Lecture Notes on Solving Moral-Hazard Problems Using the Dantzig-Wolfe Algorithm

Edward Simpson Prescott

Prepared for ICE '05, July 2005

Outline

1. Why compute?
    - Answer quantitative questions
    - Analyze difficult problems
2. Quick introduction to linear programs
3. Develop Dantzig-Wolfe decomposition algorithm
4. Introduce moral hazard problem
5. Extensions to other private information programs

Linear Programs (standard form)

$$\max_{\mathbf{x} \geq \mathbf{0}} \ \mathbf{ex}$$

$$\text{s.t. } \mathbf{Ax} = \mathbf{b}$$

$\mathbf{x} \sim (\mathbf{n} \times \mathbf{1})$
$\mathbf{e} \sim (\mathbf{1} \times \mathbf{n})$ (lp textbooks usually use $\mathbf{c}$, for economists c is consumption)
$\mathbf{A} \sim (\mathbf{m} \times \mathbf{n})$ (the constraint matrix)
$\mathbf{b} \sim (\mathbf{m} \times \mathbf{1})$

Basic Feasible Solutions

Definitions:

A *basic solution* is an $\mathbf{x} \in \Re^{\mathbf{n}}_{+}$ such that $\mathbf{Ax} = \mathbf{b}$ and there exists column indices $B(1), ..., B(m)$ such that the columns $\mathbf{B} = \mathbf{A_{B(1)}}, ..., \mathbf{A_{B(m)}}$ are linearly independent and $x_i = 0$ for $i \neq B(1), ..., B(m)$.

A *basic feasible solution* is a basic solution such that $\mathbf{x} \geq 0$.

For a basic solution, a *basic variable* is any one of these m variables. A *non-basic variable* is any one of the $n - m$ variables.

$\mathbf{B}$ is called the *basis matrix* .

Important Properties

1. There is a finite number of basic feasible solutions.
2. If a solution exists to the linear program then there exists an optimum that is also a basic feasible solution.

Simplex-Based Algorithms

Simplex algorithms

1. Searches over set of basic feasible solutions.
2. Traditionally the main class of algorithms used to solve lp's. (Developed by Dantzig in the 40's.)

Simplex prices (or simplex multipliers)
Exist for each basis

$$\mu = \mathbf{e}'_{\mathbf{B}}\mathbf{B}^{-\mathbf{1}}$$

The marginal benefit of introducing non-basic variable j into the basis is

$$e_j - \mu\mathbf{A_j} \tag{1}$$

Called reduced costs in lp textbooks because most textbook problems minimize. For us it increases utility.

Notice: At an optimum, μ is the vector of Lagrangian multipliers. For variables in the basis the expression equals zero. For non-basic feasible variables (with value 0) the expression is less than or equal to zero.

Simplex-Based Algorithms (cont.)

1. Start with a basic feasible solution (ways to find, e.g. solve an auxiliary lp).
2. Calculate the increased utility (1) of introducing each non-basic variable into the basis (thus removing one of the basic variables).
3. If (1) ≤ 0 for all non-basic variables then bfs is optimal. Stop.
4. If (1) > 0 for any non-basic variable one of those can be made basic, increasing the value of the objective function.
5. The algorithm takes the new basis, calculates new simplex multipliers, and continues until it finds an optimum.

The algorithm also has steps to take care of degeneracy and unbounded solutions. For our purposes, we can ignore them.

Simplex-Based Algorithms

Important theoretical property

- Finds solution in a finite number of steps.
- Intuition: At each step solution improves. There are only a finite number of basic feasible solution so eventually an optimum is found.
- **GLOBAL METHOD**

In practice

- Good convergence properties.
- Often, number of constraints effects number of pivots (basis changes) needed to find a solution.
- Memory is an issue. Larger the constraint matrix the larger the needed memory.

Dantzig-Wolfe Decomposition Algorithm

A Simple Block-Angular Linear Program

$$\begin{aligned} \max_{x \geq 0} \ & \mathbf{ex} \\ \text{s.t. } & \mathbf{D_1 x_1} + \mathbf{D_2 x_2} \leq \mathbf{b} \\ & \mathbf{A_1 x_1} \leq 0 \\ & \mathbf{A_2 x_2} \leq 0 \end{aligned}$$

First set of constraints are called *connecting* constraints. The $\mathbf{A_i x_i}$ are the *blocks*.

Note:

1. Nothing special about having two blocks. Just for simplicity.
2. Setting right-hand side variables for blocks to zero because moral-hazard problem has this property.
   (a) Can still use Dantzig-Wolfe even if non-zero but problem takes a slightly different form.

Representation Theorem

Representation Theorem
Any convex polyhedron with at least one extreme point can be represented as a convex combination of its extreme points and a non-negative combination of its extreme rays.

Our blocks are $P_i = \{\mathbf{x_i} \geq \mathbf{0} | \mathbf{A_1 x_1} \leq \mathbf{0}\}$.

$\mathbf{0}$ is an extreme point.

Because of homogeneity only need extreme rays. Let $\mathbf{w_i^k}$ be the extreme rays of P_i. Then by the representation theorem

$$\mathbf{x_i} = \sum_{\mathbf{k}} \theta_{\mathbf{i}}^{\mathbf{k}} \mathbf{w_i^k}$$

for some $\{\theta_i^k\} \geq 0$.

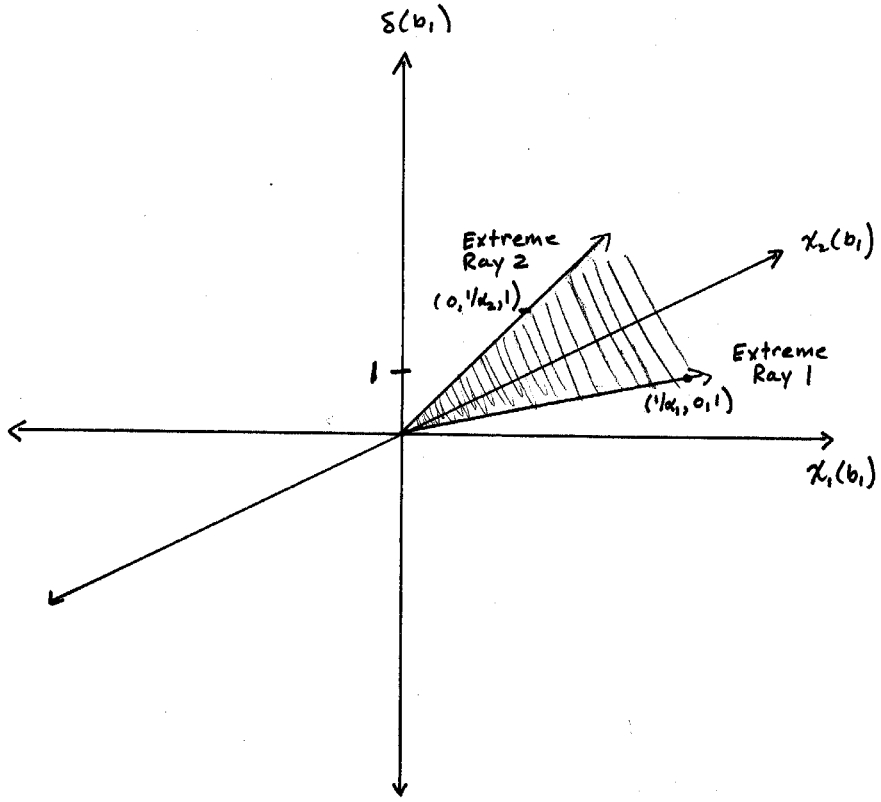


Figure 2: Description of the set $\{(x_1(b_1), x_2(b_1), \delta(b_1)) \in \Re^3_+ | x_1(b_1) + x_2(b_1) - \delta(b_1) = 0\}$, which is illustrated by the shaded area. The two extreme rays are $(1/\alpha_1, 0, 1)$ and $(0, 1/\alpha_2, 1)$. Both are scaled to $\delta(b_1) = 1$. As is apparent in the figure, the above set can also be represented by the set of all non-negative linear combinations of its extreme rays.

Master Program

Master Program (rays only - moral hazard program has this feature)

$$\max \sum_i \sum_{k \in K_i} \theta_i^k (\mathbf{e_i w_i^k})$$

$$\text{s.t. } \sum_i \sum_{k \in K_i} \theta_i^k (\mathbf{D_i^k w_i^k}) \leq \mathbf{b}$$

$$\theta_i^k \geq 0, \quad \forall i, k$$

Notice:
Each *scalar* $(\mathbf{e_i w_i^k})$ is a coefficient of the objective function. Each *column vector* $(\mathbf{D_i w_i^k})$ are coefficients in the constraint matrix.

Computation Issue

- Master Program is a linear program. Less constraints but LOTS more variables.
- For block i, the number of candidate rays (not all are feasible) is $\binom{n_i}{m_i}$. Big number!
- Lots of non-basic variables to check

Master Program and Checking Optimality

Need to check for each non-basic k

$$(\mathbf{e_i w_i^k}) - \mu(\mathbf{D_i w_i^k}) \tag{2}$$

Lots of k. How do we check them all? Here is the beauty of the Dantzig-Wolfe algorithm. For each block i solve:

Subprogram

$$\begin{aligned} \max_{x_i \geq 0} (\mathbf{e_i} - \mu \mathbf{D_i}) \mathbf{x_i} \\ \text{s.t. } \mathbf{A_i x_i} \leq \mathbf{0} \\ \mathbf{1 x}_i = 1 \end{aligned}$$

The additional constraint keeps the solution bounded and scales the rays. (Textbook treatment does not add this constraint.) This can be done in the moral-hazard problem because it is bounded. I do it because it gives the moral-hazard subprograms a nice geometric interpretation as well as tying the problem closer to the moral-hazard literature.

Basic feasible solutions to subprogram i's constraint set define the set of scaled extreme rays for block i in the master program. Therefore, if the solution to the subprogram satisfies (2) for each i then we have checked all the extreme rays and found the optimum.

Dantzig-Wolfe Decomposition Algorithm

Dantzig-Wolfe algorithm

1. Solve Master Program with $k \in \hat{K}_i$. (Called restricted master program.) (Assume it has a feasible solution.)
2. Generate the simplex multipliers μ
3. For each block i, solve subprogram i with the multipliers in the objective function.
4. Check the optimality condition, that is, $(\mathbf{e_i w_i^k}) - \mu(\mathbf{D_i^k w_i^k})) \leq \mathbf{0}$, $\forall i, k$.
5. If satisfied, solution to restricted master program is optimal. Stop.
6. If not satisfied, add rays generated from subprograms to $\hat{K}_i$.
7. Repeat from step 1.

Computational advantage

Memory is the main one. At any one time only need to store the constraint matrices of the subprogram and the restricted master program in memory.

Moral-Hazard Problem

Principal hires an agent to operate a technology.

Environment

Consumption - $c \in C$
Action/Effort - $a \in A$
Output - $q \in Q$, cardinality of Q finite

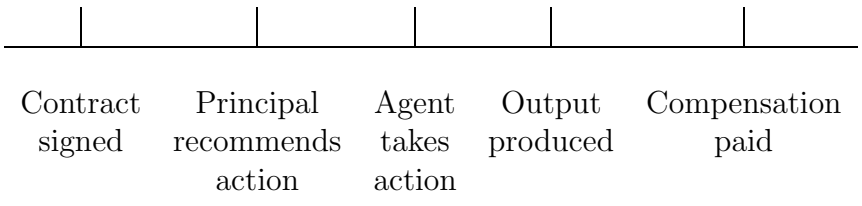
Preferences:

Agent $U(c, a)$
Principal $W(q - c)$
Agent has outside opportunity worth $\bar{U}$ utils.

Technology:

$p(q|a) > 0, \forall a, q$
Agent's action is *private information*

Timeline



Deterministic Contracts

A *deterministic contract* is a recommended action a and an output-dependent compensation schedule $c(q)$.

Program: Deterministic contract

$$\max_{a,c(q)} \sum_q p(q|a)W(q - c(q))$$

$$\text{s.t.} \quad \sum_q p(q|a)U(c(q), a) \geq \bar{U}, \quad \textbf{(PC)}$$

$$a \text{ solves } \max_{\tilde{a}} \sum_q p(q|\tilde{a})U(c(q), \tilde{a}), \quad \textbf{(IC)}.$$

Incentive constraints

Because of private information.

Without IC (and if agent is risk averse) would get full insurance, $c(q) = \bar{c}$.

Lots of contracts are output dependent (e.g performance pay, sharecropping), which is why moral hazard model is so heavily used in economics.

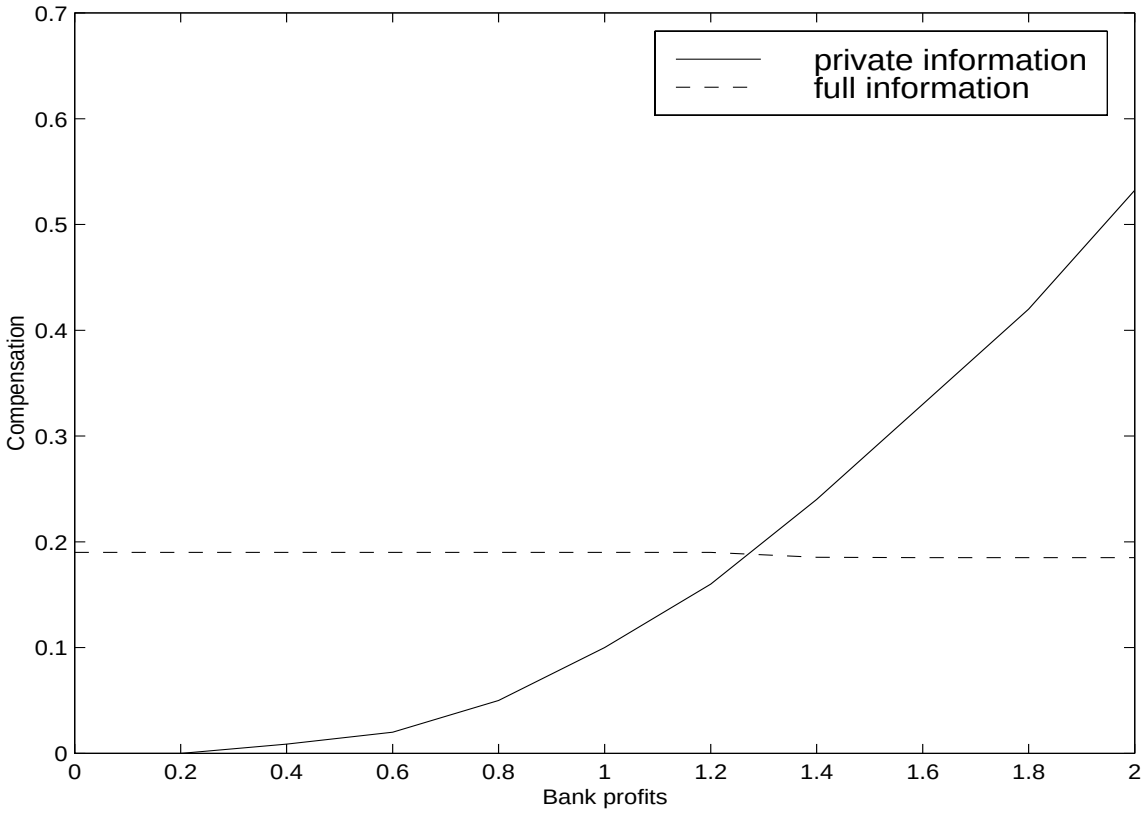


Figure 1: Informational assumptions and consumption sharing rules

Alternative specification of IC

$$\sum_q p(q|a)U(c(q),a) \geq \sum_q p(q|\hat{a})U(c(q),\hat{a}), \quad \forall \hat{a} \qquad \textbf{(AIC)}$$

Lots of incentive constraints. Desirable to replace with first-order condition. For separable preferences $U(c) - V(a)$:

$$\sum_q p_a(q|a)U(c(q)) = V'(a)$$

Problem with replacement:

- Agent's maximization problem (the IC) is concave only under restrictive conditions on $p(q|a)$.
  - (e.g. monotone likelihood ratio property, a convexity condition on the cumulative density function)
- In general, first-order condition to subproblem is not sufficient.

Computational methods need to handle incentive constraints globally (check them all)

Moral Hazard with Lotteries

Different contracts: Environment as before but now, make C and A finite too.

Contractual terms:
Recommended action
$\pi(a)$ - A probability distribution over actions
Compensation schedule
$\pi(c|q,a)$ - Conditional probability distribution, depends on q and recommended a.

A *contract with lotteries* is a $\pi(a)$ and $\pi(c|q,a)$.

This is a direct mechanism. *Revelation Principle* implies that it is sufficient to consider this class of contracts.

In general, contract with lotteries is *not* an approximation to the deterministic contract. Randomization can improve upon deterministic contracts in two ways:

1. Action lotteries can remove non-convexities in implementable utilities
2. Consumption lotteries can weaken incentive constraints

Lotteries are also valuable for decentralization. (Prescott and Townsend (1984))

Formulating the Linear Program

An identity:

$$\pi(c, q, a) = \pi(c|q, a)p(q|a)\pi(a) \tag{3}$$

Strategy: make $\pi(c, q, a)$ the choice variable and add constraints ((TC) below) to guarantee that (3) holds.

Program with lottery contracts

$$\max_{\pi \geq 0} \sum_{c,q,a} \pi(c, q, a)W(q - c)$$

$$\text{s.t.} \quad \sum_{c,q,a} \pi(c, q, a)U(c, a) \geq \bar{U}, \quad \textbf{(PC)}$$

$$\forall \bar{q}, \bar{a}, \quad \sum_{c} \pi(c, \bar{q}, \bar{a}) = p(\bar{q}|\bar{a}) \sum_{c,q} \pi(c, q, \bar{a}), \quad \textbf{(TC)}$$

$$\sum_{c,q} \pi(c, q, a)u(c, a) \geq \sum_{c,q} \pi(c, q, a)\frac{p(q|\hat{a})}{p(q|a)}u(c, \hat{a}), \forall a, \hat{a}, \textbf{(IC)}$$

$$\sum_{c,q,a} \pi(c, q, a) = 1.$$

This is a linear program.

Direct computation

Storage requirements: Number of non-zero entries because good lp code uses sparse matrices.

$$n_a(n_a - 1 + n_q + 2)n_c n_q$$

If $n_c = 100$, $n_q = 50$, $n_a = 400$ get 902 million non-zero entries (This is bad.)

Back to Dantzig-Wolfe

The constraint matrix has the following structure:

$$\begin{bmatrix} \mathbf{u_1} & \mathbf{u_2} & \mathbf{u_3} & . & . & . & \mathbf{u_n} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & . & . & . & \mathbf{1} \\ \mathbf{A_1} & & & & & & \\ & \mathbf{A_2} & & & & & \\ & & \mathbf{A_3} & & & & \\ & & & . & & & \\ & & & & . & & \\ & & & & & . & \\ & & & & & & \mathbf{A_n} \end{bmatrix}$$

Each $\mathbf{A}_i$ is a block that corresponds to variables $\pi(c, q, a_i)$ and the (IC) and (TC) for a_i.

One of these blocks (with the scaling) is

$$\begin{aligned} \forall \hat{a} \in A, \ \sum_{c,q} \pi(c, q, a_i)(U(c, a_i) - \frac{p(q|\hat{a})}{p(q|a)} U(c, \hat{a})) &\geq 0, \\ \forall \bar{q}, \ \sum_{c} \pi(c, \bar{q}, a_i)(1 - p(\bar{q}|a_i)) - \sum_{c, q \neq \bar{q}} \pi(c, q, a_i) p(\bar{q}|a_i) &= 0, \\ \forall c, q, \ \pi(c, q, a_i) \geq 0, \ \text{ and } \ \sum_{c,q} \pi(c, q, a_i) &= 1. \end{aligned}$$

Take the extreme points of each block and calculate principal's and agent's utility from them.

$$x_i = (x_{w_i}, x_{u_i})$$

Solving with Dantzig-Wolfe

Master Program

$$\max_{\tilde{\pi}\geq 0} \sum_{x_i,i} \tilde{\pi}(x_i) x_{w_i}$$

$$\text{s.t.} \quad \sum_{x_i,i} \tilde{\pi}(x_i) x_{u_i} \geq \bar{U}, \tag{4}$$

$$\sum_{x_i,i} \tilde{\pi}(x_i) = 1, \tag{5}$$

$$\forall i, \quad x_i \in X_i. \tag{6}$$

The $\tilde{\pi}(x_i)$ correspond to the $\theta's$. They are probabilities.

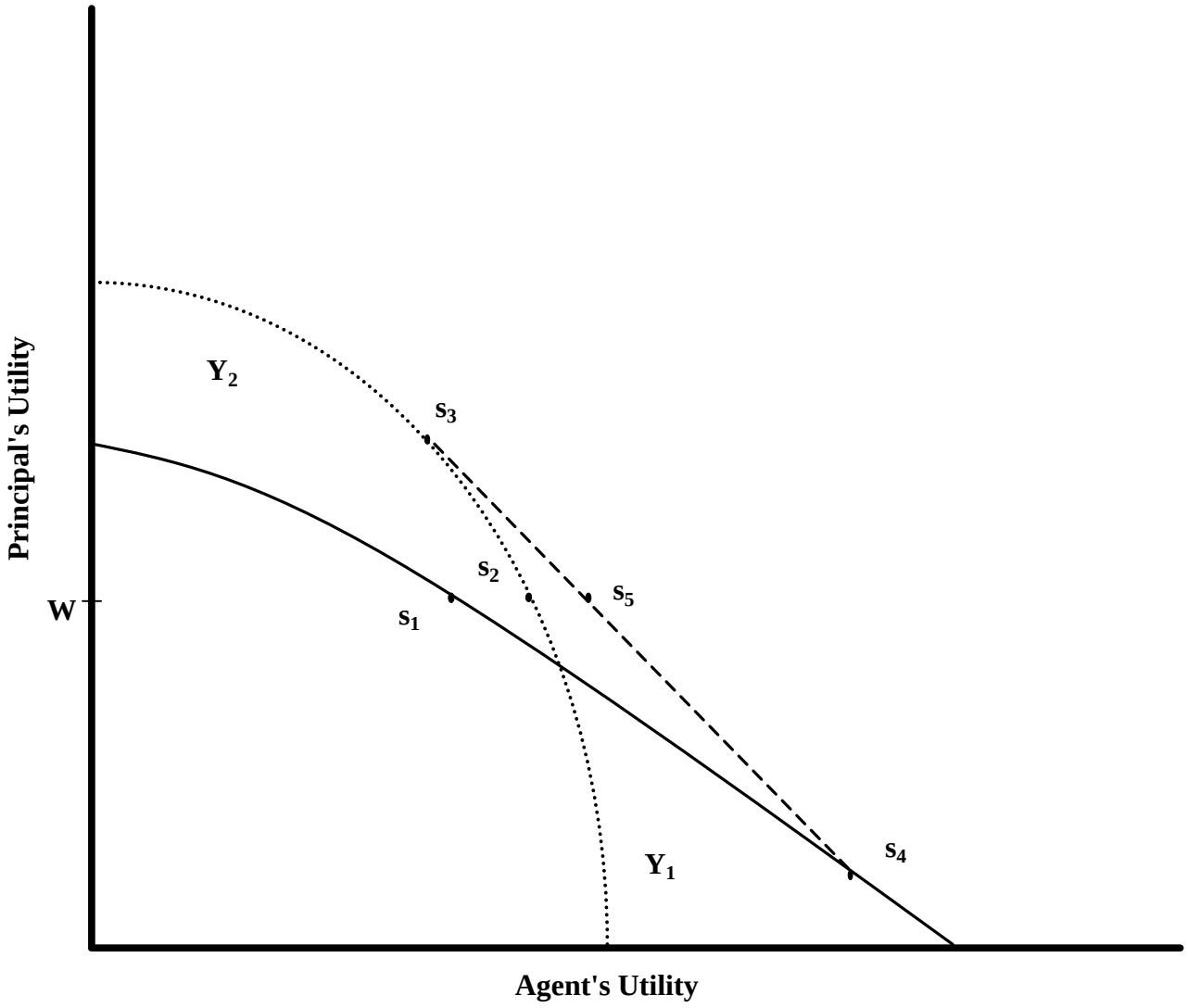
Optimality condition: $(\mu_u \leq 0)$

$$\forall i, \forall x_i, \quad x_{w_i} - \mu_u x_{u_i} \leq \mu_p, \tag{7}$$

Subprogram i

$$\max_{\pi(c,q,a_i)} \sum_{c,q} \pi(c,q,a_i)(W(q-c) - \mu_u U(c,a_i))$$

$$\text{s.t. the block } i \text{ constraints (scaled)},$$


Principal's Utility
Agent's Utility
W
Y2
Y1
s1
s2
s3
s4
s5

Computing

D-W Memory requirements: Effectively only need to worry about subprograms.

Number of non-zero entries per subprogram is

$$(n_a - 1 + n_q + 1)n_c n_q$$

Simplex Memory Requirements: Number of non-zero entries is

$$n_a(n_a - 1 + n_q + 2)n_c n_q$$

Biggest savings is in terms of number of actions. If have a lot, then this algorithm helps.

Example 1 from Prescott (2004)

Set $n_c = 200$, $n_q = 50$, and varied n_a.

Number of actions	Standard	Dantzig-Wolfe
5	0.76	1.01
10	4.63	2.14
15	9.08	3.34
20	14.95	4.70
25	22.23	5.20
30	*	8.47
35	*	9.36
50	*	13.27
75	*	27.99
100	*	51.32
150	*	101.71
200	*	169.83
300	*	358.03
400	*	612.47
500	*	828.14

Table 1: Time to solve Example 1 in cpu minutes for various number of action grid points and for both algorithms.

Speed good for D-W here. That was specific to example. Had lots of infeasible subproblems. Also, algorithm converged relatively quickly.

Decomposition methods can be parallelized. I did not exploit that feature.

Two-Dimensional Action Choice

Preferences

$$U(c, a_m, a_s) = c^{0.5} + 0.25(2 - a_m) + a_s$$

Agent likes risk. Bank regulation/corporate finance applications.

$$W(c, q) = \begin{cases} q - c - 20 & q \leq 0.21 \\ q - c & q > 0.21 \end{cases}$$

Principal is loss averse. Bank regulators, bureaucracies.

Grids: $n_{A_m} = 10$, $n_{A_s} = 10$, $n_q = 50$, $n_c = 100$

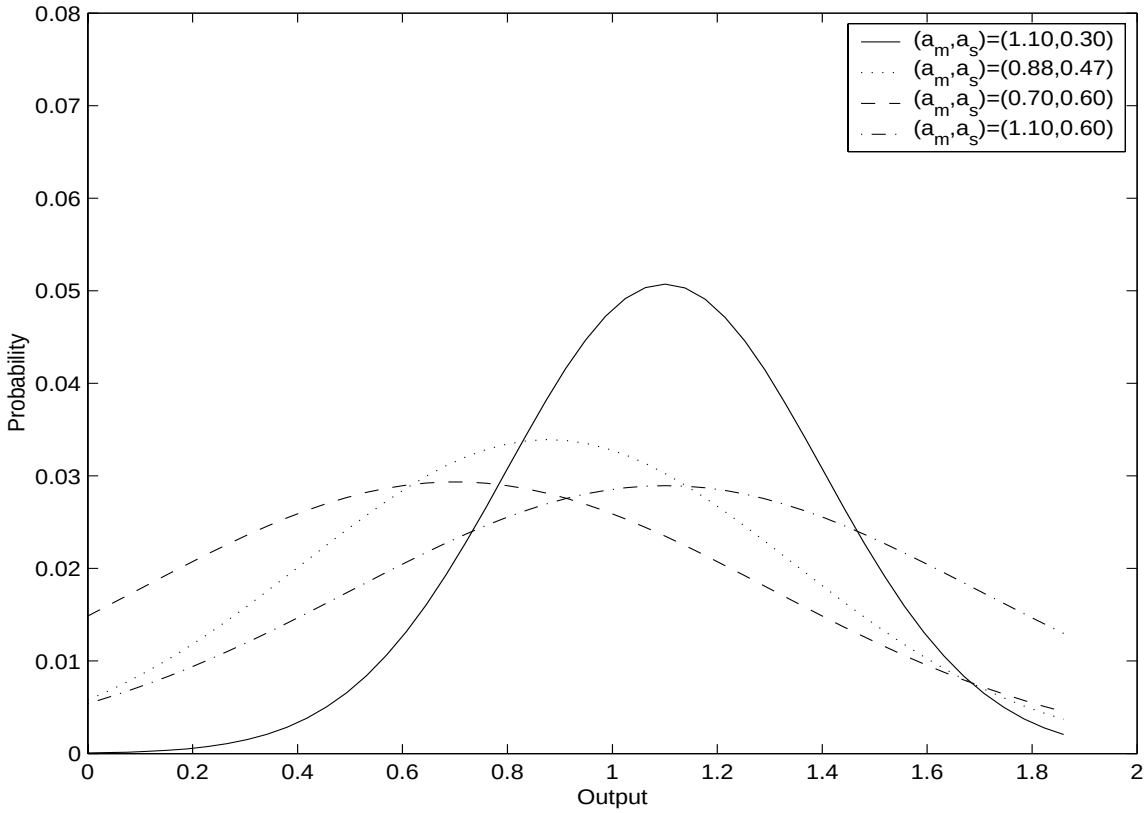


Figure 2: Probability Distribution for Selected Actions

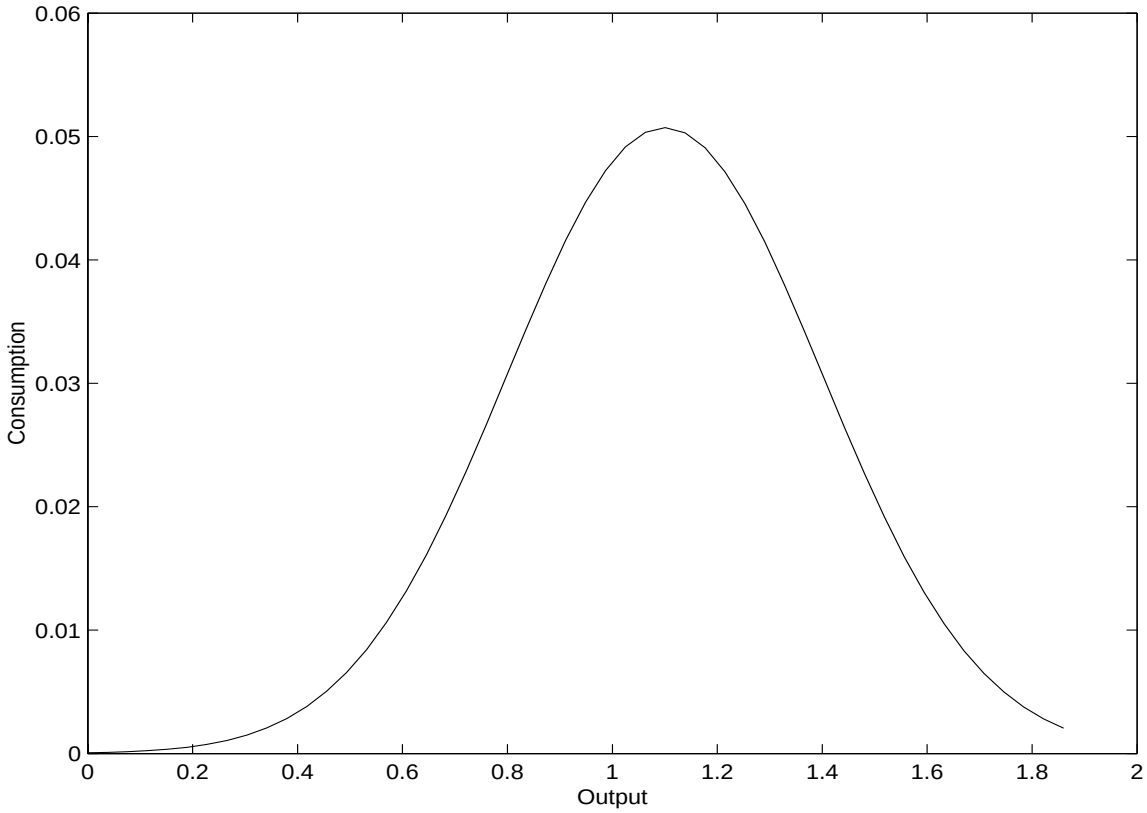


Figure 3: Optimal Consumption Sharing Rule

Other Problems Amenable to Techniques

Simple extensions to moral hazard problem

- Publicly observed input k so $p(q|a,k)$ is technology
  - Choose $\pi(c,q,a,k)$
  - One block per (a,k) pair
- Multiple agents, each working separate projects, no correlation, risk-neutral principal who can borrow
  - Choose $\pi_j(c,q,a)$
  - one block per (a,j) pair
  - more connecting constraints

A not so simple extension

- Hidden information followed by moral hazard
- Constraint matrix has special structure
  - Can use max operator to reduce number of off-equilibrium path incentive constraints
  - Prescott (2004)
  - Should also be able to use decomposition methods

More on Private Information Problems

Representation as linear programs

- *Revelation Principle*
- Private information environments with lotteries and finite numbers of agents, choices, and shocks can be formulated as linear programs.
  - In multi-agent problems, related to *correlated equilibria*.

Correlated Equilibria

Two players, move simultaneously (from Myerson (1991))

	Agent 2	
Agent 1	y_1	y_2
x_1	(5,1)	(0,0)
x_2	(4,4)	(1,5)

2 pure strategy Nash equilibria - (x_1, y_1), (x_2, y_2)

With equal weights in Planner's problem either gives social welfare of 3. Dominated by $(4, 4)$ but that allocation is not a Nash equilibria.

Correlated Equilibria (cont)

Correlated equilibria - planner sends a *private* message to each agent recommending an action. Recommendation may be chosen randomly.

Joint distribution of recommendations is $\pi(x, y)$.

Agent 1's incentive constraints

$$\begin{aligned}(5-4)\pi(x_1, y_1) + (0-1)\pi(x_1, y_2) &\geq 0, \\ (4-5)\pi(x_2, y_1) + (1-0)\pi(x_2, y_2) &\geq 0. \quad \textbf{(IC1)}\end{aligned}$$

Agent 2's incentive constraints

$$\begin{aligned}(1-0)\pi(x_1, y_1) + (4-5)\pi(x_2, y_1) &\geq 0, \\ (0-1)\pi(x_1, y_2) + (5-4)\pi(x_2, y_2) &\geq 0. \quad \textbf{(IC2)}\end{aligned}$$

Correlated Equilibria (cont)

Equal Pareto Weight Planner's Program

$$\max_{\pi(x,y)\geq 0} 6\pi(x_1,y_1) + 0\pi(x_1,y_2) + 8\pi(x_2,y_1) + 6\pi(x_2,y_2)$$

s.t. **(IC1)**, **(IC2)**, and

$$\sum_{x,y} \pi(x,y) = 1.$$

This is a linear program. Decomposition *not* useful here.

Solution

$$\pi(x_1,y_1) = \pi(x_2,y_1) = \pi(x_2,y_2) = 1/3.$$

Social welfare is $10/3 > 3$.

Summary

Main Points:

- Private information problems can be formulated as LP's.
- Many have sparse matrix structures that can be exploited with decomposition or other methods.

Background Reading for Lecture on Computational Methods for Solving Private Information Models

Edward Simpson Prescott
Federal Reserve Bank of Richmond

Most of the lecture will be about the moral hazard model, how to formulate it as a linear program, and how to solve the linear program using the Dantzig-Wolfe decomposition algorithm. Much of the material is drawn from the following two papers:

Prescott, Edward S. ``A Primer on Moral Hazard Models.'' *Federal Reserve Bank of Richmond Economic Quarterly* vol. 85 (1999), pp. 47-77.
www.richmondfed.org/publications/economic_research/economic_quarterly/pdfs/winter1999/prescott.pdf

Prescott, Edward Simpson. ``Computing Solutions to Moral-Hazard Programs Using the Dantzig-Wolfe Decomposition Algorithm." *Journal of Economic Dynamics & Control* vol. 28 (2004), pp. 777-800.

Some background knowledge of linear programming is needed. There are many good textbooks on linear programming. Below is one that I like:

Bertsimas, D., Tsitsiklis, J. *Introduction to Linear Optimization*. Athena Scientific, Belmont, Massachusetts. (1997). (Has a good description of the Dantzig-Wolfe decomposition algorithm.)

Many mechanism design or contract theory models can be formulated as linear programs. Related, equilibria of some game theoretic concepts can be formulated as systems of linear inequalities. Below are several sources with models along this line:

Myerson, Roger B. *Game Theory: Analysis of Conflict*. Harvard University Press, Cambridge, Mass. (1991), Sections 6.1-6.3. (Discussions of correlated equilibria and games with communication.)

Phelan, Christopher and Robert M. Townsend. ``Computing Multi-Period, Information-Constrained Optima." *Review of Economic Studies* vol. 58 (1991), pp. 853-81. (Infinite horizon, repeated moral hazard.)

Prescott, Edward Simpson. ``Communication in Private Information Models: Theory and Computation." *Geneva Papers on Risk and Insurance Theory* vol. 28 (2003), pp. 105-130. (Hidden information followed by moral hazard. Has a method for dealing with off-equilibrium strategies that is useful for computation.)

Townsend, Robert M. *The Medieval Village Economy: A Study of the Pareto Mapping in General Equilibrium Models*. Princeton University Press, Princeton, NJ. (1993). (Covers a variety of models.)

Townsend, Robert M. ``Arrow-Debreu Programs as Microfoundations of Macroeconomics." In Truman F. Bewley, ed., *Advances in Economic Theory: Fifth World Congress*. Cambridge, Cambridge University Press, 1987. (Develops a variety of models.)

Also, see the somewhat dated references in Appendix C of the Primer.