IE 495 – Lecture 8

SMPS and the Recourse Function

Prof. Jeff Linderoth

February 5, 2003

Outline

- Formulating Stochastic Program(s).
  - ◇ SMPS format
  - ◇ Jacob & MIT
- Two stage problems with recourse – Expected value function

Please don't call on me!

- What are the KKT conditions?
- Who is Karmarkar?
- The simplex method is a polynomial time algorithm for linear programming.
- True or False: MPS format is a concise, new format for expressing linear and integer programs?
- Explain two ways to model "your favorite eight letter word"?

Multistage Formulation—Implicit Nonanticipativity

maximize

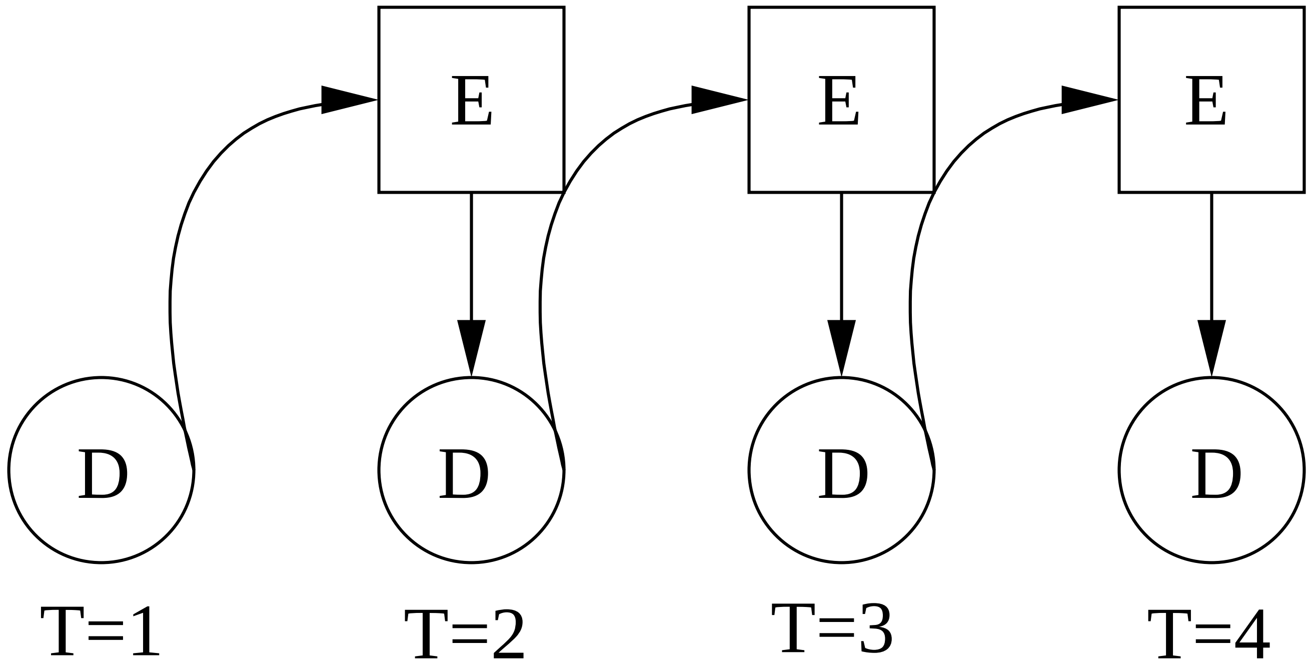
$$\sum_{s \in S} p_s(qy_s - rw_s)$$

subject to

$$\begin{aligned}
\sum_{i \in N} x_{i1} &= b \\
\sum_{i \in N} \omega_{il} x_{i,A(l)} &= \sum_{i \in N} x_{il} \qquad \forall l \in \mathcal{L} \setminus 1 \\
\sum_{i \in N} \omega_{iA(s)} x_{iA(s)} - y_s + w_s &= G \qquad \forall s \in S
\end{aligned}$$

SMPS

- Multistage problems are based on the "event-decision" model



- All random "stuff" must be in the stage associated with the decision.

ω in "Wrong" Stage

- To get SMPS to "work". Let's rewrite the problem so that the ω_{it} are associated with variables in stage t
- Let x_{it} : Amount of money invested in i in time t
- Let y_{it} : Amount of money you have in i at time t
- $(1/\omega_{it})y_{it} = x_{i,t-1}$

An Equivalent (Longer) Formulation

maximize

$$qE + rU$$

subject to

$$\begin{aligned}
\sum_{i \in N} x_{i1} &= b \\
\omega_{it}^{-1} y_{it} - x_{i,t-1} &= 0 \qquad \forall i \in N, \forall t \in \mathcal{T} \setminus 1 \\
\sum_{i \in N} x_{it} - \sum_{i \in N} y_{it} &= 0 \qquad \forall t \in \mathcal{T} \setminus 1 \\
\sum_{i \in N} y_{iT} - E + B &= G \\
x_{it} &\geq 0 \qquad \forall i \in N, t \in \mathcal{T} \\
y_{it} &\geq 0 \qquad \forall i \in N, t \in \mathcal{T} \setminus 1 \\
E, U &\geq 0
\end{aligned}$$

The Matrix

- N = {S, B}
- T=4

x_{S1}	x_{B1}	y_{S2}	y_{B2}	x_{S2}	x_{B2}	y_{S3}	y_{B3}	x_{S3}	x_{B3}	y_{S4}	y_{B4}	E	U	
1	1													b
-1		$1/\omega_S$												0
	-1		$1/\omega_B$											0
		-1	-1	1	1									0
				-1		$1/\omega_S$								0
					-1		$1/\omega_B$							0
						-1	-1	1	1					0
								-1		$1/\omega_S$				0
									-1		$1/\omega_B$			0
										1	1	-1	1	G

SMPS Format

- How do we specify a stochastic programming instance to the solver?
- We could form the deterministic equivalent ourselves, but you saw how unnatural that seems.
  - ◇ For *really* big problems, forming the deterministic equivalent is out of the questions.
  - ◇ We need to just specify the random parts of the model.
- We can do this using SMPS format
  - ◇ Actually, other than (explicity) forming the deterministic equivalent in a modeling language, this is the ONLY way.
  - ⋆ There *is* some recent research work in developed stochastic programming support in an AML.

SMPS Components

- Core file
  - Like MPS file for “base” instance
- Time file
  - Specifies the time dependence structure
- Stoch file
  - Specifies the randomness

SMPS

- The SMPS format is *broad*[a]
  - There are very few (if any) full implementations, that simply read the full format
  - No solver will solve all instances that can be expressed in the format.
- The SMPS format is (seemingly) being changed.
- A good site...
  - `http://www.mgmt.dal.ca/sba/profs/hgassmann/SMPS2.htm`

[a] *Too* broad, IMO

Papers

- I handed out to you...
  - ◇ J.R. Birge, M.A.H. Dempster, H.I. Gassmann, E.A. Gunn, A.J. King and S.W. Wallace, "A standard input format for multiperiod stochastic linear programs", COAL Newsletter #17 (1987) pp. 1-19.
- There are more. For example...
  - ◇ H.I. Gassmann and E. Schweitzer, "A comprehensive input format for stochastic linear programs", Annals of Operations Research 104 (2001) 89-125.
- Any questions?

SMPS Core File

- Like an MPS file specifying a “base” scenario

⋆ *Must* permute the rows and columns so that the time indexing is sequential

```
NAME          jake
ROWS
 N  obj
 E  c1
 E  c2
 E  c3
 E  c4
 E  c5
 E  c6
 E  c7
 E  c8
 E  c9
 E  c10
COLUMNS
    xs1       c1                  1   c2                  -1
    xb1       c1                  1   c3                  -1
    ys2       c2                999   c4                  -1
    yb2       c3                888   c4                  -1
    xs2       c4                  1   c5                  -1
    xb2       c4                  1   c6                  -1
    ys3       c5                999   c7                  -1
    yb3       c6                888   c7                  -1
    xs3       c7                  1   c8                  -1
    xb3       c7                  1   c9                  -1
    ys4       c8                999   c10                  1
    yb4       c9                888   c10                  1
    t         obj             -0.05   c10                 -1
    s         obj               1.1   c10                  1
RHS
    rhs       c1              10000   c10              15000
ENDATA
```

jake.time

- ◇ Specify which row/column starts each time period.
- ⋆ Must be sequential!

```
*23456789 123456789 123456789
TIME          jake
PERIODS       IMPLICIT
    xs1       c1                      T1
    ys2       c2                      T2
    ys3       c5                      T3
    ys4       c8                      T4
ENDATA
```

Stoch File

- BLOCKS
  - Specify a "block" of parameters that changes together
- INDEP
  - Specify that all the parameters you are specifying are all independent random variables
- SCENARIO
  - Specify a "base" scenario
  - Specify what things change and when...

jake.stoch

```
*23456789 123456789
STOCH         jake
*23456789 123456789 123456789 123456789
BLOCKS        DISCRETE
 BL BLOCK1    T2          0.5
    ys2       c2          0.8
    yb2       c3          0.8772
 BL BLOCK1    T2          0.5
    ys2       c2          0.9434
    yb2       c3          0.8929
 BL BLOCK2    T3          0.5
    ys3       c5          0.8
    yb3       c6          0.8772
 BL BLOCK2    T3          0.5
    ys3       c5          0.9434
    yb3       c6          0.8929
 BL BLOCK3    T4          0.5
    ys4       c8          0.8
    yb4       c9          0.8772
 BL BLOCK3    T4          0.5
    ys4       c8          0.9434
    yb4       c9          0.8929
ENDATA
```

An INDEP Example

```
*23456789 123456789
STOCH         jake
*23456789 123456789 123456789 123456789
INDEP         DISCRETE
    ys2       c2             0.8                 0.333
    ys2       c2             1.0                 0.333
    ys2       c2             1.05                0.333
*
    yb2       c3             0.9                 0.125
    yb2       c3             0.99                0.675
    yb2       c3             1.0                 0.125
ENDATA
```

NEOS

- If we have files in SMPS format, how do we solve the resulting instance?
- We can use NEOS!

`http://www.mcs.anl.gov/neos`

- There are currently two solvers
  - ◇ CPA—Works for two-stage LP
  - ◇ MSLIP—Works for multistage LP
- I am working on getting others that we can use.

⋆ You will (likely) be asked to solve some an instance on your next homework assignment

Math Time! Two-Stage SLP w/Fixed Recourse

minimize

$$c^T x + \mathbb{E}_\omega \left[q^T y \right]$$

subject to

$$\begin{aligned} Ax &= b \\ T(\omega)x + Wy(\omega) &= h(\omega) \qquad \forall \omega \in \Omega \\ x &\in X \\ y(\omega) &\in Y \end{aligned}$$

- $Q(x, \omega) = \min_{y \in Y} \{ q^T y : Wy = h(\omega) - T(\omega)x \}$

Some Notation Review

$$\min_{x \in X : Ax = b} \left\{ c^T x + \mathbb{E}_\omega \left[\min_{y \in Y} \{ q^T y : Wy = h(\omega) - T(\omega)x \} \right] \right\}$$

- *Second stage value function*, or *recourse (penalty) function* $v : \Re^m \mapsto \Re$.
- $v(z) \equiv \min_{y \in Y} \{ q^T y : Wy = z \}$,
  - ◇ Given “policy” x and realization of randomness ω
  - ◇ If z measures the first-stage deviation $z = h(\omega) - T(\omega)x$, $v(z)$ is the minimum cost way to “correct” so that the constraints hold again.
- $Q(x, \omega) = v(h(\omega) - T(\omega)x)$

More Notation

- *Expected Value Function*, or *Expected minimium recourse function* $\mathcal{Q} : \Re^n \mapsto \Re$.
  - ◇ $\mathcal{Q}(x) \equiv \mathbb{E}_\omega[Q(x, \omega)]$
  - ◇ For any policy $x \in \Re^n$, it describes the expected cost of the recourse.

The SP Problem

- Using these definitions, we can write our recourse problem in terms only of the x variables:

$$\min_{x \in X} \{c^T x + \mathcal{Q}(x) : Ax = b\}$$

- This is a (nonlinear) programming problem in $\Re^n$.

⇒ The ease of solving such a problem depends on the properties of $\mathcal{Q}(x)$.

? Does anyone know what $\mathcal{Q}(x)$ is?

  ◇ Linear,(?) Convex,(?) Continuous,(?) Differentiable(?)

- This is what we are going to study for a while...

Here We Go

- For the time being, let $Y = \Re^p_+$.

$$v(z) = \min_{y \in \Re^p_+} \{q^T y : Wy = z\}, z \in \Re^m$$

- Thus, for a fixed z, we solve a linear program to evaluate $v(z)$.
- Assume that LP duality holds $\forall z \in \Re^m$
  - ◇ Only for the time being!
  - ◇ We'll reconsider later...
  - ◇ $-\infty < v(z) < \infty$

⋆ This is some notation. Let

- $\{y \in \Re^p_+ : Wy = z\} = \emptyset \Rightarrow v(z) = \infty$
- $\exists d \in \Re^n_+ : Wd = 0, q^T d < 0 \Rightarrow v(z) = -\infty$.

Proofs

- So if LP duality holds...

$$v(z) = \min_{y \in \Re^p_+} \{q^T y : Wy = z\} = \max_{t \in \Re^m} \{z^T t : W^T t \leq q\}$$

- Let $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_{|\Lambda|}\}$ be the set of extreme points of $\{t \in \Re^m | W^T t \leq q\}$.
  - ◇ Each of those extreme points λ_k is potentially an optimal solution to the LP.
  - ◇ In fact, we are sure that there is no optimal solution better than one that occurs at an extreme point, so we can write...

$$v(z) = \max_{k=1,\ldots,|\Lambda|} \{z^T \lambda_k\}, z \in \Re^m.$$

What's All This?

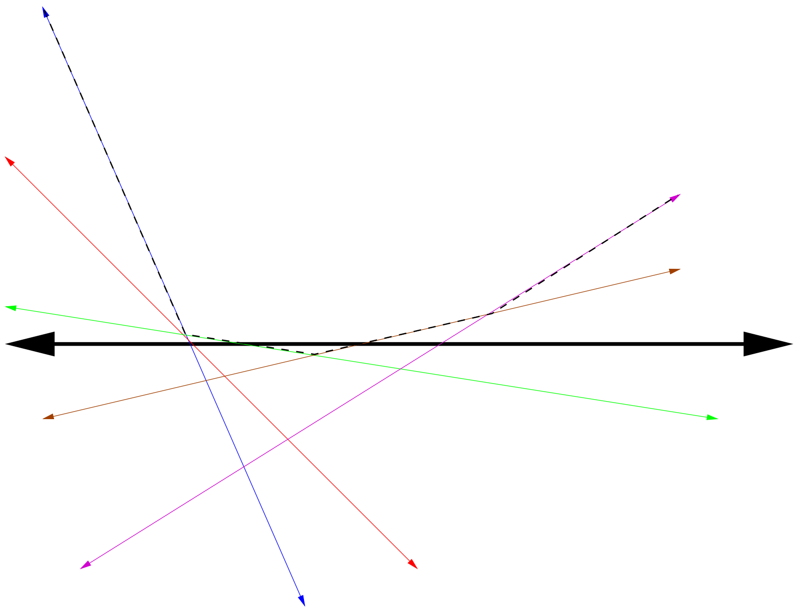
$$\begin{aligned}
\alpha v(z_1) + (1-\alpha) v(z_2) &= \max_{k=1,2,\ldots|\Lambda|} z_1^T \lambda_k + (1-\alpha) \max_{k=1,2,\ldots|\Lambda|} z_2^T \lambda_k \\
&\leq \alpha z_1^T \lambda_k^* + (1-\alpha) z_2^T \lambda_k^* \\
&= (\alpha z_1 + (1-\alpha) z_2)^T \lambda_k^* \\
&\leq \max_{k=1,2,\ldots|\Lambda|} [(\alpha z_1 + (1-\alpha) z_2)^T \lambda_k] \\
&= v((\alpha z_1 + (1-\alpha) z_2))
\end{aligned}$$

Quite Enough Done

? What did I just prove?

Convex!

- $v(z)$ is convex of $z \in \Re^m$.
- In fact...
  - ◇ **Thm:** If $f_1(x), f_2(x), \ldots f_q(x)$ is an arbitrary collection of convex functions, then $M(x) = \max\{f_1(x), f_2(x), \ldots f_q(x)\}$ is also a convex function.



I Don't Care About v

? What about $Q(x, \omega)$?

- Recall $Q(x, \omega) \equiv v(h(\omega) - T(\omega)x)$

$$\lambda Q(x_1, \omega) + (1 - \lambda)Q(x_2, \omega)$$

$$\begin{aligned}
&= \lambda v(h(\omega) - T(\omega)x_1) + (1 - \lambda)v(h(\omega) - T(\omega)x_2) \\
&\geq v(\lambda(h(\omega) - T(\omega)x_1) + (1 - \lambda)(h(\omega) - T(\omega)x_2)) \\
&= v(h(\omega) - T(\omega)(\lambda x_1 + (1 - \lambda)x_2)) \\
&= Q(\lambda x_1 + (1 - \lambda)x_2, \omega)
\end{aligned}$$

Quite Enough Done

Continuing On

- So $Q(x, \omega)$ is convex in x for a fixed ω.
- In fact...
  - **Thm:** If A is a linear transformation from $\Re^n \mapsto \Re^n$, and $f(x)$ is a convex function on $\Re^m$, the composite function $(fA)(x) \equiv f(Ax)$ is a convex function on $\Re^n$.

Almost Done...

- What about $\mathcal{Q}(x) \equiv \mathbb{E}_\omega Q(x, \omega)$
  - ◇ From the remainder of today, let's assume that ω comes from a probability space with finite support.
  - ◇ This means that there are finite number of discrete values $\{\omega_1, \omega_2, \ldots, \omega_m\}$ that ω can take.

$$\mathcal{Q}(x) = \sum_{i=1}^{m} P(\omega = \omega_i) Q(x, \omega_i)$$

Finishing up

- **Thm:** If $f(x)$ is convex, and $\alpha \geq 0$, $g(x) \equiv \alpha f(x)$ is convex.
- **Thm:** If $f_k(x), k = 1, 2, \ldots K$ are convex functions, so is $g(x) \equiv \sum_{k=1}^{K} f_k(x)$.
- Put it all together and you get...
  - ⋆ $\mathcal{Q}(x)$ is a convex function of x.

What assumptions have we made so far?

- LP duality holds $\forall z = (h(\omega) - T(\omega)x) \in \Re^m$
- ω has finite support

Next time

- Discuss these assumptions
- Show more properties of $\mathcal{Q}(x)$
  - Optimality conditions
- Homework #2. `:-(`
  - It will be less time consuming that HW#1