Hands-on Tutorial on Optimization

F. Eberle, R. Hoeksma, and N. Megow

September 26, 2018

Branch & Bound

Branch & Bound: A General Framework for ILPs

- Introduced in the 1960's by Land and Doig
- Based on two principle ideas
  1. Branching
  2. Bounding
- Complete enumeration might be performed

A First Glance

$$\begin{aligned} \min\ & c^T x \\ \text{s.t.}\quad & Ax \leq b \\ & x \geq 0 \\ & x \in \mathbb{Z}^n \end{aligned}$$

Branching:

- Compute a solution x' of the current subproblem
- If $x' \in \mathbb{Z}^n$, compare to the current upper bound.
  - If better, store it as the new current best solution.
  - If worse, prune the current branch.
- If $x' \notin \mathbb{Z}^n$, choose $x'_i \notin \mathbb{Z}$ and create two subproblems
  - Add $x_i \leq \lfloor x'_i \rfloor$.
  - Add $x_i \geq \lceil x'_i \rceil$.

A First Glance

$$\begin{aligned} \min\ & c^T x \\ \text{s.t.}\quad & Ax \leq b \\ & x \geq 0 \\ & x \in \mathbb{Z}^n \end{aligned}$$

Branching:

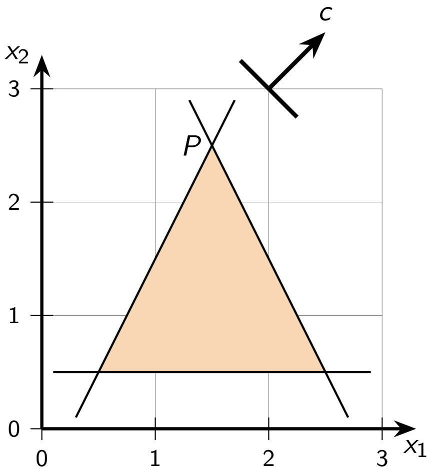
- Compute a solution x' of the current subproblem
- If $x' \in \mathbb{Z}^n$, compare to the current upper bound.
  - If better, store it as the new current best solution.
  - If worse, prune the current branch.
- If $x' \notin \mathbb{Z}^n$, choose $x'_i \notin \mathbb{Z}$ and create two subproblems
  - Add $x_i \leq \lfloor x'_i \rfloor$.
  - Add $x_i \geq \lceil x'_i \rceil$.

Bounding:

If $c^T x' \geq U$, where x' is the LP solution of the current subproblem and U the current upper bound, the current branch can be pruned.

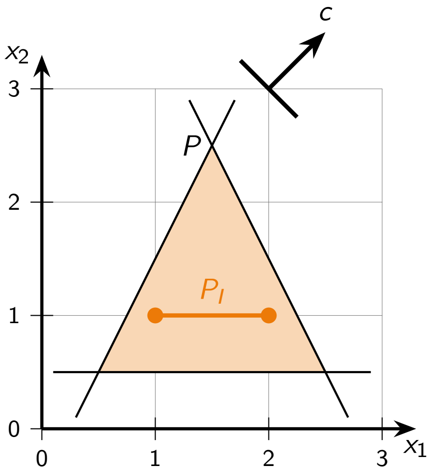
Example: Integer LP

$$
\begin{array}{lrcrcr}
\max & x_1 & & x_2 & & \\
\text{s.t.} & & & x_2 & \geq & \frac{1}{2} \\
& 2x_1 & + & x_2 & \leq & \frac{11}{2} \\
& -2x_1 & + & x_2 & \leq & -\frac{1}{2}
\end{array}
$$



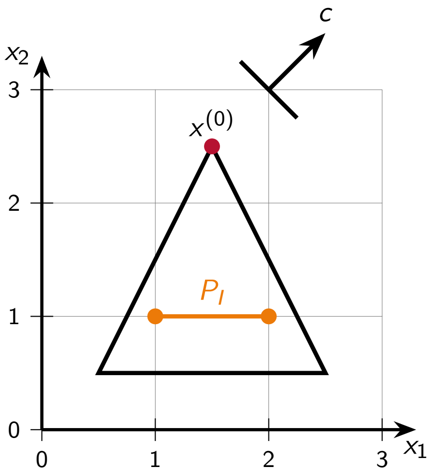
Example: Integer LP

$$
\begin{array}{llllll}
\max & x_1 & & x_2 & & \\
\text{s.t.} & & & x_2 & \geq & \frac{1}{2} \\
 & 2x_1 & + & x_2 & \leq & \frac{11}{2} \\
 & -2x_1 & + & x_2 & \leq & -\frac{1}{2} \\
 & & x & & \in & \mathbb{Z}^2
\end{array}
$$



Example: Branching

1. solve LP relaxation

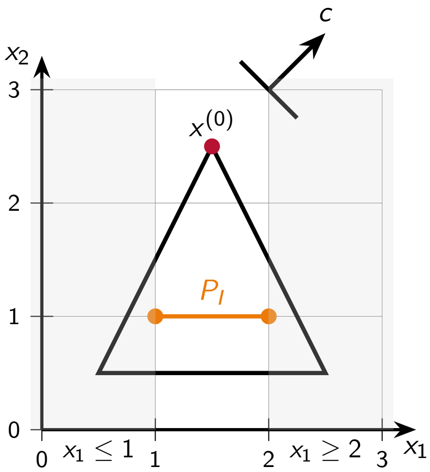


Example: Branching

1. solve LP relaxation
2. $x_i^{(0)} \notin \mathbb{Z} \rightarrow$ branching

$$x_i \leq \lfloor x_i^{(0)} \rfloor$$

$$x_i \geq \lceil x_i^{(0)} \rceil$$



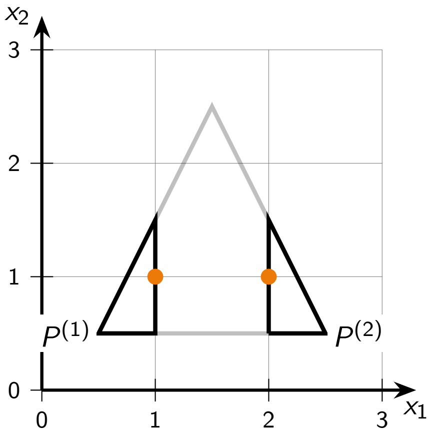
Example: Branching

1. solve LP relaxation
2. $x_i^{(0)} \notin \mathbb{Z} \rightarrow$ branching

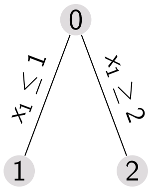
$$x_i \leq \lfloor x_i^{(0)} \rfloor$$

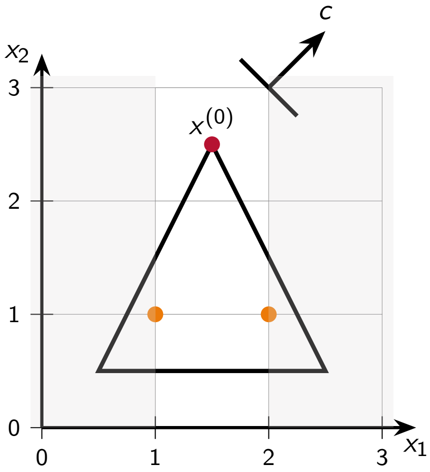
$$x_i \geq \lceil x_i^{(0)} \rceil$$

3. two subproblems

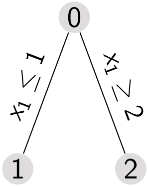


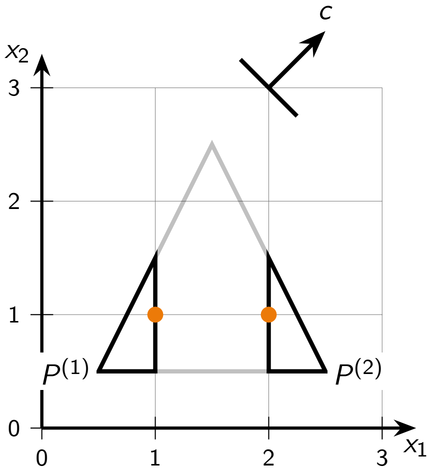
Example: Branch & Bound Tree



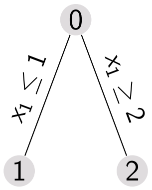


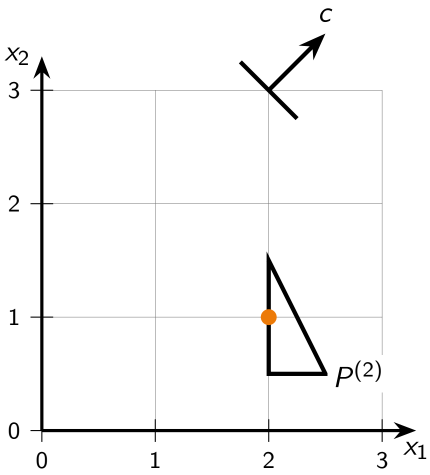
Example: Branch & Bound Tree



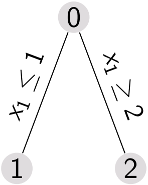


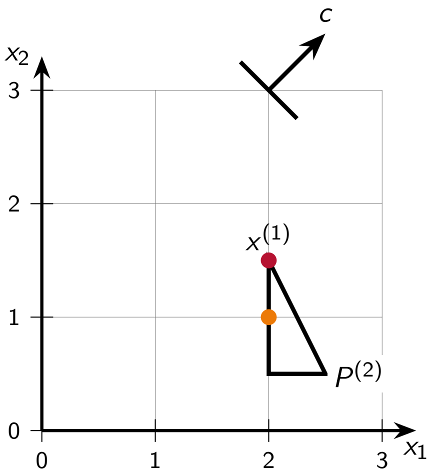
Example: Branch & Bound Tree



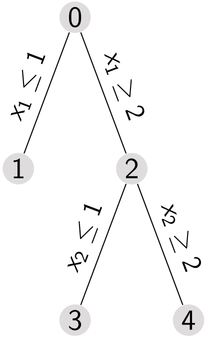


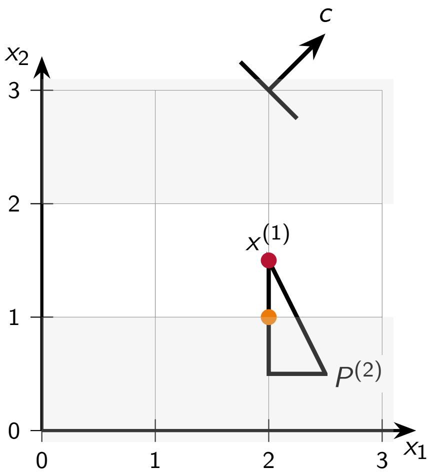
Example: Branch & Bound Tree



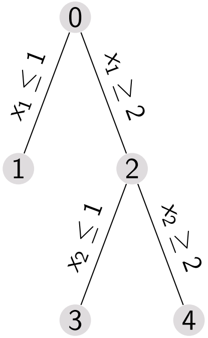


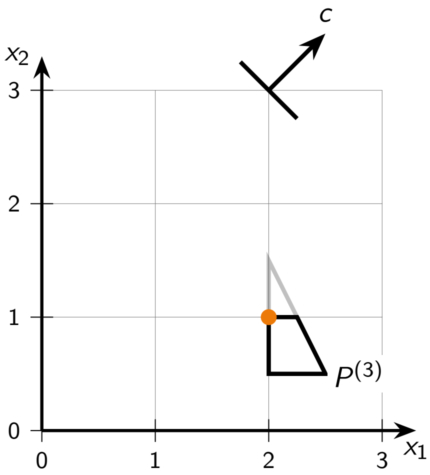
Example: Branch & Bound Tree



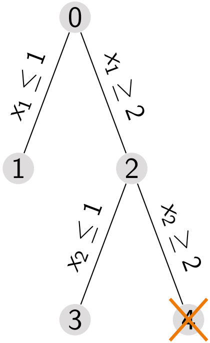


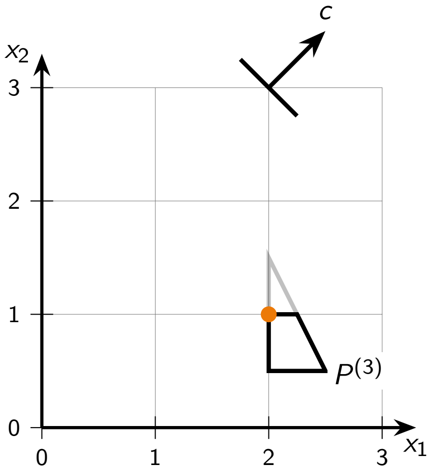
Example: Branch & Bound Tree



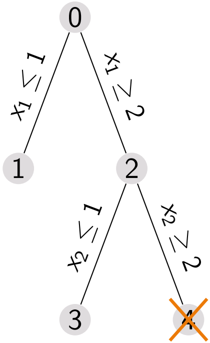


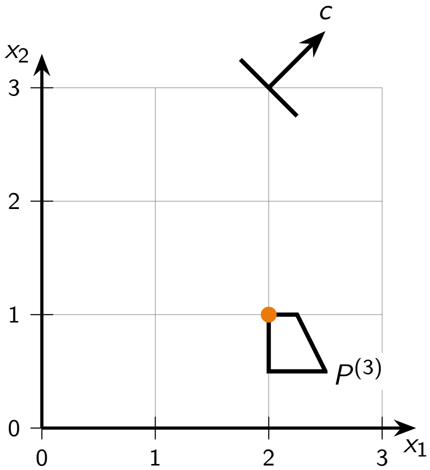
Example: Branch & Bound Tree



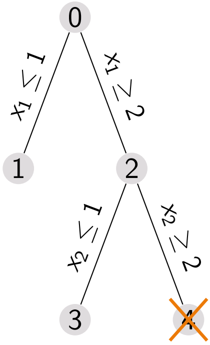


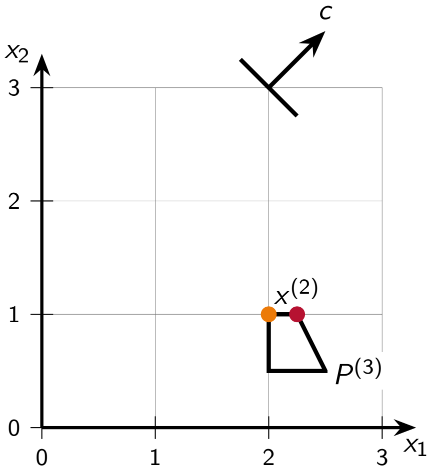
Example: Branch & Bound Tree



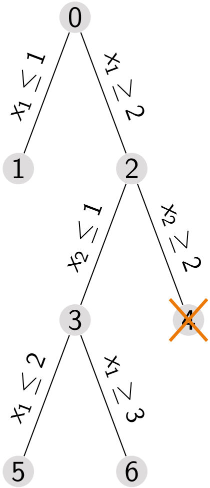


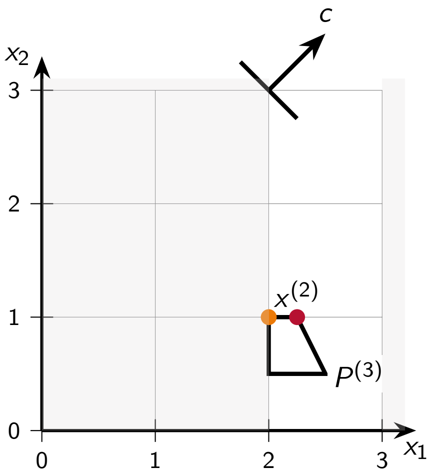
Example: Branch & Bound Tree



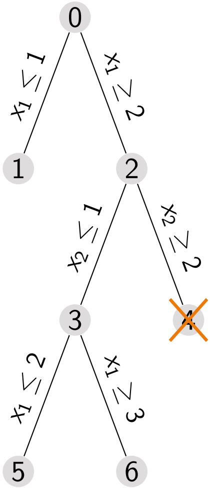


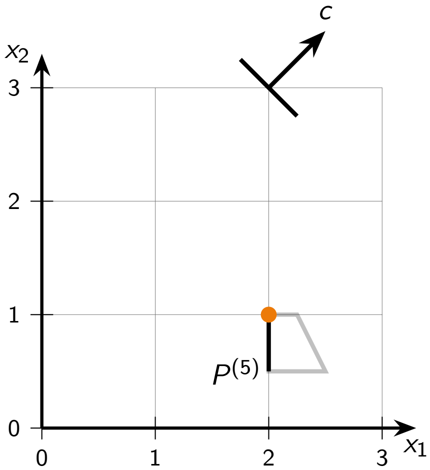
Example: Branch & Bound Tree



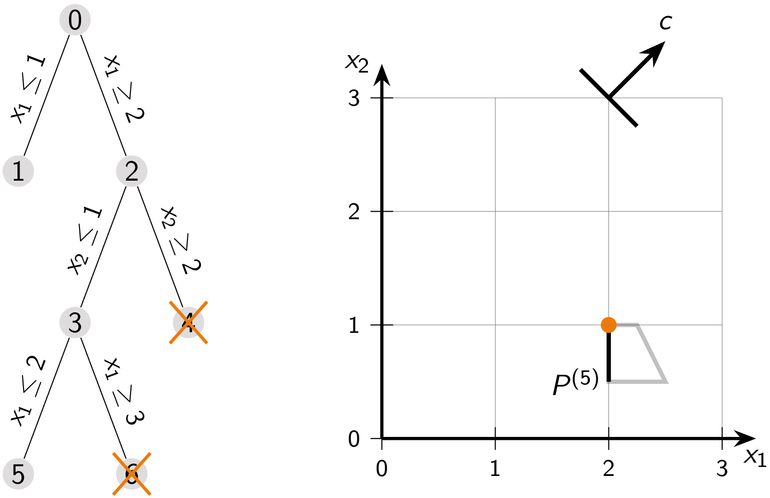


Example: Branch & Bound Tree

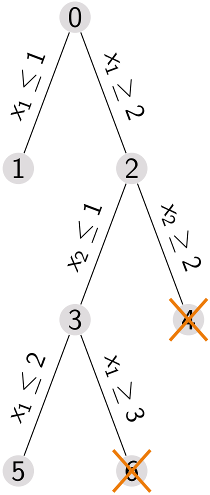


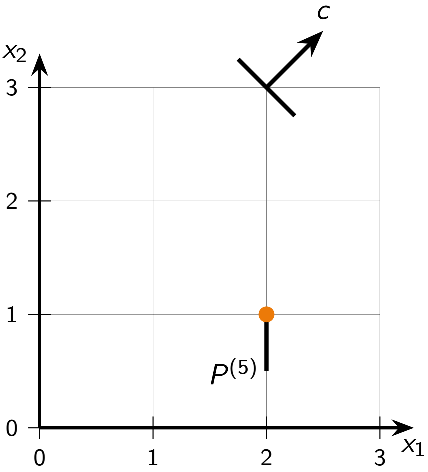


Example: Branch & Bound Tree

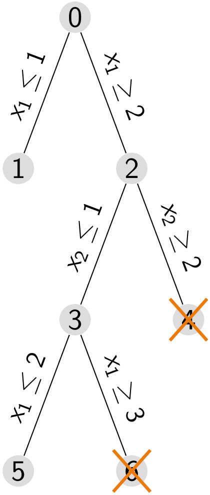


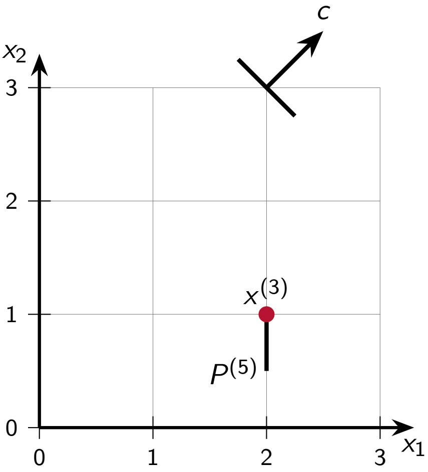
Example: Branch & Bound Tree



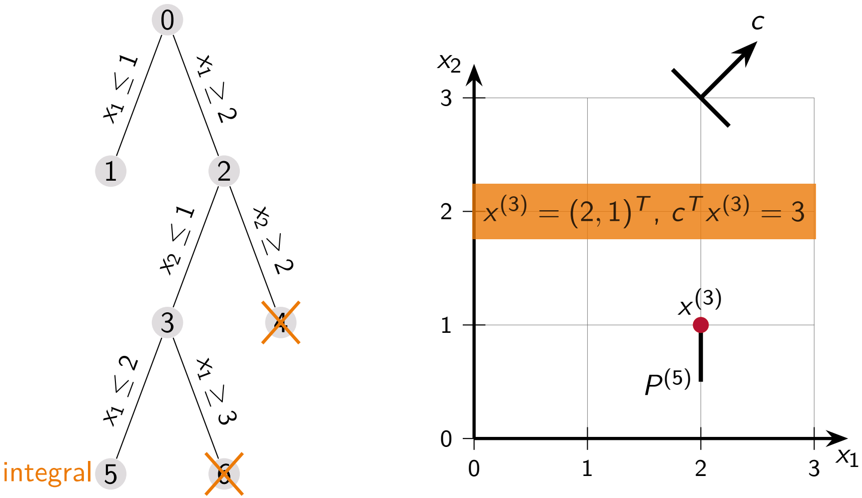


Example: Branch & Bound Tree





Example: Branch & Bound Tree



Bounding and Pruning

- problem: large branch & bound tree

Bounding and Pruning

- problem: large branch & bound tree
- idea: prune nodes

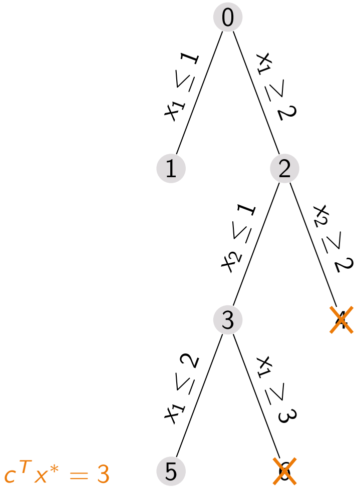
Bounding and Pruning

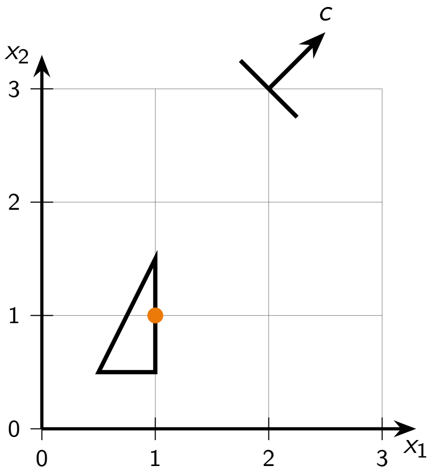
- problem: large branch & bound tree
- idea: prune nodes
  - for each node: determine upper bound S
  - $S \leq c^T x^*$ for best known solution x^*
    $\rightarrow$ prune node

Bounding and Pruning

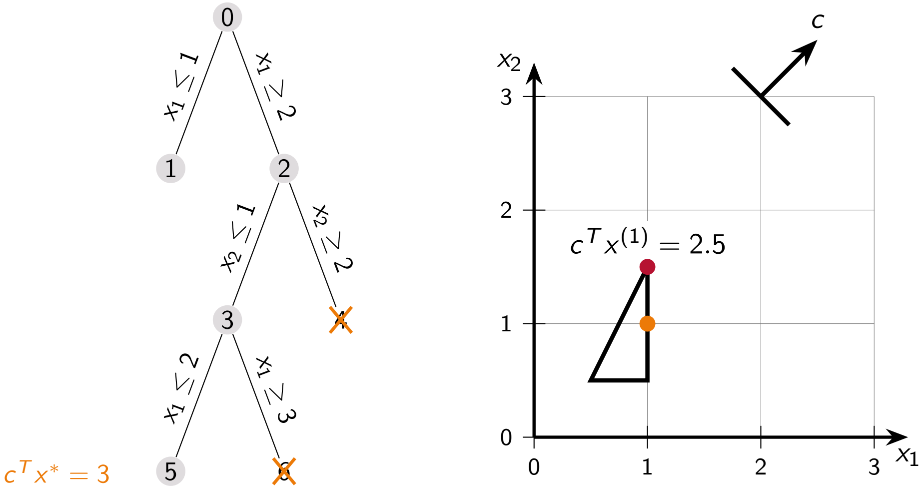
- problem: large branch & bound tree
- idea: prune nodes
  - for each node: determine upper bound S
  - $S \leq c^T x^*$ for best known solution x^*
    $\rightarrow$ prune node
- for max-ILP: upper bound = LP relaxation

Example: Bounding and Pruning

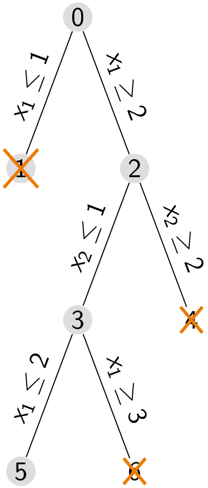




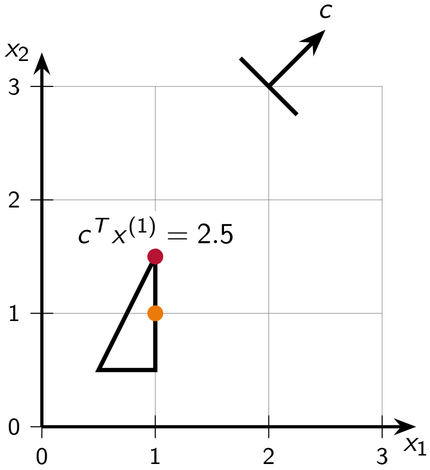
Example: Bounding and Pruning



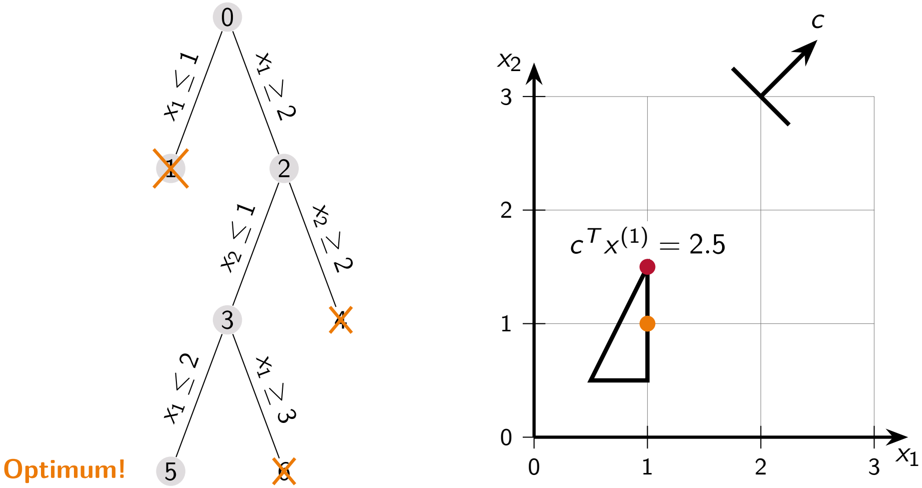
Example: Bounding and Pruning



$c^T x^* = 3$



Example: Bounding and Pruning



Interactive: Knapsack

Problem: Knapsack

Given: $n \in \mathbb{N}$ items with
values $v_i \in \mathbb{N}$ and
weights $w_i \in \mathbb{N}$
knapsack capacity $W \in \mathbb{N}$

Task: Maximize the total value of the knapsack
while not exceeding the capacity

Interactive: Knapsack

Problem: Knapsack

Given: $n \in \mathbb{N}$ items with
values $v_i \in \mathbb{N}$ and
weights $w_i \in \mathbb{N}$
knapsack capacity $W \in \mathbb{N}$

Task: Maximize the total value of the knapsack
while not exceeding the capacity

$$
\begin{array}{lrcl}
\max & \sum_{i=1}^{n} v_i x_i & & \\
\text{s.t.} & \sum_{i=1}^{n} w_i x_i & \leq & W \\
 & x & \in & \{0,1\}^n
\end{array}
$$

Interactive: Knapsack (Multiple Copies)

Problem: Knapsack (Multiple Copies)	
Given:	$n \in \mathbb{N}$ items with values $v_i \in \mathbb{N}$, weights $w_i \in \mathbb{N}$, and number of copies $p_i \in \mathbb{N}$ knapsack capacity $W \in \mathbb{N}$
Task:	Maximize the total value of the knapsack while not exceeding the capacity

Interactive: Knapsack (Multiple Copies)

Problem: Knapsack (Multiple Copies)

Given: $n \in \mathbb{N}$ items with
values $v_i \in \mathbb{N}$,
weights $w_i \in \mathbb{N}$, and
number of copies $p_i \in \mathbb{N}$
knapsack capacity $W \in \mathbb{N}$

Task: Maximize the total value of the knapsack
while not exceeding the capacity

$$\begin{array}{llcll} \max & \sum_{i=1}^{n} v_i y_i & & & \\ \text{s.t.} & \sum_{i=1}^{n} w_i y_i & \leq & W & \\ & y_i & \leq & p_i & \text{for all } i \in [n] \\ & y & \in & \mathbb{N}^n & \end{array}$$

Interactive: TSP

Problem: TSP

Given:	$n \in \mathbb{N}$ cities with distances d_{ij} between $i, j \in [n]$
Task:	Minimize the length of a tour that visits all cities exactly once

Interactive: TSP

Problem: TSP

Given: $n \in \mathbb{N}$ cities with
distances d_{ij} between $i, j \in [n]$

Task: Minimize the length of a tour that visits all cities exactly once

$$
\begin{array}{lrcll}
\max & \sum_{i,j} d_{ij} x_{ij} & & & \\
\text{s.t.} & \sum_j x_{ij} & = & 1 & \text{for all } i \in [n] \\
 & \sum_j x_{ji} & = & 1 & \text{for all } i \in [n] \\
 & x_{ij} & \in & \{0,1\} & \text{for all } i,j \in [n]
\end{array}
$$

Interactive: TSP

Problem: TSP

Given: $n \in \mathbb{N}$ cities with
distances d_{ij} between $i, j \in [n]$

Task: Minimize the length of a tour that visits all cities exactly once

$$
\begin{array}{llcll}
\max & \sum_{i,j} d_{ij} x_{ij} & & & \\
\text{s.t.} & \sum_j x_{ij} & = & 1 & \text{for all } i \in [n] \\
 & \sum_j x_{ji} & = & 1 & \text{for all } i \in [n] \\
 & u_i - u_j + n \cdot x_{ij} & \leq & n - 1 & \text{for all } i \neq j \in \{2, \dots, n\} \\
 & x_{ij} & \in & \{0, 1\} & \text{for all } i, j \in [n]
\end{array}
$$