Computational Integer Programming
Universidad de los Andes

Lecture 5

Dr. Ted Ralphs
Reading for This Lecture

- Wolsey, Chapters 10 and 11
- Nemhauser and Wolsey Sections II.3.1, II.3.6, II.3.7, II.5.4
The Decomposition Principle

- Again, we consider a pure integer program $IP$ defined by

$$z_{IP} = \max \{ cx | x \in S \},$$

$$S = \{ x \in \mathbb{Z}_+^n | Ax \leq b \}.$$  

- We also assume all variables have finite upper and lower bounds.

- Recall the concept of Lagrangian relaxation: we relax some constraints and then penalize their violation.

- The principle of decomposition is to divide the inequalities describing $S$ into two sets:
  - the “easy constraints,” and
  - the “complicating constraints,” and

is such a way that removing the complicating constraints results in a integer program we can solve effectively.
The Lagrangian Relaxation

• Suppose as before that our IP is defined by

\[
\begin{align*}
\max & \quad cx \\
\text{s.t.} & \quad A^1x \leq b^1 \text{ (the “complicating” constraints)} \\
& \quad A^2x \leq b^2 \text{ (the “nice” constraints)} \\
& \quad x \in \mathbb{Z}^n
\end{align*}
\]

where optimizing over \(S_{LR} = \{x \in \mathbb{Z}^n \mid A^2x \leq b^2\}\) is “easy.”

• **Lagrangian Relaxation** (for \(u \geq 0\)):

\[
LR(u) : z_{LR}(u) = ub^1 + \max_{x \in S_{LR}} \{(c - uA^1)x\}.
\]
The Lagrangian Dual

- The next step is to obtain a dual problem formed by allowing \( u \) to vary.
- We are looking for the value of \( u \geq 0 \) that yield the lowest upper bound.
- The Lagrangian dual problem, \( LD \), is
  \[
  z_{LD} = \min_{u \geq 0} z_{LR}(u)
  \]

- The Lagrangian dual can be rewritten as the following LP
  \[
  z_{LD} = \min_{\eta, u} \{ \eta + ub^1 \mid \eta \geq (c - uA^1)x^i, i \in 1, \ldots, T, u \geq 0 \}
  \]
  where \( \{x^i\}_{i=1}^T \) are the extreme points of \( \text{conv}(S_{LR}) \).
- This can be solved using a cutting plane algorithm where the separation problem is an optimization problem over the set \( S_{LR} \).
Solving the Lagrangian Dual with Subgradient Optimization

• Note that \((c - uA^T)x\) is an affine function of \(u\) for a fixed \(x\).

• This tells us that \(z_{LR}(u)\), when viewed as a function of \(u\), is the maximum of a finite number of affine functions.

• Hence, it is piecewise linear and convex on the domain over which it is finite.

• We can easily minimize any convex function which we can evaluate and subdifferentiate using a technique called subgradient optimization.

• This technique is covered in detail in nonlinear programming.

• The procedure iteratively adjusts the weights according to the degree of violation of each constraint.
Subgradient Algorithm for the Lagrangian Dual

• The idea of the subgradient algorithm is to first fix $u$ and determine $x$ by optimizing over $S_{LR}$.

• Then update $u$ according to the observed violations.

• Here is a basic subgradient algorithm for solving the Lagrangian dual:

  1. Choose initial Lagrange multipliers $u^0 \geq 0$ and set $t = 0$.
  2. Solve the Lagrangian subproblem $LR(u^t)$.
  3. Calculate the current violation of the complicating constraints $s = b^1 - A^1x$.
  4. Set $u^{t+1}_j \leftarrow \max\{u^t_j - \mu^t \frac{s_j}{\|s\|}, 0\}$ where $\mu^t$ is the chosen step size.
  5. Set $t \leftarrow t + 1$ and go to step 2.

• This algorithm is guaranteed to converge to the optimal solution as long as $\{\mu^t\}_{t=0}^\infty \to 0$ and $\sum_{t=0}^\infty \mu^t = \infty$

• In practice, one usually uses a geometric progression for the step sizes.

• Sometimes, it’s difficult to know when the optimal solution has been reached.
Dantzig-Wolfe Decomposition

- In this technique, we utilize the fact that every point in $\text{conv}(S_{LR})$ can be written as the convex combination of extreme points of $\text{conv}(S_{LR})$.

- Here is the Dantzig-Wolfe LP:

$$
\begin{align*}
\max & \sum_{i=1}^{T} c x^i \lambda^i \\
\text{s.t.} & \sum_{i=1}^{T} A^1 x^i \lambda^i \leq b^1 \\
& \sum_{i=1}^{T} \lambda^i = 1 \\
& \lambda \in \mathbb{R}^T_+
\end{align*}
$$

where $\{x^i\}_{i=1}^{T}$ are the extreme points of $\text{conv}(S_{LR})$.

- This is a relaxation of $IP$; solving yields an upper bound.
Solving the Dantzig-Wolfe LP

- We can solve this LP using column generation.
- The column generation subproblem is again an optimization problem over $S_{LR}$.
- Note that this LP is exactly the dual of the LP we derived as being equivalent to the Lagrangian dual!
- Hence, this gives the same bound as the Lagrangian dual.
Comparing Dantzig-Wolfe to Lagrangian Relaxation

- Because they are conceptually equivalent, the distinction between Dantzig-Wolfe and Lagrangian relaxation is a bit artificial.

- Philosophically, the distinction between them is in the solution methodology typically applied and in the form of the output.

- The Lagrangian dual produces only a dual solution and does not include any explicit primal solution information.

- Dantzig–Wolfe is required to produce both a primal and a dual solution.

- The primal solution information can be used to perform separation and tighten the relaxation.
The Strength of the Decomposition Bound

• We can characterize its strength of the bound obtained by decomposition as follows:

\[ z_D = \max \{ cx \mid A^1 x \leq b^1, x \in \text{conv}(S_{LR}) \} \]

• Using this fact, we can characterize exactly when the decomposition bound is strong.

Proposition 1. \( z_{IP} = z_D \) for all objective functions if and only if

\[ \text{conv}\{S_{LR} \cap \{x \in \mathbb{R}_+^n \mid A^1 x \leq b^1\}\} = \text{conv}(S_{LR}) \cap \{x \in \mathbb{R}_+^n \mid A^1 x \leq b^1\} \]
Example

\[
\begin{align*}
\min & \quad x_1 \\
-x_1 - x_2 & \geq -8, \quad (1) \\
-0.4x_1 + x_2 & \geq 0.3, \quad (2) \\
x_1 + x_2 & \geq 4.5, \quad (3) \\
3x_1 + x_2 & \geq 9.5, \quad (4) \\
0.25x_1 - x_2 & \geq -3, \quad (5) \\
7x_1 - x_2 & \geq 13, \quad (6) \\
x_2 & \geq 1, \quad (7) \\
-x_1 + x_2 & \geq -3, \quad (8) \\
-4x_1 - x_2 & \geq -27, \quad (9) \\
-x_2 & \geq -5, \quad (10) \\
0.2x_1 - x_2 & \geq -4, \quad (11) \\
x & \in \mathbb{Z}^2. \quad (12)
\end{align*}
\]
Illustrating the Strength of the Lagrangian Dual

\[ \mathcal{P} = \text{conv}\{x \in \mathbb{Z}^2 \mid x \text{ satisfies } (1) - (11)\}, \]
\[ \mathcal{P}^1 = \{x \in \mathbb{R}^2 \mid x \text{ satisfies } (1) - (5)\}, \text{ and} \]
\[ \mathcal{P}^2 = \{x \in \mathbb{R}^2 \mid x \text{ satisfies } (6) - (11)\}, \]
\[ S_{LR} = \mathcal{P}^2 \cap \mathbb{Z}^2. \]
Comparing the Decomposition Bound to the LP bound

- The following proposition follows again from the characterization of $z_{LD}$.

**Proposition 2.** The LP relaxation of IP gives the bound $z_D$ for all objective functions if $\{x \in \mathbb{R}^n_+ \mid A^2 x \leq b^2\}$ is an integral polyhedron.

- This follows from the fact that $\text{conv}(S_{LR}) = \{x \in \mathbb{R}^n_+ \mid A^2 x \leq b^2\}$ in this case.

- Because of the equivalence of optimization and separation, we can in theory always attain this bound using a cutting plane algorithm (why?).

- However, in some cases, decomposition methods can compute this bound more efficiently.

- The advantage of the LP relaxation is that it can be further strengthened using cutting planes valid for $S$.

- It is also possible to strengthen the Lagrangian dual in this way.
Choosing a Decomposition

- Often, there are multiple choices for the decomposition.
- The definition of the set $S_{LR}$ determines the strength of the bound.
- However, it is important to choose a relaxation that can be solved relatively easily (but not too easily).
- The relaxation must be solved iteratively in order to obtain the bound.
- Recall the TSP example.
Comparing Decomposition-based Bounding to LP-based Bounding

- The class of methods we have just discussed are called decomposition-based methods because they decompose the problem into two parts.

- Up until the mid-1970’s, these methods were very popular for solving integer programming problems.

- They can effectively strengthen the bound obtained by LP relaxation alone.

- However, after methods based on strengthening the LP relaxation using polyhedral cutting planes were introduced, these methods fell out of favor.

- It is possible to combine these two approaches.

- This is one of the current frontiers of research in integer programming.