Stochastic Programming
Lecture 11

Dr. Brian Denton

NCSU

October 12, 2010

Today's Class

1. Summary of Today's Class
2. Inner Linearization
3. Extreme Point Methods
4. Special Cases

Outline

Summary of Today's Class

So far we have discussed the L-shaped method and several extensions including

- Multi-cut L-shaped method
- Full decomposability
- Bunching

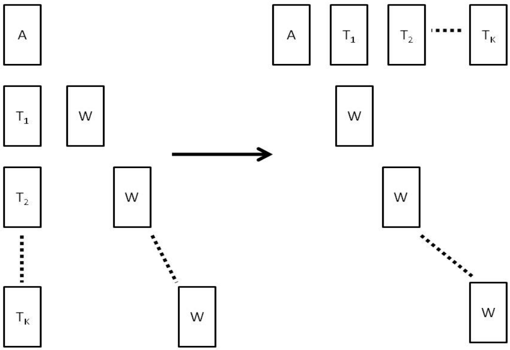
Today we will discuss:

- Inner Linearization
- Extreme point methods
- Special cases (simple recourse, network flow)

Outline

Inner Linearization

An alternative to the L-shaped method (outer linerization) is inner linearization.



Inner Linearization

We can relate D-W Decomposition to the L-shaped method by examining the dual of the L-shaped master problem:

$$\begin{aligned}
&\min z = cx + \theta \\
&\text{s.t. } \; Ax = b \\
&D_\ell x \geq d_\ell, \;\; \ell = 1, ..., r, \\
&E_\ell x + \theta \geq e_\ell, \;\; \ell = 1, ..., s, \\
&x \geq 0, \theta \in \Re.
\end{aligned}$$

Dual of L-shaped master:

$$\begin{aligned}
&\max w = \rho b + \sum_{\ell=1}^{r} \bar{\sigma}_\ell d_\ell + \sum_{\ell=1}^{s} \bar{\pi}_\ell e_\ell \\
&\text{s.t. } \; \rho A + \textstyle\sum_{\ell=1}^{r} \bar{\sigma}_\ell D_\ell + \sum_{\ell=1}^{s} \bar{\pi}_\ell E_\ell \leq c \\
&\textstyle\sum_{\ell=1}^{s} \bar{\pi}_\ell = 1, \bar{\sigma}_\ell \geq 0, \ell = 1, ..., r, \bar{\pi}_\ell \geq 0, \ell = 1, ..., s.
\end{aligned}$$

Inner Linearization

The dual of the L-shaped master includes columns that are:

- Convex combinations of expectations of subproblem extreme points
- Directions of recession (*extreme rays*)

Consider the dual to the subproblem at iteration ν:

$$\max\{\pi(h_k - T_k x^\nu) \mid s.t.\ \pi W \leq q\} \tag{1}$$

If (1) is unbounded for any k then add a column (d_{r+1}, D_{r+1}) to the dual master problem.

If (1) has finite optimal value for all k then add a column (e_{s+1}, E_{s+1}) to the dual master problem.

Inner Linearization

Algorithm:

Step 0: $r = s = \nu = 0$

Step 1: Set $\nu = \nu + 1$ and solve master. Let solution be $(\rho^\nu, \bar{\sigma}^\nu, \bar{\pi}^\nu)$ with a dual solution , (x^ν, θ^ν).

Step 2: For $k = 1, ..., K$ solve (1). If an unbounded solution with extreme ray σ^ν is found for any k, then form new column $(d_{r+1} = \sigma^\nu h_k, D_{r+1} = \sigma^\nu T_k)$, set $r = r + 1$, and return to Step 1. If (1) is solvable for all k then add a column (e_{s+1}),(E_{s+1}) to the master.

Step 3: If $e_{s+1} - E_{s+1} x^\nu - \theta^\nu \leq 0$, then stop; Otherwise return to Step 1.

Outline

Extreme Point Methods

Important Points:

- Extreme point methods are powerful approaches for solving large-scale LPs.
- Basis factorization is a computationally intensive part of extreme point methods
- The 2SLP has a constraint matrix with special structure for basis factorization (see Birge and Qi, 1988).

The basis at an iteration of an extreme point method can be written as:

$$W = \begin{bmatrix} A_{I_0} & & & \\ T_{I_0} & W_{J_1} & & \\ \vdots & & \ddots & \\ T_{I_0} & & & W_{J_K} \end{bmatrix}$$

Extreme Point Methods

Basic idea is to take advantage of the block diagonal structure of part of the matrix to decompose basis factorization.

Proposition 5 (Birge and Louveaux, p. 180): A basis matrix, B, is equivalent after permutation P to

$$B' = PB = \left[\begin{array}{cc} D & C \\ F & L \end{array} \right]$$

where D is square invertible and at most $n_1 \times n_1$ and L is an invertible matrix of K invertible blocks of sizes at most $m_2 \times m_2$ each.

Extreme Point Methods

Using proposition 5 the linear system can be written as:

$$Dx_B + Cy_B = b', \quad Fx_B + Ly_B = h'$$

Using this form we can write the following efficient way to evaluate x_B, y_B:

$$y_B = L^{-1}(h' - Fx_B)$$

and substituting back into the first system of equations yields

$$(D - CL^{-1}F)x_B = b' - CL^{-1}h'$$

Most of the effort in computing x_B and y_B involves basis $(D - CL^{-1}F)$ and using L^{-1} which is easy to evaluate because it is block separable.

Outline

Special Cases

Sometimes the recourse LP has a special structure that can be leverage to achieve computational advantage. Two common special cases are

- Simple recourse
- Network flow

Simple recourse problems are *Fully Decomposable* owing to their separable second stage.

Computational advantages can be achieved when network flow structure exists either

- As part of the first stage decision
- Or in the second stage recourse problem