \bar{z}^k$ is an upper bound on z.
  - Only $\bar{z}$ is definitely an upper bound.
- $\underline{z} = \max_k \underline{z}^k$ is a lower bound on z.
  - In fact all $\underline{z}^k$'s are lower bounds but $\underline{z}$ is the tightest one.

Bounds (Maximization)

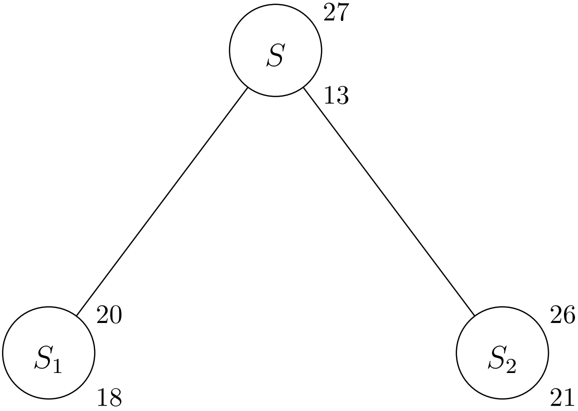


Pruned by optimality (Maximization)



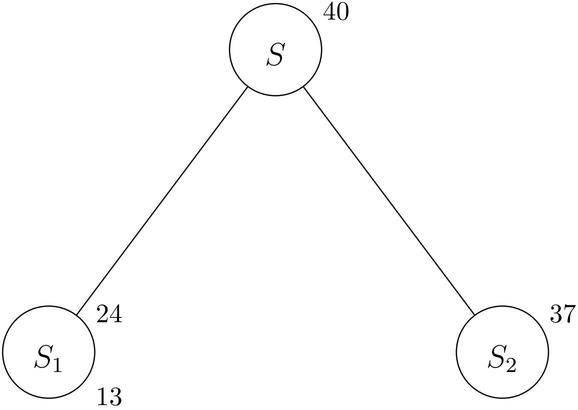
- $\bar{z}_1 = \underline{z}_1 = 20 = z_1$ — no reason to examine S1.

Pruned by bound (Maximization)



- $\bar{z} = \max_k \bar{z}^k = \max\{20, 26\} = 26$.
- $\underline{z} = \max_k \underline{z}^k = \max\{18, 21\} = 21$.
- $\bar{z}_1 = 20, \underline{z}_2 = 21$ — no reason to examine S1.

No pruning possible (Maximization)



- $\bar{z} = \max_k \bar{z}^k = \max\{24, 37\} = 37.$
- $\underline{z} = \max_k \underline{z}^k = \max\{13, -\} = 13.$
- No conclusions can be drawn – we have to examine S1 and S2.

Pruning possibilities (Maximization)

Based on the three cases we have:

- Pruning by optimality: $z^t = \max\{cx : x \in S_t\}$ has been solved.
- Pruning by bound: $\bar{z}^t \leq \underline{z}$
- Pruning by infeasibility $S_t = \emptyset$

Otherwise: Branching

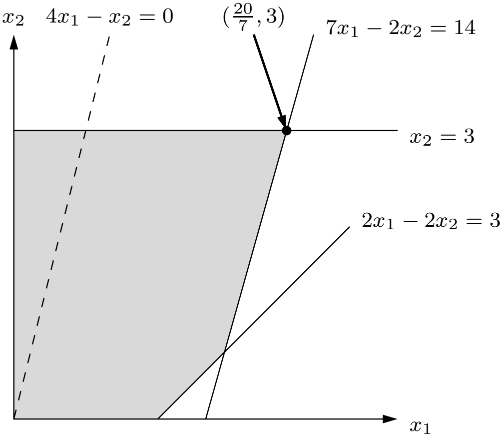
Example

Let us use branch and bound to solve the following integer programming problem:

$$
\begin{aligned}
z = \max \quad 4x_1 - x_2 & \\
7x_1 - 2x_2 & \leq 14 \\
x_2 & \leq 3 \\
2x_1 - 2x_2 & \leq 3 \\
x \in \mathbb{Z}^2_+ &
\end{aligned}
$$

For generating upper bounds we will use the **LP relaxation**.

Example



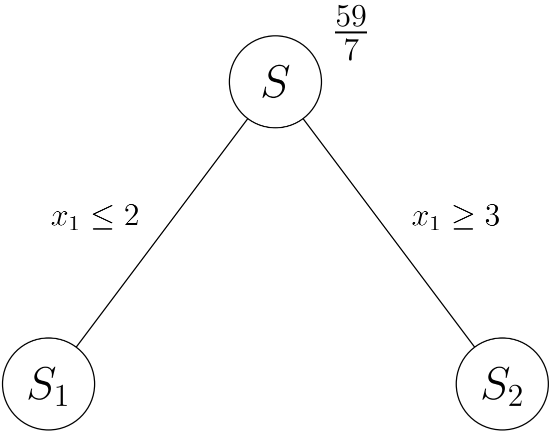
Investigating the root node

- Optimal solution for the LP is $z = \frac{59}{7}$.
- $(x_1, x_2) = (\frac{20}{7}, 3)$.
- Clearly not an integer solution, so how do we proceed?

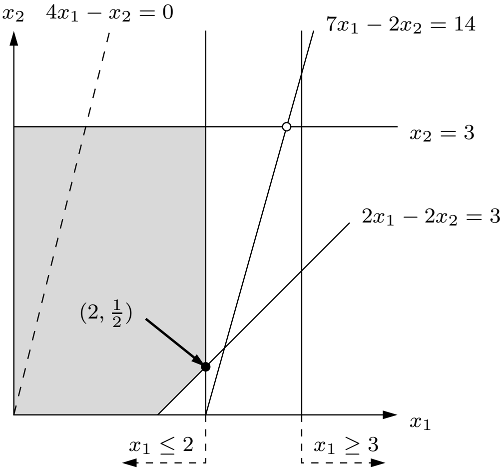
Branching

- As $\underline{z} < \bar{z}$ we need to branch and bound.
- To branch identify an integer variable that is fractional. Define two subproblems:
  - $S_1 = S \cap \{x : x_j \leq \lfloor \bar{x}_j \rfloor\}$.
  - $S_2 = S \cap \{x : x_j \geq \lceil \bar{x}_j \rceil\}$.
- Note that: $S = S_1 \cup S_2, S_1 \cap S_2 = \emptyset$
- Furthermore note:
  - $\bar{x}$ of LP(S) is not feasible in either LP(S_1) or LP(S_2).
  - This implies that if there is no degeneracy so $\max\{\bar{z}_1, \bar{z}_2\} < \bar{z}$

Branching



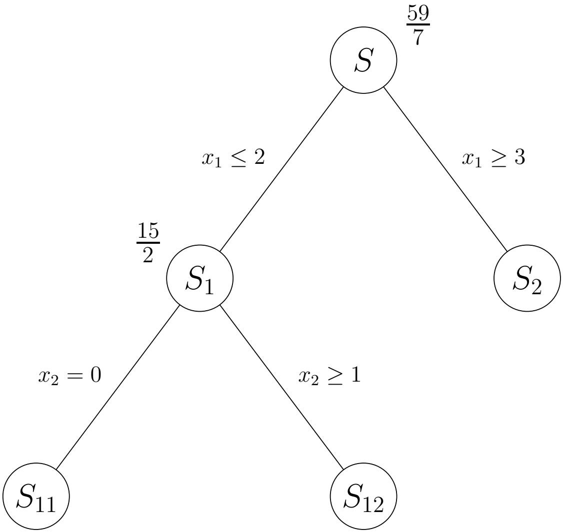
Continue the search....



Continue the search....

- Investigate S_1, so add $x_1 \leq 2$ to the LP and re-optimize.
- $(x_1, x_2) = (2, \frac{1}{2})$.
- Solution not integer so we have to branch (we do it on x_2)

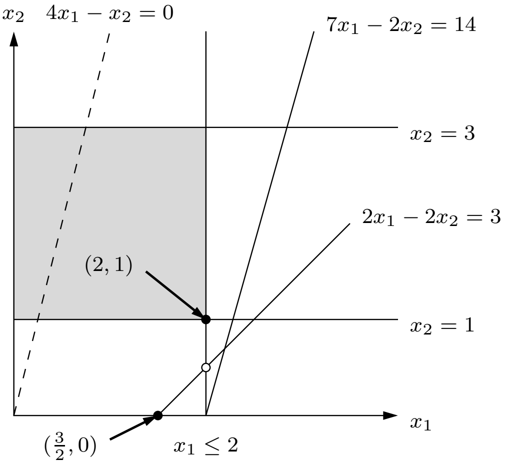
And the search goes on....



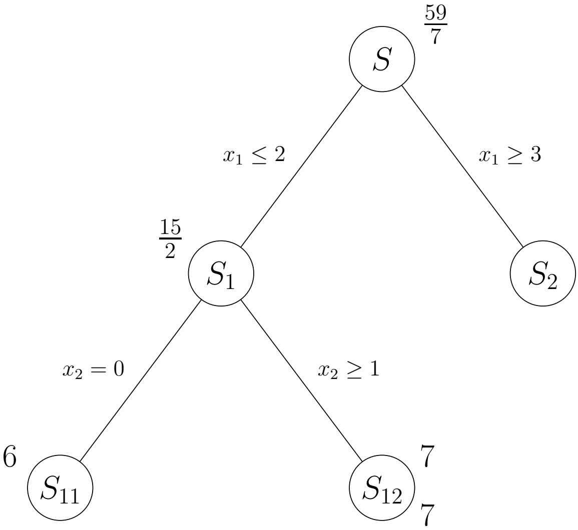
And the search goes on....

- Investigate S_2, so add $x_1 \geq 3$ to the LP and re-optimize.
- By inspection we see that this we can prune by infeasibility.
- So now only the branches of S_1 remains.

And the search goes on....



And the search goes on....



After some pruning we stop

- Investigate S_{12}. LP solution is $z = 7$ with $x_1 = 2$ and $x_2 = 1$. Solution integer $\longrightarrow$ prune by optimality.
- Investigate S_{11}. LP solution is $z = 6$ with $x_1 = \frac{3}{2}$ and $x_2 = 0$.
- S_{11} can now be pruned, since $\bar{z}_{11} = 6 \leq 7 = \underline{z}_{12}$.

Practical aspects

- **Storing the tree:** List of *active* nodes, best known dual bound, variable lower and upper bounds, optimal/near-optimal basis.
- **How to bound:** LP-relaxation and LP-solver.
- **How to branch:**
  - Branch on most *fractional* variable.
  - Branch on least *fractional* variable.
  - "Estimate the cost of forcing x_j to become integer."
- **How to choose a node:** Next time

Modelling with "big M"

- What looks innocent from an IP point of view can deteriorate performance as it results in weaker LP bounds.

Uncapacitated Lot-sizing		
M	LB	Gap (in %)
$M = 1000000$	408.001	4.67
$M = 100000$	408.112	4.67
$M = 1000$	409.12	4.41
$M = 100$	419.12	2.07
$M = 60$	426.67	1.31

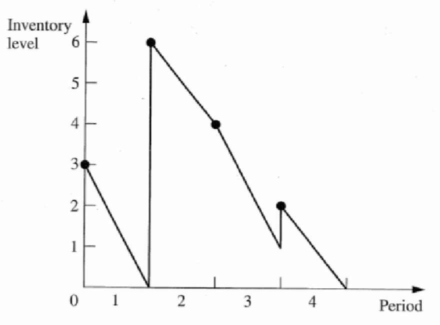
- Larger bounds create more nodes in the tree. For $M = 100$ the tree had 34 nodes and for $M = 10000$ the tree had 211 nodes.

Uncapacitated Lot Sizing

Demand each month

i	1	2	3	4
d_i	3	2	3	2

- fixed startup cost $f = 20$
- holding cost $h = 2$



Uncapacitated Lot Sizing

MIP model

$$
\begin{array}{lll}
\min & \sum_{t=1}^{n} hs_t + fy_t & \\
\text{s.t.} & s_{t-1} + x_t = d_t + s_t, & t = 1, \ldots, n \\
& x_t \leq My_t, & t = 1, \ldots, n \\
& y_t \in \{0, 1\}, x_t, s_t \geq 0 & t = 1, \ldots, n
\end{array}
$$

where $y_t = 1$ if produce in period t

$$x_t > 0 \Rightarrow y_t = 1$$

Modelling with "big M"

IP-solution

- Optimal solution $x_1 = 10, x_2 = 0, x_3 = 0, x_4 = 0$ and $z = 48$

LP-solution

- $x_1 = 3, x_2 = 2, x_3 = 3, x_4 = 2$
- $M = 10$: $y_1 = 0.3, y_2 = 0.2, y_3 = 0.3, y_4 = 0.2$ and $\underline{z} = 20$
- $M = 100$: $y_1 = 0.03, y_2 = 0.02, y_3 = 0.03, y_4 = 0.02$ and $\underline{z} = 2$
- $M = 1000$: $y_1 = 0.003, y_2 = 0.002, y_3 = 0.003, y_4 = 0.002$ and $\underline{z} = 0.2$

- $M_1 = 10, M_2 = 7, M_3 = 5, M_4 = 2$ then
  $x_1 = 3, x_2 = 2, x_3 = 5, x_4 = 0$
  $y_1 = 0.3, y_2 = 0.29, y_3 = 1, y_4 = 0$ and $\underline{z} = 35.7$

Preprocessing

Idea: Detect and eliminate redundant constraints and variables, and tighten bounds where possible.

- Tightening bounds: use known bounds on some variables to tighten bounds on others.
- Redundant constraints
- Variable fixing (by duality)

Tightening bounds

Consider the IP problem:

$$
\begin{array}{lll}
\max & 2x_1 + x_2 - x_3 & \\
\text{s.t.} & 5x_1 - 2x_2 + 8x_3 & \leq 15 \\
& 8x_1 + 3x_2 - x_3 & \geq 9 \\
& x_1 + x_2 + x_3 & \leq 6 \\
& 0 \leq x_1 \leq 3 & \\
& 0 \leq x_2 \leq 1 & \\
& 1 \leq x_3 & \\
& x \in \mathbb{Z}^4 &
\end{array}
$$

Tightening bounds, implementation in solver

Change objective, and solve LP-relaxation

$$
\begin{aligned}
\max \quad & x_1 \\
\text{s.t.} \quad & 5x_1 - 2x_2 + 8x_3 \leq 15 \\
& 8x_1 + 3x_2 - x_3 \geq 9 \\
& x_1 + x_2 + x_3 \leq 6 \\
& 0 \leq x_1 \leq 3 \\
& 0 \leq x_2 \leq 1 \\
& 1 \leq x_3
\end{aligned}
$$

Bounds:

	lower	upper
x_1	0.875	1.8
x_2	0	1
x_3	1	1.536

These bounds do not tighten formulation (all info was in LP-model)

Tightening bounds, implementation in solver

Change objective, and solve LP-relaxation

$$
\begin{array}{lrl}
\max & x_1 & \\
\text{s.t.} & 5x_1 - 2x_2 + 8x_3 & \leq 15 \\
& 8x_1 + 3x_2 - x_3 & \geq 9 \\
& x_1 + x_2 + x_3 & \leq 6 \\
& 0 \leq x_1 \leq 3 & \\
& 0 \leq x_2 \leq 1 & \\
& 1 \leq x_3 &
\end{array}
$$

Round up/down:

	lower	upper
x_1	1	1
x_2	0	1
x_3	1	1

Rounding gives tighter formulation

Preprocessing - for BIPs

- Generating logical inequalities
- Combining pairs of logical inequalities
- Simplifying

Logical inequalities

Consider the constraints of a BIP problem:

$$
\begin{aligned}
7x_1 + 3x_2 - 4x_3 - 2x_4 &\leq 1 \\
-2x_1 + 7x_2 + 3x_3 + x_4 &\leq 6 \\
-2x_2 - 3x_3 - 6x_4 &\leq -5 \\
3x_1 - 2x_3 &\geq -1 \\
x \in \mathbb{B}^4 &
\end{aligned}
$$

Row 1: $x_1 \leq x_3$ $\quad x_1 \leq x_4$ $\quad x_1 + x_2 \leq 1$

Row 2: $x_2 \leq x_1$ $\quad x_2 + x_3 \leq 1$

Row 3: $x_2 + x_4 \geq 1$ $\quad x_3 + x_4 \geq 1$

Row 4: $x_1 \geq x_3$

Logical inequalities

Row 1: $x_1 \leq x_3$ $\quad x_1 \leq x_4$ $\quad x_1 + x_2 \leq 1$

Row 2: $x_2 \leq x_1$ $\quad x_2 + x_3 \leq 1$

Row 3: $x_2 + x_4 \geq 1$ $\quad x_3 + x_4 \geq 1$

Row 4: $x_1 \geq x_3$

Combining logical inequalities

Rows 1 and 4: $x_1 = x_3$

Rows 1 and 2: $x_2 = 0$

Row 3 and above: $x_4 = 1$

Only 2 feasible solutions:

x_1	x_2	x_3	x_4
0	0	0	1
1	0	1	1

Logical inequalities, implementation in solver

For each pair of binary variables x_i and x_j try to fix their values, solve LP-problem, and check whether it is feasible

x_i	x_j	feasible	feasible	feasible	feasible	feasible	feasible
0	0	no	yes	yes	yes	yes	no
0	1	yes	yes	no	yes	no	yes
1	0	yes	yes	yes	no	no	yes
1	1	yes	no	yes	yes	yes	no
		$x_i + x_j \geq 1$	$x_i + x_j \leq 1$	$x_i \geq x_j$	$x_i \leq x_j$	$x_i = x_j$	$x_i = 1 - x_j$