Stochastic Programming
Lecture 12

Dr. Brian Denton

NCSU

October 7, 2010

Today's Class

1. More Methods for Solving 2SLPs
2. Regularized Decomposition
3. C/C++ - CPLEX Example

Outline

1. More Methods for Solving 2SLPs
2. Regularized Decomposition
3. C/C++ - CPLEX Example

More Methods for Solving 2SLPs

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

- Multi-cut L-shaped method
- Full decomposability
- Bunching
- Inner Linearization
- Extreme point methods
- Special cases of recourse (simple recourse, network flow)

Today we will discuss:

- Regularized decomposition
- A CPLEX Example using C/C++ and the callable library

Outline

Regularized Decomposition

A common problem with the L-shaped method is that early iterations do not produce useful optimality cuts.

Regularized Decomposition provides a means for dampening variation in the first stage solution using a quadratic penalty term (Ruszczynski, 1986).

Regularized Decomposition

Algorithm:

Step 0: Set $r = \nu = 0$, $s_k = 0$ for all $k = 1, ..., K$. Select a^1, a feasible solution.

Step 1: Set $\nu = \nu + 1$. Solve the regularized master:

$$
\begin{aligned}
&\min z = cx + \sum_{k=1}^{K} \theta_k + \alpha(x - a^\nu)^2 \\
&\text{s.t. } Ax = b \\
&D_\ell \geq d_\ell, \;\; \ell = 1, ..., r, \\
&E_{\ell(k)}x + \theta_k \geq e_{\ell(k)}, \;\; \ell(k) = 1, ..., s_k, \\
&x \geq 0, \theta \in \mathfrak{R}.
\end{aligned}
$$

Let (x^ν, θ^ν) be an optimal solution. If $cx^\nu + e\theta^\nu = ca^\nu + Q(a^\nu)$ stop with a^ν optimal.

Regularized Decomposition

Step 2: Solve Phase 1 LPs for $k = 1, ..., K$. If a feasibility cut is generated set $a^{\nu+1} = a^{\nu}$ (called a *null infeasible step*) and return to Step 1.

Step 3: Solve subproblems for $k = 1, ..., K$. If stopping criteria is not satisfied ($\theta_k < p_k(\pi_k)(h_k - T_k x^{\nu})$ for some k) then generate an optimality cut. Set $s_k = s_k + 1$. Otherwise continue.

Step 4: If $\theta_k \leq p_k(\pi_k)(h_k - T_k x^{\nu})$ for all k then set $a^{\nu+1} = x^{\nu}$ (called an *exact serious step*). Go to Step 1.

Step 5: If $cx^{\nu} + Q(x^{\nu}) \leq ca^{\nu} + Q(a^{\nu})$ then $a^{\nu+1} = x^{\nu}$ (called an approximate serious step). Go to Step 1. Else set $a^{\nu+1} = a^{\nu}$ (null feasible step) and go to Step 1.

Regularized Decomposition

Convergence of regularized decomposition was established by Ruszczynski (1986)

Lemma 1 (B&L Ch 6, Section 6.1).

$e^T\theta^\nu \leq \eta(x^\nu, \theta^\nu, a^\nu) \leq Q(a^\nu)$.

Proof Sketch:

The first inequality follows from $||x^\mu - a^\nu||^2 \geq 0$.

The second inequality follows from the fact that a^ν is always feasible. The solution $(a^\nu, \hat{\theta})$ where $\hat{\theta}_k = p_k Q_k(a^\nu), k = 1, ..., K$ satisfies all optimality cuts since θ_k is a lower bound on $p_k Q_k(\cdot)$. It follows that $\eta(x^\nu, \theta^\nu, a^\nu) \leq \eta(x^\nu, \hat{\theta}^\nu, a^\nu) = Q(a^\nu)$.

Regularized Decomposition

Lemma 2 (B&L Ch 6, Section 6.1).

If the regularized decomposition algorithm stops at Step 1, then a^ν solves the original problem.

Proof Sketch:

By Lemma 1 and the optimality criterion, $e\theta^\nu = Q(a^\nu)$, and it follows $e\theta^\nu = \eta(x^\nu, \theta^\nu, a^\nu)$, which implies $||x^\nu - a^\nu||^2 = 0$ and therefore $x^\nu = a^\nu$. Thus, a^ν solves the regularized master and the unregularized master which means a^ν solves the original problem.

Regularized Decomposition

Lemma 3 (B&L Ch 6, Section 6.1).

If there is a null step at iteration ν, then

$$\eta(x^{\nu+1}, \theta^{\nu+1}, a^{\nu+1}) > \eta(x^{\nu}, \theta^{\nu}, a^{\nu})$$

Proof Sketch: Because the regularized master problem is strictly convex it has a unique optimal solution. A null step at iteration ν may be either a null feasible step or a null infeasible step. In the first case a cut is generated that renders x^{ν} infeasible. In the second case a cut renders x^{ν}, θ^{ν} infeasible. Thus the objective function necessarily increases.

Regularized Decomposition

Lemmas 4-5 establishes finite convergence as the number of serious and approximate serious steps are finite. Lemmas 1 - 5 are used to prove the main result...

Theorem 8 (B&L Ch 6, Section 6.1).

If the original problem has a solution, then the algorithm stops after a finite number of iterations. Otherwise, it generates a sequence of feasible points a^ν *such that* $Q(a^\nu)$ *tends to* $-\infty$ *as* $\nu \to \infty$.

Outline

C/C++ - CPLEX Example

- ILOG CPLEX comes with a callable C/C++ library that includes simplex, network simplex, dual simplex, interior point, quadratice, and MIP solvers
- Using the callable library requires some understanding of C/C++ and the CPLEX functions used to build, solve, modify, and extract information from LPs