Summary so far

\[ z = \max \{ c^T x : Ax \leq b, \ x \in \mathbb{Z}_+^n \} \]

- Modeling with IP (and MIP, and BIP) problems
  - Formulation for a discrete set that is a feasible region of an IP
  - Alternative formulations for the same set; strengthening a formulation with valid inequalities
- Bounds on optimal values
  - Primal bounds: from feasible solutions
  - Dual bounds: from relaxations and/or dual problems
- Solving IPs
  - “Easy” IPs (for which an ideal formulation is known and easy to work with) can be solved by solving their LP relaxations with the Simplex method
  - Cutting plane algorithms, e.g., Gomory’s fractional cutting plane algorithm

Most IPs are not “easy,” and Gomory’s algorithm takes a very long time...

### Divide and conquer approach

**maximize** \( 9x_1 + 5x_2 + 6x_3 + 4x_4 \)
subject to
\[
\begin{align*}
6x_1 + 3x_2 + 5x_3 + 2x_4 & \leq 10 \\
-x_1 + x_3 + x_4 & \leq 1 \\
-x_2 + x_4 & \leq 1 \\
x_j & \in \{0, 1\}, \text{ all } j
\end{align*}
\]

**LP-relaxation -- value 16.5**
\[
x = (0.8333, 1, 0, 1) \\
x_1 = 1, \text{ or } x_1 = 0
\]

\[
x_1 = 1 \text{ value } 16.2 \\
x = (1, 0.8, 0, 0)
\]

\[
x_1 = 1, x_2 = 0 \text{ value } 13.8 \\
x = (1, 0, 0.8, 0)
\]

\[
x_1 = 1, x_2 = 0 \text{ value } 9
\]

\[
x_1 = 1, x_2 = 0, x_3 = 0 \text{ INFEASIBLE}
\]

\[
x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 0 \text{ value } 14
\]

\[
x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 0 \text{ INFEASIBLE}
\]
Divide and conquer approach

\[ z = \max\{c^T x : x \in S\} \]

**Idea:** If a problem is hard, partition its feasible region \( S \) into subsets, and solve each of the smaller subproblems

**Proposition 7.1**

Let \( S = S_1 \cup \ldots \cup S_K \) be a decomposition of the set \( S \), and let \( z^k = \max\{c^T x : x \in S_k\} \) for \( k = 1, \ldots, K \). Then \( z = \max_k z^k \).

- We refer to problems \( \max\{c^T x : x \in S_k\} \) for \( k = 1, \ldots, K \) as **subproblems**.
- Solution to the original problem can be found by solving subproblems \( 1, \ldots, K \), and taking the best solution found
  - If any of the subproblems is still too hard to be solved directly, the set \( S_k \) can be further decomposed, etc., thus forming a “tree” of subproblems.
  - Taken to the extreme, this leads to complete enumeration

Implicit (and intelligent) enumeration

- In principle, this approach can lead to complete enumeration of the feasible solutions! — undesirable.
- Use information about subproblems to decide which “branches” of the tree are not worth exploring

\[ z = \max\{c^T x : x \in S\} \]

**Proposition 7.2**

Let \( S = S_1 \cup \ldots \cup S_K \) be a decomposition of \( S \) into smaller sets, and let \( z^k = \max\{c^T x : x \in S_k\} \) for \( k = 1, \ldots, K \), \( U^k \) be an upper bound on \( z^k \), and \( L^k \) be a lower bound on \( z^k \).

- \( U = \max_k U^k \) is an upper bound on \( z \),
- \( L = \max_k L^k \) is a lower bound on \( z \).

\(^a\)The textbook uses notation \( \bar{z} = U \) and \( z = L \)
Example: LP-relaxation based B&B

(IP) $z = \max \begin{align*}
4x_1 - x_2 \\
7x_1 - 2x_2 &\leq 14 \\
x_2 &\leq 3 \\
2x_1 - 2x_2 &\leq 3 \\
x &\in \mathbb{Z}^2_+
\end{align*}$

Let $S$ be the feasible region of (IP)

Begin by solving the LP relaxation of (IP)

- If LP is infeasible, then (IP) is infeasible, done
- If LP has an integer optimal solution, we have the (IP) optimal solution, done
- If optimal solution of LP is not feasible for (IP)
  - We found a dual bound $U \geq z$
  - Do we have a primal bound $L \leq z$?
  - We need to branch, i.e., create subproblems of (IP), and solve them in the same way (recursively)
    - If $\bar{x}$ solves LP and $\bar{x}_j \notin \mathbb{Z}$, decompose $S$ into
      $S \cap \{ x : x_j \leq \lfloor \bar{x}_j \rfloor \}$ and $S \cap \{ x : x_j \geq \lceil \bar{x}_j \rceil + 1 \}$

Pruning subproblems in a B&B tree

Combining bound and feasibility information, there are three potential ways to "prune" a subproblem $S_k$ (i.e., not create any further subproblems from it):$^2$

- By infeasibility: $S_k = \emptyset$
- By optimality: $z^k = \max \{ c^T x : x \in S_k \}$ has been solved (i.e., $U^k = L^k$)
- By bound: If a primal bound $L$ is available, $U^k \leq L$

To be able to prune by bound, keep information on the incumbent: $x^* \in S : L = c^T x^*$

$^2$For briefness, sometimes we refer to $S_k$’s as subproblems, although technically they are feasible regions of subproblems!
A generic Branch and Bound algorithm

Initialization: Add (sub)problem $S$ to the list, incumbent $x^*$ void, $L = -\infty$

1. Select and remove an active subproblem $S_k$ from the list
2. If $S_k$ is infeasible, prune it by infeasibility and go to 6.; otherwise, obtain a dual (upper) bound $U^k$
3. If $U^k \leq L$, prune $S_k$ by bound and go to 6.
4. If solution to $S_k$ can be found, prune by optimality; if $L < z^k$, update the incumbent $x^*$ to be the optimal solution of $S_k$ and $L = z^k$; go to 6.
5. Decompose $S_k$ into further subproblems and add them to the list
6. If the list of active subproblems is empty, stop, report incumbent $x^*$ and $z = L$ as the optimal solution/value

---

Designing a Branch and Bound algorithm

- How to obtain dual bounds? How to obtain primal bounds?
- How to branch? Separate into two parts, or into several parts?
- In what order to examine subproblems?

In the above example, we...
- ...used LP relaxations of subproblems to obtain dual bounds; known feasible solutions of subproblems (if known) to obtain primal bounds
- ...branched by considering fractional optimal values of variables in LP relaxation
- ... chose active subproblems from the list in arbitrarily order
Obtaining bounds

Want: good primal and dual bounds that can be obtained quickly

- A primal bound becomes available whenever a subproblem is pruned by optimality.
- To obtain a dual bound $U^k$, solve a relaxation of the subproblem:
  - an LP relaxation (see example)
  - a combinatorial relaxation, e.g.,
    - Assignment relaxation for TSP
    - 1-tree relaxation for STSP
  - Lagrangian relaxation
- If using LP relaxations:
  - Better formulations $\Rightarrow$ better dual bounds $\Rightarrow$ more pruning by bound! ($U^k \leq L$)
  - Improve each new subproblem’s formulation by adding VI’s
    - VI’s remain valid in further branches, so keep them!
    - This is the so-called “Branch and Cut” algorithm
- If a dual problem is available, it can also be used to obtain dual bounds

How to partition into subproblems

Want:

- create a “balanced” search tree
- ideally, create subproblems in which $x_k$ — solution of the relaxation — is not feasible (to improve dual bound)

If using LP relaxations: Let $x^k \not\in \mathbb{Z}^n$ solve LP relaxation of $S_k$.

- **Variable dichotomy** If $x^k_j \not\in \mathbb{Z}$, create
  
  $S_{k,j} = S_k \cap \{x : x_j \leq \lfloor x^k_j \rfloor \}$ and 
  $S_{k,j+1} = S_k \cap \{x : x_j \geq \lfloor x^k_j \rfloor + 1 \}$

- **BIP: GUB dichotomy** Works with constraints $\sum_{j \in Q} x_j = 1$.
  
  Choose $Q_1 \subset Q$: $0 < \sum_{j \in Q_1} x_j < 1$, and create
  
  $S_{k,0} = S_k \cup \{x : \sum_{j \in Q_1} x_j = 0 \}$ and 
  $S_{k,1} = S_k \cup \{x : \sum_{j \notin Q_1} x_j = 0 \}$.

If using other relaxations: Specific to each problem class; important consideration: should be able to apply the same type of relaxation to the new subproblems.
Example: LP relaxation-based BIP

maximize 9x₁ + 5x₂ + 6x₃ + 4x₄
subject to
6x₁ + 3x₂ + 5x₃ + 2x₄ ≤ 10
x₁ + x₃ ≤ 1
-x₁ + x₃ ≤ 0
-x₂ + x₄ ≤ 1
xⱼ ∈ {0, 1}, all j

LP-relaxation -- value 16.5
x = (0.8333, 1, 0, 1)
L = -∞; U = 16.5

Node 1: x₁ = 1 value 16.2
x = (1, 0.8, 0, 0); L = ∞; U = 16.5
Pruned by optimality

Node 2: x₁ = 1, x₂ = 0 value 13.8
x = (1, 0.8, 0, 0); L = ∞; U = 16.5
Pruned by optimality

Node 3: x₁ = 1, x₂ = 0, x₃ = 1
value 9
L = 9; U = 16.5
Pruned by optimality

Node 4: x₁ = 1, x₂ = 0, x₃ = 0
value 15.2
L = 9; U = 16.5
Pruned by optimality

Node 5: x₁ = 1, x₂ = 1 value 15.2
x = (1, 1, 0.2, 0); L = 9; U = 16.5

Node 6: x₁ = 1, x₂ = 1, x₄ = 0
value 16
x = (1, 1, 0, 0.5); L = 9; U = 16.5

Node 7: x₁ = 1, x₂ = 1, x₃ = 0, x₄ = 0
value 14; L = 14; U = 16.5
Pruned by optimality

Node 8: x₁ = 1, x₂ = 1, x₃ = 1, x₄ = 0
INFEASIBLE

Node 9: x₁ = 1, x₂ = 1, x₄ = 1
INFEASIBLE

Node 10: x₁ = 0
value 9.0
x = (0, 1, 0, 1)
Pruned by optimality

Example: LP relaxation-based BIP

maximize 9x₁ + 5x₂ + 6x₃ + 4x₄
subject to
6x₁ + 3x₂ + 5x₃ + 2x₄ ≤ 10
-x₁ + x₃ ≤ 0
-x₂ + x₄ ≤ 1
xⱼ ∈ {0, 1}, all j

LP-relaxation -- value 16.5
x = (0.8333, 1, 0, 1)
L = -∞; U = 16.5

Node 2: x₁ = 1 value 16.2
x = (1, 0.8, 0, 0); L = ∞; U = 16.5

Node 7: x₁ = 1, x₂ = 0 value 13.8
x = (1, 0.8, 0, 0); L = 14; U = 15.2

Node 3: x₁ = 1, x₂ = 1 value 16
x = (1, 1, 0.5, 0); L = 9; U = 16.2

Node 5: x₁ = 1, x₂ = 1, x₄ = 0
value 15.2
x = (1, 1, 0.2, 0); L = 9; U = 15.2

Node 6: x₁ = 1, x₂ = 1, x₃ = 0, x₄ = 0
value 14; L = 14; U = 16.2

Node 8: x₁ = 1, x₂ = 1, x₃ = 1, x₄ = 0
INFEASIBLE

Node 4: x₁ = 1, x₂ = 1, x₄ = 1
INFEASIBLE
How to choose a node (subproblem)

Recall: the order in which the nodes are considered affects which subproblems can be pruned by bound.

Some possible choices:

- **Depth-first**
  - Reach integer solution quickly (improvement of $L$ value)
  - Fast LP relaxation re-solves
  - But: if unlucky, will explore a wrong branch for a long tie

- **Breadth-first**
  - Get quick estimates of the bound early in the process (improvement of $U$ value)
  - But: too long to find an integer solution

- **Best-Node First**
  - Choosing the node with the best dual bound might lead to finding a good incumbent quickly, thus eliminating the need to solve the other subproblems.
  - But: need to solve relaxations at a lot of nodes to get the bound.

In practice, a mixture of these strategies is used.

A B&B method for TSP

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\
& \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, \ldots, n \\
& \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, n \\
& \text{No subtours} \\
& x_{ij} \in \{0, 1\} \quad \forall i, \forall j
\end{align*}
\]

- **Bounding** Solve the assignment relaxation of the subproblems

- **Branching** Find a subtour constraint that’s violated; suppose the subtour has length $k \leq n/2$. Create $k$ subproblems by adding constraint $x_{ij} = 0$ prohibiting one of the arcs of one of the subtours. (Note: “fixing” $x_{ij} = 0$ can also be interpreted as changing the cost $c_{ij}$ so that it becomes prohibitively high. Thus, the resulting subproblems are still TSPs, and assignment relaxation can be used for each subproblem.)
A combinatorial relaxation of symmetric TSP

**STSP:** a TSP with $c_{ij} = c_{ji}$

- **Bounding** Solve the 1-tree relaxation of the subproblems:

$$z^{STSP} = \min_T \left\{ \sum_{e \in T} c_e : T \text{ is a tour} \right\}$$

$$\geq \min_T \left\{ \sum_{e \in T} c_e : T \text{ is a 1-tree} \right\} = z^{1\text{-tree}}$$

- **Branching**
  - If a 1-tree is not feasible for STSP, what constraint(s) does it violate?
  - Can you suggest a branching strategy?
  - Can you still formulate/solve a 1-tree relaxation of the resulting subproblem?