Reading for This Lecture

- Wolsey Chapter 2
- Nemhauser and Wolsey Sections II.3.1, II.3.6, II.4.1, II.4.2, II.5.4
- “Duality for Mixed-Integer Linear Programs,” Güzelsoy and Ralphs
The Efficiency of Branch and Bound

- In general, our goal is to solve the problem at hand as quickly as possible.
- The overall solution time is the product of the number of nodes enumerated and the time to process each node.
- Typically, by spending more time in processing, we can achieve a reduction in tree size by computing stronger (closer to optimal) bounds.
- This highlights another of the many tradeoffs we must navigate.
- Our goal in bounding is to achieve a balance between the strength of the bound and the efficiency with which we can compute it.
- How do we compute bounds?
  - Relaxation: Relax some of the constraints and solve the resulting mathematical optimization problem.
  - Duality: Formulate a “dual” problem and find a feasible to it.
- In practice, we will use a combination of these two closely-related approaches.
Relaxation

As usual, we consider the MILP

\[
z_{IP} = \max \{ c^\top x \mid x \in S \}, \tag{MILP}
\]

where

\[
P = \{ x \in \mathbb{R}^n \mid Ax \leq b \} \tag{FEAS-LP}
\]

\[
S = P \cap (\mathbb{Z}_+^p \times \mathbb{R}_+^{n-p}) \tag{FEAS-MIP}
\]

**Definition 1.** A relaxation of (MILP) is a maximization problem defined as

\[
z_R = \max \{ z_R(x) \mid x \in S_R \}
\]

with the following two properties:

\[
S \subseteq S_R
\]

\[
c^\top x \leq z_R(x), \ \forall x \in S.
\]
**Importance of Relaxations**

- The main purpose of a relaxation is to obtain an upper bound on $z_{IP}$.
- Solving a relaxation is one simple method of bounding in branch and bound.
- The idea is to choose a relaxation that is much easier to solve than the original problem, but still yields a bound that is “strong enough.”
- Note that the relaxation must be solved to optimality to yield a valid bound.
- We consider three types of “formulation-based” relaxations.
  - LP relaxation
  - Combinatorial relaxation
  - Lagrangian relaxation
- Relaxations are also used in some other bounding schemes we’ll look at.
Aside: How Do You Spell “Lagrangian?”

• Some spell it “Lagrangean.”

• Some spell it “Lagrangian.”

• We ask Google.

• In 2002:
  – “Lagrangean” returned 5,620 hits.
  – “Lagrangian” returned 14,300 hits.

• In 2007:
  – “Lagrangean” returns 208,000 hits.
  – “Lagrangian” returns 5,820,000 hits.

• In 2010:
  – “Lagrangean” returns 110,000 hits (and asks “Did you mean: Lagrangian?”)
  – “Lagrangian” returns 2,610,000 hits.

• In 2014 (strange regression!):
  – “Lagrangean” returns 1,140,000 hits
  – “Lagrangian” returns 1,820,000 hits.

• In 2019 (huh?):
  – “Lagrangean” returns 197,000 hits
  – “Lagrangian” returns 6,680,000 hits.
Obtaining and Using Relaxations

• Properties of relaxations
  – If a relaxation of (MILP) is infeasible, then so is (MILP).
  – If $z_R(x) = c^T x$, then for $x^* \in \arg\max_{x \in S_R} z_R(x)$, if $x^* \in S$, then $x^*$ is optimal for (MILP).

• The easiest way to obtain relaxations of (MILP) is to relax some of the constraints defining the feasible set $S$.

• It is “obvious” how to obtain an LP relaxation, but combinatorial relaxations are not as obvious.
**Example: Traveling Salesman Problem**

The TSP is a combinatorial problem \((E, F)\) whose ground set is the edge set of a graph \(G = (V, E)\).

- \(V = \{1, \ldots, n\}\) is the set of customers.
- \(E\) is the set of travel links between the customers.

A feasible solution is a subset of \(E\) consisting of edges of the form \({i, \sigma(i)}\) for \(i \in V\), where \(\sigma\) is a simple permutation \(V\) specifying the order in which the customers are visited.

**IP Formulation:**

\[
\sum_{j=1}^{n} x_{ij} = 2 \quad \forall i \in V \setminus \{1\}
\]

\[
\sum_{i \in S} x_{ij} \geq 2 \quad \forall S \subset V, |S| > 1.
\]

where \(x_{ij}\) is a binary variable indicating whether \(\sigma(i) = j\).
Combinatorial Relaxations of the TSP

- The Traveling Salesman Problem has several well-known combinatorial relaxations.

- **Assignment Problem**
  - The problem of assigning $n$ people to $n$ different tasks.
  - Can be solved in polynomial time.
  - Obtained by dropping the subtour elimination constraints and the upper bounds on the variables.

- **Minimum 1-tree Problem**
  - A *1-tree* in a graph is a spanning tree of nodes $\{2, \ldots, n\}$ plus exactly two edges incident to node one.
  - A minimum 1-tree can be found in polynomial time.
  - This relaxation is obtained by dropping all subtour elimination constraints involving node 1 and also all degree constraints not involving node 1.
Exploiting Relaxations

• How can we use our ability to solve a relaxation to full advantage?
• The most obvious way is simply to straightforwardly use the relaxation to obtain a bound.
• However, by solving the relaxation repeatedly, we can get additional information.
• For example, we can generate extreme points of $\text{conv}(S_R)$.
• In an indirect way (using the Farkas Lemma), we can even obtain facet-defining inequalities for $\text{conv}(S_R)$.
• We can use this information to strengthen the original formulation.
• This is one of the basic principles of many solution methods.
Lagrangian Relaxation

- A Lagrangian relaxation is obtained by relaxing a set of constraints from the original formulation to improve tractability.
- However, we also try to improve the bound by modifying the objective function, penalizing violation of the dropped constraints.
- Consider a pure IP defined by

\[
\begin{align*}
\max \ & c^\top x \\
\text{s.t.} \ & A'x \leq b' \\
& A''x \leq b'' \\
& x \in \mathbb{Z}_+^n,
\end{align*}
\]

\[(IP)\]

where \( S_R = \{ x \in \mathbb{Z}_+^n \mid A'x \leq b' \} \) bounded and optimization over \( S_R \) is "easy.”

- Lagrangian Relaxation:

\[
LR(u) : z_{LR}(u) = ub'' + \max_{x \in S_R} \{(c - uA'')x\}.
\]
Properties of the Lagrangian Relaxation

- For any \( u \geq 0 \), \( LR(u) \) is a relaxation of (IP) (why?).
- Solving \( LR(u) \) yields an upper bound on the value of an optimal solution.
- Recalling LP duality, one can think of \( u \) as a vector of "dual variables."
- The Lagrangian dual problem is that of determining

\[
\min_{u \geq 0} L(u),
\]

the "best bound" that can be obtained by optimization over \( S_R \).
- This bound is at least as good as the bound yielded by solving the LP relaxation.
- We will examine this problem in much more detail later in the course.
The Lagrangian Dual Function

- We can obtain a dual function from a Lagrangian relaxation by letting

\[ L(\beta, u) = \max_{x \in S_R(\beta')} (c - uA'')x + u\beta'' , \]

where \( S_R(d) = \{x \in \mathbb{Z}^n_+ | A'x \leq d\} \)

- For fixed \( \beta \), this function is the max of linear functions, i.e., a convex cone.

- Then the Lagrangian dual function, \( \phi_{LD} \), is

\[ \phi_{LD}(\beta) = \min_{u \geq 0} L(\beta, u) \]

- This is then the minimum of convex cones and bounds the value function from above (we are in the maximization case here).

- We will see a number of ways of efficiently computing \( \phi_{LD}(b) \) later in the course.
Relaxations from Conceptual Reformulations

- From what we have seen so far, we have two conceptual reformulations of a given integer optimization problem.
- The first is in terms of *disjunction*:

\[
\max \left\{ c^\top x \mid x \in \left( \bigcup_{i=1}^{k} \mathcal{P}_i + \text{intcone}\{r^1, \ldots, r^t\} \right) \right\} \quad \text{(DIS)}
\]

- The second is in terms of *valid inequalities*:

\[
\max \left\{ c^\top x \mid x \in \text{conv}(S) \right\} \quad \text{(CP)}
\]

where *S* is the feasible region.
- In principle, if we had a method for generating either of these reformulations, this would lead to a practical method of solution.
- Instead, we usually begin with a relaxation derived from one of these two reformulations and iteratively approximate the full formulation.
A Generic Algorithmic Framework

• Many algorithms in optimization consist of the iterative solution of a certain relaxation or “dual”.

• The relaxation or dual is improved dynamically until an optimality criterion is achieved.

• A simple algorithm for solving MILPs is to start by solving the LP relaxation to obtain

$$\hat{x} \in \arg\max_{x \in \mathcal{P}} c^\top x$$

and the upper bound $$U = c^\top \hat{x} \geq z_{IP}$$

• Then determine either a valid disjunction or a valid inequality that is violated by \(\hat{x}\) and “add” it to the relaxation.

• Re-solve the strengthened relaxation and continue this process until $$U = z_{IP}$$ (the solution to the relaxation is in \(S\)).

• This vague algorithm is, at a high level, how we solve MILPs and we will see that branch-and-bound fits into this framework.

• The condition that $$U = z_{IP}$$ is the basic optimality condition used in a wide range of optimization algorithms.
The Branch and Bound Tree as a “Meta-Relaxation”

• The branch-and-bound tree itself encodes a relaxation of our original problem, as we mentioned in the last lecture.

• As observed previously, the set \( T \) of leaf nodes of the tree (including those that have been pruned) constitute a valid disjunction, as follows.
  – When we branch using admissible disjunctions, we associate with each \( t \in T \) a polyhedron \( X_t \) described by the imposed branching constraints.
  – The collection \( \{X_t\}_{t \in T} \) then defines a disjunction.

• The subproblem associated with node \( i \) is an integer program with feasible region \( S \cap \mathcal{P} \cap X_t \).

• The problem
  \[
  \max_{t \in T} \max_{x \in \mathcal{P} \cap X_t} c^\top x \tag{OPT}
  \]
  is then a relaxation according to our definition.

• Branch and bound can be seen as a method of iteratively strengthening this relaxation.

• We will later see how we can add valid inequalities to the constraint of \( \mathcal{P} \cap X_t \) to strengthen further.