Computational Integer Programming

Lecture 9: Relaxation and Duality

Dr. Ted Ralphs

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

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

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

where

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

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

Definition 1. *A relaxation of IP is a maximization problem defined as*

$$z_R = \max\{z_R(x) \mid x \in \mathcal{S}_R\}$$

with the following two properties:

$$\begin{aligned} \mathcal{S} &\subseteq \mathcal{S}_R \\ c^\top x &\leq z_R(x), \ \forall x \in \mathcal{S}. \end{aligned}$$

Importance of Relaxations

- The main purpose of a relaxation is to obtain an upper bound on z_{IP}.
- Solving a relaxation is 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.

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 previously, let the set T of leaf nodes of the tree (including those that have been pruned) define a partition of the feasible region, as follows.
- When we branch using admissible disjunctions, the branching constraints define a polyhedral region X_t with respect to each leaf node $t \in T$.
- The collection $\{X_t\}_{t \in T}$ then defines a disjunction.
- The *subproblem* associated with node i is an integer program formulated by the polyhedron $\mathcal{P}_t = \mathcal{P} \cap X_t$.
- The problem

  $$\max_{t \in T} \max_{x \in \mathcal{P}_t} c^\top x \tag{OPT}$$

  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 $\mathcal{P}_t \cap X_t$ to strengthen further.

Obtaining and Using Relaxations

- Properties of relaxations
  - If a relaxation of (MILP) is infeasible, then so is (MILP).
  - If $z_R(x) = c^\top x$, then for $x^* \in \operatorname{argmax}_{x \in S_R} z_R(x)$, if $x^* \in \mathcal{S}$, then x^* is optimal for (MILP).
- The easiest way to obtain relaxations of IP is to drop some of the constraints defining the feasible set $\mathcal{S}$.
- 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 $(E, \mathcal{F})$ whose ground set is the edge set of a graph $G = (V, E)$.

- V 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 σ is a simple permutation V specifying the order in which the customers are visited.

IP Formulation:

$$\begin{array}{rcll} \sum_{j=1}^{n} x_{ij} & = & 2 & \forall i \in N^{-} \\ \sum_{\substack{i \in S \\ j \notin S}} x_{ij} & \geq & 2 & \forall S \subset V,\ |S| > 1. \end{array}$$

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 $\text{conv}(\mathcal{S}_R)$.
- In an indirect way (using the Farkas Lemma), we can even obtain facet-defining inequalities for $\text{conv}(\mathcal{S}_R)$.
- 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.
- However, we also try to improve the bound by modifying the objective function, penalizing violation of the dropped constraints.
- Suppose our IP is defined by

$$\begin{array}{rcl} \max\ c^\top x & & \\ \text{s.t. } A'x & \leq & b' \\ A''x & \leq & b'' \\ x & \in & \mathbb{Z}^n_+ \end{array}$$

where optimizing over $\mathcal{S}_R = \{x \in \mathbb{Z}^n_+ \mid A'x \leq b'\}$ is "easy."

- Lagrangian Relaxation:

$$LR(u) : z_{LR}(u) = \max_{x \in \mathcal{S}_R} \{(c - uA'')x + ub''\}.$$

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 the optimal solution.
- We will show later that this bound is at least as good as the bound yielded by solving the LP relaxation.
- Generally, we try to choose a relaxation that allows $LR(u)$ to be evaluated relatively easily.
- Recalling LP duality, one can think of u as a vector of "dual variables."

A (Very) Brief Tour of Duality

- If we have a dual to (**??**), then we can easily obtain bounds on the value of an optimal solution.
- The advantage of a dual is that we need not solve it to optimality.
- Any feasible solution to the dual yields a valid bound.
- The three main categories of duals used computationally are
  - LP duals
  - Lagrangian duals

A Quick Overview of LP Duality

- We consider the LP relaxation of (MILP) in standard form

$$\left\{ x \in \mathbb{R}^n_+ \;\middle|\; \bar{A}x = b \right\}, \tag{LP}$$

  where $\bar{A} = [A \mid I]$ (the additional columns are for the slack variables).

- Recall that there always exists an optimal solution that is *basic*.

- We construct basic solutions by
  - Choosing a *basis* B of m linearly independent columns of $\bar{A}$.
  - Solving the system $Bx_B = b$ to obtain the values of the *basic variables*.
  - Setting remaining variables to value 0.

- If $x_B \geq 0$, then the associated basic solution is *feasible*.

- With respect to any basic feasible solution, it is easy to determine the impact of increasing a given activity.

- The *reduced cost*

$$\bar{c}_j = c_j - c_B^\top B^{-1} \bar{A}_j.$$

  of (nonbasic) variable j tells us how the objective function value changes if we increase the level of activity j by one unit.

The LP Value Function

- From the resource (dual) perspective, the quantity $u = c_B B^{-1}$ is a vector that tells us the marginal economic value of each resource.
- Thus, the vector u gives us a *price* for each resource.
- This price vector can be seen as the gradients of the *value function*

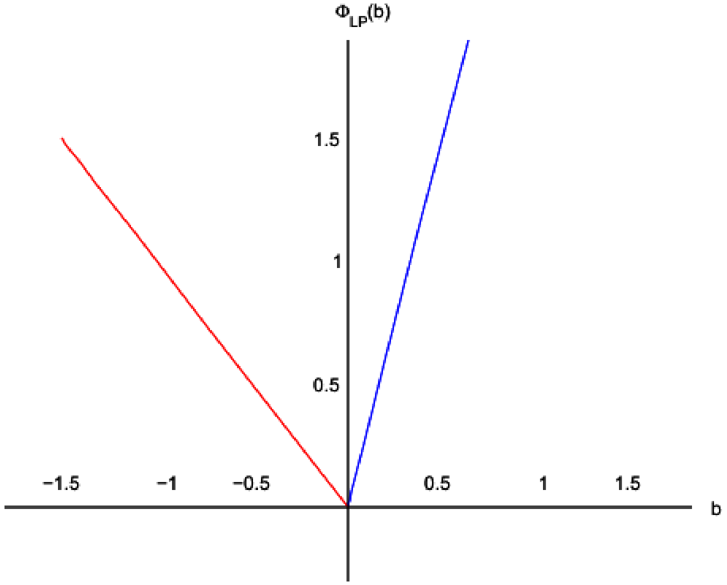
$$\phi_{LP}(\beta) = \max_{x \in \mathcal{S}(\beta)} c^\top x, \qquad \text{(LPVF)}$$

  of an LP, where for a given $\beta \in \mathbb{R}^m$, $\mathcal{S}(d) = \{x \in \mathbb{R}^n_+ \mid \bar{A}x = d\}$.

- We let $\phi_{LP}(\beta) = -\infty$ if $\beta \in \Omega = \{\beta \in \mathbb{R}^m \mid \mathcal{S}(\beta) = \emptyset\}$.
- These gradients can be seen as *linear over-estimators* of the value function.
- The dual problems we'll consider are essentially aimed at producing such over-estimators.
- We'll generalize to *non-linear functions*.

LP Value Function Example

$$\begin{aligned} \phi_{LP}(\beta) = \min\ & 6y_1 + 7y_2 + 5y_3 \\ \text{s.t. } & 2y_1 - 7y_2 + y_3 = \beta \\ & y_1, y_2, y_3 \in \mathbb{R}_+ \end{aligned}$$



Note that we are minimizing here!

The LP Dual

- To understand the structure of the value function in more detail, first note that it is easy to see ϕ_{LP} is concave.
- Now consider an optimal basis matrix B for the instance (LP).
  - The gradient of ϕ_{LP} at b is $\hat{u} = c_B B^{-1}$.
  - Since $\phi_{LP}(b) = \hat{u}^\top b$ and ϕ_{LP} is concave, we know that $\phi_{LP}(\beta) \leq \hat{u}^\top \beta$ for all $\beta \in \mathbb{R}^m$.
- The traditional LP dual problem can be viewed as that of finding a linear function that bounds the value function from above and has minimum value at b.

The LP Dual (cont'd)

- As we have seen, for any $u \in \mathbb{R}^m$, the following gives a upper bound on $\phi_{LP}(b)$.

$$\begin{aligned} g(u) = \max_{x \geq 0} \left[c^\top x + u^\top (b - \bar{A}x)\right] &\geq c^\top x^* + u^\top (b - \bar{A}x^*) \\ &= c^\top x^* \\ &= \phi_{LP}(b) \end{aligned}$$

- With some simplification, we can obtain an explicit form for this function.

$$\begin{aligned} g(u) &= \max_{x \geq 0} \left[c^\top x + u^\top (b - \bar{A}x)\right] \\ &= u^\top b + \max_{x \geq 0} (c^\top - u^\top \bar{A})x \end{aligned}$$

- Note that

$$\max_{x \geq 0} (c^\top - u^\top \bar{A})x = \begin{cases} 0, & \text{if } c^\top - u^\top \bar{A} \leq \mathbf{0}^\top, \\ \infty, & \text{otherwise,} \end{cases}$$

The LP Dual (cont'd)

- So we have

$$g(u) = \begin{cases} u^\top b, & \text{if } c^\top - u^\top \bar{A} \leq \mathbf{0}^\top, \\ \infty, & \text{otherwise,} \end{cases}$$

  which is again a linear over-estimator of the value function.

- An LP dual problem is obtained by computing the strongest linear over-estimator with respect to b.

$$\begin{aligned} \min_{u \in \mathbb{R}^m} g(u) = \min \ & b^\top u \\ \text{s.t. } & u^\top \bar{A} \geq c^\top \end{aligned} \tag{LPD}$$

Combinatorial Representation of the LP Value Function

- From the fact that there is always an extremal optimum to (LPD), we conclude that the LP value function can be described combinatorially.

$$\phi_{LP}(\beta) = \min_{u \in \mathcal{E}} u^\top \beta \qquad \text{(LPVF)}$$

  for $\beta \in \mathbb{R}^m$, where

$$\mathcal{E} = \{c_B \bar{A}_E^{-1} \mid E \text{ is the index set of a dual feasible bases of } \bar{A}\}$$

- Note that $\mathcal{E}$ is also the set of extreme points of the *dual polyhedron* $\{u \in \mathbb{R}^m \mid u^\top \bar{A} \geq c^\top\}$.

The MILP Value Function

- We now generalize the notions seen so far to the MILP case.
- The *value function* associated with (MILP) is

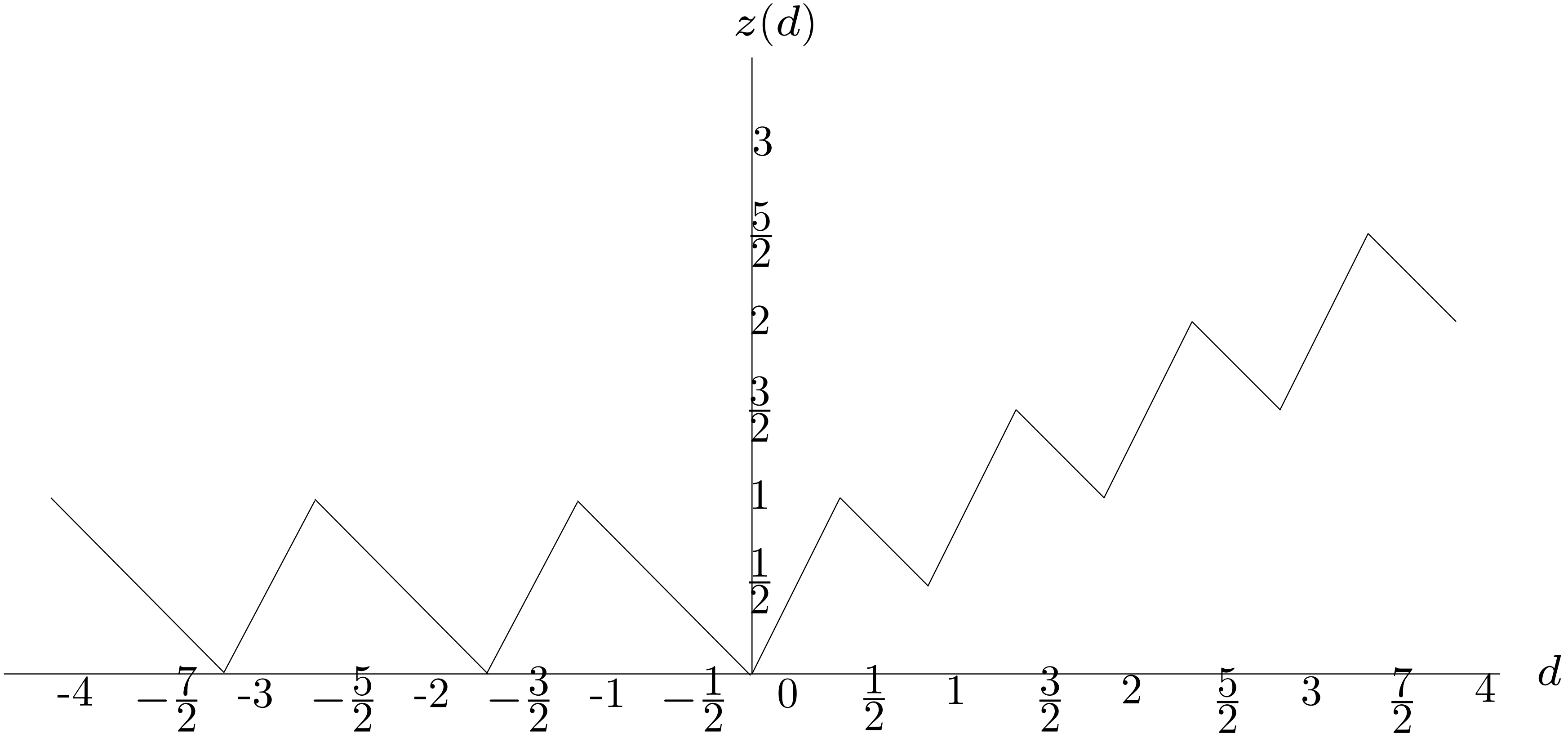
$$\phi(\beta) = \max_{x \in \mathcal{S}(\beta)} c^\top x \tag{VF}$$

for $\beta \in \mathbb{R}^m$, where $\mathcal{S}(\beta) = \{x \in \mathbb{Z}^p_+ \times \mathbb{R}^{n-p}_+ \mid \bar{A}x = \beta\}$.

- Again, we let $\phi(\beta) = -\infty$ if $\beta \in \Omega = \{\beta \in \mathbb{R}^m \mid \mathcal{S}(\beta) = \emptyset\}$.

Example: MILP Value Function

$$\begin{aligned} \phi(\beta) = \min \quad & \tfrac{1}{2}x_1 + 2x_3 + x_4 \\ \text{s.t} \quad & x_1 - \tfrac{3}{2}x_2 + x_3 - x_4 = \beta \quad \text{and} \\ & x_1, x_2 \in \mathbb{Z}_+, x_3, x_4 \in \mathbb{R}_+. \end{aligned}$$



Note again that we are minimizing here!

A General Dual Problem

- A *dual function* $F : \mathbb{R}^m \to \mathbb{R}$ is one that satisfies $F(\beta) \geq \phi(\beta)$ for all $\beta \in \mathbb{R}^m$.
- How to select such a function?
- We choose may choose one that is easy to construct/evaluate or for which $F(b) \approx \phi(b)$.
- This results in the following generalized *dual* of (MILP).

$$\min \{F(b) : F(\beta) \geq \phi(\beta),\ \beta \in \mathbb{R}^m, F \in \Upsilon^m\} \qquad \text{(D)}$$

where $\Upsilon^m \subseteq \{f \mid f : \mathbb{R}^m \to \mathcal{R}\}$.

- We call F^* *strong* for this instance if F^* is a *feasible* dual function and $F^*(b) = \phi(b)$.
- This dual instance always has a solution F^* that is strong if the value function is bounded and $\Upsilon^m \equiv \{f \mid f : \mathbb{R}^m \to \mathbb{R}\}$. Why?

LP Dual Function

- It is straightforward to obtain a dual function using linear programming.
- Simply take the dual of any LP relaxation.
- In practice, working with this dual just means using dual simplex to solve the relaxations.
- Note again that since dual simplex maintains a dual feasible solution at all times, we can stop anytime we like.
- In particular, as soon as the upper bound goes below the current lower bound, we can stop solving the LP.
- This can save significant effort.
- With an LP dual, we can "close the gap" by adding valid inequalities to strengthen the LP relaxation.
- The size of the gap in this case is a measure of how well we are able to approximate the convex hull of feasible solutions (near the optimum).

The Lagrangian Dual

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

$$L(\beta, u) = \max_{x \in \mathcal{S}_R(\beta)} (c - uA'')x + u\beta'',$$

where $\mathcal{S}_R(d) = \{x \in \mathbb{Z}^n_+ \mid A'x \leq d\}$

- Then the Lagrangian dual function, ϕ_{LD}, is

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

- We will see a number of ways of computing $\phi_{LD}(b)$ later in the course.

Dual Functions from Branch-and-Bound

As before, let $\mathcal{T}$ be set of the terminating nodes of the tree. Then, assuming we are branching on variable disjunctions, in a leaf node $t \in \mathcal{T}$, the relaxation we solve is:

$$\begin{aligned} \phi^t(\beta) = \max\ & c^\top x \\ \text{s.t. } & \bar{A}x = \beta, \\ & l^t \leq x \leq u^t, x \geq 0 \end{aligned}$$

The dual at node t:

$$\begin{aligned} \phi^t(\beta) = \min\ & \{\pi^t \beta + \underline{\pi}^t l^t + \bar{\pi}^t u^t\} \\ \text{s.t. } & \pi^t A + \underline{\pi}^t + \bar{\pi}^t \geq c^\top \\ & \underline{\pi} \geq 0, \bar{\pi} \leq 0 \end{aligned}$$

We obtain the following strong dual function:

$$\max_{t \in \mathcal{T}} \{\hat{\pi}^t \beta + \hat{\underline{\pi}}^t l^t + \hat{\bar{\pi}}^t u^t\},$$

where $(\hat{\pi}^t, \hat{\underline{\pi}}^t, \hat{\bar{\pi}}^t)$ is an optimal solution to the dual at node t.

The Duality Gap

- In most cases, the the value of an optimal solution to a given dual problem is not equal to the value of an optimal solution to (MILP).
- The difference between these values for a particular instance is known as the *duality gap* or just *the gap*.
- It is typically reported as a percentage of the value of the best known solution (this is called the *relative gap*).
- The size of the relative gap is a rough measure of the difficulty of a problem.
- It can help us estimate how long it will take to solve a given problem by branch and bound.

Strong Duality

- When the duality gap is guaranteed to be zero, we say we have a *strong dual*.
- For linear programs, the LP dual is a strong dual.
- For integer programs, the dual (D) is a strong dual, since the value function itself is a solution for which the gap is zero.
- Of course, obtaining a description of the value function is more difficult than solving theinteger program itself.