Econ 172A - Slides from Lecture 1

Joel Sobel

September 27, 2012

WELCOME

1. Outlines Distributed.
2. Copying Information from Slides is Probably a Waste of Time

INTRODUCTIONS

- Joel, 311 Econ, TuTh 2:30–3:20
- TAs
  1. Leland Farmer, 128 Econ, M 2:30–3:30
  2. Erin Giffin, 117 Econ, Tu 5:00–6:00

Class email: econ172af12@gmail.com

COURSE MATERIAL

1. Book: Expensive, available, old editions OK. Buying it unnecessary.
2. Lecture "notes": Class web page, Soft Reserves.
3. Old problems, exams, solutions: Class web page
4. Excel's Solver
5. Sections Tuesdays in York Hall 226 from 8-9 PM and from 9-10 PM.

http://www.econ.ucsd.edu/%7Ejsobel/172f12/172f12home.htm

Prerequisites

1. Whatever is in the catalog. (Intermediate Micro, Probability, Linear Algebra)
2. In fact: Linear Algebra, ability to follow a logical argument, spreadsheet experience.

WAIT LIST

Not my job.
Go to departmental office, Sequoyah Hall 245.

GRADING

1. 15% quizzes (4 given, best 3 count)
2. 35% midterm (November 1)
3. 50% final (December 10)

Ground rules: No books, no notes, no electronic aids.

Curve? Not really.

Straight scale? Not really.

ADMINISTRATIVE

1. Wait List? 245 Sequoyah.
2. Don't have access to Excel? Installed on campus machines.
3. Google docs spreadsheet also has a solver.
4. Don't have access to campus machines? See me.
5. Can't make the quizzes, midterm, final at scheduled time? One quiz: Drop it. Otherwise: Drop class.
6. If you are not sure what cheating is, talk to me.

HOW TO STUDY

1. Work problems.
2. Learn terminology and results from lecture.
3. Ask yourself questions.

HOW NOT TO STUDY

1. "Read" lecture notes and book.
2. Review answers to problems prior to working problems.

WHAT (I hope) YOU WILL LEARN

1. Translate verbal statement to math and back. (Econ/Management)
2. Basic Structure of Linear Programming. (Math)
3. What is an Algorithm. (Computer Science)
4. What solutions look like and how they change when problem changes. (Math/Econ)

TOPICS

Problem Formulation

Graph

Duality

Complementary Slackness

Sensitivity Analysis

Integer Programming

Branch and Bound

Network Algorithms

Transportation and Assignment Problems

WARNING

1. Second time slide presentation for UG.
2. It did not work well last time.
3. This time I plan to post slides but use a lot of chalk in lectures.

OPERATIONS RESEARCH

1. Originally: research designed to "optimize" military operations.
2. Currently: Use of mathematical models to study problems that arise in managerial/industrial decision making.
3. Aspects:
   3.1 Pure Mathematics: When do problems have solutions? Characterization? Existence/properties of algorithms.
   3.2 Computer Science: Design of algorithms. Complexity of Problems.
   3.3 Management Science: Domain of applications. Interpretation of Solutions.

MATHEMATICAL PROGRAMMING PROBLEM

Problem of the form:

$$\max f(x) \text{ subject to } x \in S.$$

1. f is called the **objective function**.
2. S is called the **feasible set** or **constraint set**.

MEANING

$$\max f(x) \text{ subject to } x \in S.$$

- **solution**: Special choice of x, x^*, satisfying
  1. $x^* \in S$. ("x^* is feasible.")
  2. If $x \in S$ then $f(x^*) \geq f(x)$. ("x^* is optimal.")
- **value**: $f(x^*)$.

PROPERTIES OF SOLUTIONS

1. Solutions: May not exist for two reasons.
   1.1 S is empty ("problem is **not feasible**").
   1.2 It is possible to make the value $f(x)$ arbitrarily large ("problem is **unbounded**").
2. Solution may be unique.
3. There may be more than one solution.

PROPERTIES OF VALUES

Values must be unique (if they exist).

MINIMIZATION PROBLEM

Same theory as max.
If you can solve:

$$\max f(x) \text{ subject to } x \in S,$$

then you can solve

$$\min g(x) \text{ subject to } x \in S$$

Solve:

$$\max -f(x) \text{ subject to } x \in S.$$

This describes the solution(s) to the min problem.
Value is multiplied by -1.

SPECIAL KINDS OF MATH PROG PROBLEMS

1. Linear Programming (f linear function, S defined by linear inequalities)
2. Non-linear programming (f arbitrary function, S defined by non-linear inequalities)
3. Integer Programming (Some components of x constrained to be integers.)

Linear Programming – this class. “Discrete” methods.

Non-linear Programming – 172B. “Calculus” methods.

Integer Programming – this class. Specialized algorithms.

LINEAR FUNCTION

$$f(x) = c \cdot x = \sum_{j=1}^{n} c_j x_j = c_1 x_1 + \cdots + c_n x_n$$

Here $x = (x_1, \ldots, x_n)$ are variables. $c = (c_1, \ldots, c_n)$ are constants.

LINEAR CONSTRAINT

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i$$

or

$$Ax \leq b$$

where A is **technology matrix**, with n columns and m rows. $b = (b_1, \ldots, b_m)$.

DEFINING PROPERTIES OF LINEAR FUNCTIONS

1. Additivity: $f(x + y) = f(x) + f(y)$
2. Constant Returns to Scale: $f(ax) = af(x)$ for any number a.

Economic Interpretation

- Additivity: You can combine independent production processes.
- Returns to Scale: Doubling Process Costs Twice as Much.

GENERAL FORM OF LINEAR PROGRAM

$\max c \cdot x$ subject to $Ax \leq b$, $x \geq 0$.

- $x = (x_1, \ldots, x_n)$ are the n variables.
- $c = (c_1, \ldots, c_n)$ are the n coefficients of the objective function.
- A is a matrix with n columns and m rows.
- The entry in row i and column j is a_{ij}

It turns out that any linear programming problem can be written in this way.

JARGON

Together A, b, c are the data of the problem (given constants).
Jargon:

- c: coefficients of the objective function
- b: Right hand side constants
- A: technology
- $x \geq 0$: nonnegativity constraint.

WHEN IS LINEARITY SENSIBLE?

1. Typically a good assumption about pricing.
   (if p_j is price per unit of good j; x_j is number of units of good j purchased, then
   $$p \cdot x$$
   is how much it costs to buy x.)
   But: maybe volume discounts; maybe you can't buy half an apple.
2. Linearity is not a good model of utility (when there is decreasing marginal utility).
3. Linearity is not a good assumption when there are decreasing or increasing returns to scale.
4. Linearity is not a good assumption when there are indivisibilities (units come in whole numbers).