3E4: Modelling Choice

Lecture 7

Introduction to
nonlinear programming

1

Announcements

Solutions to Lecture 4-6 Homework

- will be available from http://www.eng.cam.ac.uk/~dr241/3E4

Looking ahead to Lecture 8

- please email me any particularly hard questions for review part of lecture
- I'll hand out a copy of a sample exam paper

2

3E4 : Lecture Outline

Lecture 1. Management Science & Optimisation Modelling: Linear Programming

Lecture 2. LP: Spreadsheets and the Simplex Method

Lecture 3. LP: Sensitivity & shadow prices
Reduced cost & shadow price formulae

Lecture 4. Integer LP: branch & bound

Lecture 5. Network flows problems

Lecture 6. Multiobjective LP

Lecture 7 – 8. Introduction to nonlinear programming

3

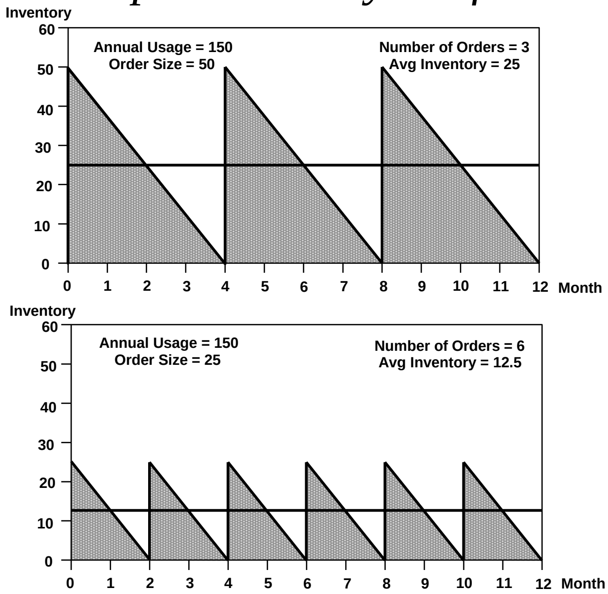
A simple Nonlinear Program

Economic Order Quantity (EOQ) Problem for Managing Inventory

- Involves determining the optimal quantity to purchase when orders are placed.
- Deals with trade off between carrying or holding cost, and ordering cost
- Small orders result in:
  - low inventory levels & carrying costs
  - frequent orders & higher ordering costs
- Large orders result in:
  - higher inventory levels & carrying costs
  - infrequent orders & lower ordering costs

4

Sample Inventory Profiles



5

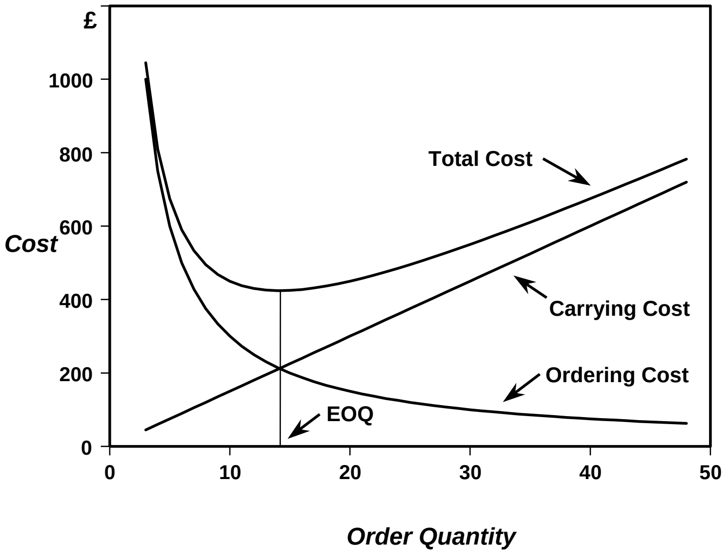
The standard EOQ model

- Data
  - **D** = annual demand for the item
  - **C** = unit purchase cost for the item
  - **S** = fixed cost of placing an order
  - ***i*** = cost of holding inventory for a year

    (expressed as a % of C)
- Variable *Q:* order quantity
- Minimize Annual Cost: MIN DC+(D/*Q*)S+(*Q/2*)*i*C

6

EOQ Cost Relationships



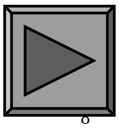
7

Variations of the EOQ Model

- A-level calculus exercise shows that the optimal order quantity Q is given by

$$Q^* = \sqrt{\frac{2DS}{Ci}}$$

- More realistic variations on the basic EOQ model need to account for
  - quantity discounts
  - storage restrictions
  - etc



8

Linear functions

- A linear (or rather affine) function is of the form

$$f(x_1,\ldots,x_n) = a_1x_1 + \ldots + a_nx_n + c$$

where $a_1,...,a_n$ and c are data and $x_1,\ldots x_n$ are variables

- In spreadsheets you are **not** leaving the realm of linear formulas if
  - you multiply linear formulas by data or constants or add them up

9

Nonlinear functions

- You may be leaving the realm of linear formulas if you
  - multiply two formulas or divide one by another
  - Use the MAX, MIN, or ABS functions
  - use the IF or ROUND functions
- If you do any of the above then your formula represents a nonlinear function

10

Levels of Nonlinearity

- Differentiable functions
  - If $f(x_1,\ldots,x_n)$ and $g_i(y_1,\ldots,y_m)$ for i=1, …, n are differentiable then $f(g_1(y),\ldots g_n(y))$ is a differentiable function of y
  - You are not leaving the realm if differentiable functions if you apply LN,EXP,^,SIN etc. to differentiable formulas
  - Typically, *nonlinear* means *differentiable* (so LP is a subclass of NLP)
- Non-differentiable functions
  - You may leave the realm of differentiable functions if you use MAX, MIN, ABS
- Discontinuous functions
  - You may leave the realm of continuous functions if you use IF, ROUND, CEILING, FLOOR, INT, LOOKUP

11

Nonlinear Programming (NLP)

- NLP problems have a nonlinear objective function and/or one or more nonlinear constraints
- NLP problems are formulated in virtually the same way as linear problems.
- The mathematics involved in solving general NLPs is quite different to LPs.
- Solver tends to hide this difference but it is important to understand the difficulties that may be encountered when solving NLPs.

12

`Solver` *on Simple NLPs*

- Min cos(πx) starting from
  - $x = 0.1$
  - $x = 0$
  - $x = -0.1$
- Min $(x-1)^2$
  - Unconstrained, or
  - Subject to $x \geq 0$

13

General Form of an Optimisation Problem

MAX (or MIN): $f\,(X_1, X_2, \ldots, X_n)$

Subject to: $g_1(X_1, X_2, \ldots, X_n) <= b_1$

$$\vdots \qquad \vdots$$

$$g_k(X_1, X_2, \ldots, X_n) >= b_k$$

$$\vdots \qquad \vdots$$

$$g_m(X_1, X_2, \ldots, X_n) = b_m$$

Here *n* and *m* are fixed dimensions:

n = no. of variables (X_i), m = no. of constraints.

Each f, g_1, …, g_m is a function. If there exists at least one nonlinear function the problem is a Nonlinear Program (NLP).

14

Unconstrained optimization

The unconstrained minimization problem:

$$\min_{x \in \Re^n} f(x)$$

where $f : \Re^n \to \Re$.

- x^* is a **global minimum** if $f(x^*) \le f(x)$ for all $x \in \Re^n$.
- x^* is a **local minimum** if there exists an open neighborhood $B(x^*)$ of x^*, such that $f(x^*) \le f(x)$ for all $x \in B(x^*)$.

15

Unconstrained optimization: Optimality conditions

- Any vector $d \in \Re^n$ is a **descent direction** of f at x^* if there exists $\alpha^* > 0$ such that $f(x^* + \alpha d) < f(x^*)$ for all $\alpha \in (0, \alpha^*)$
- First order necessary condition:

  If x^* is a local minimum then (**no descent direction at x***)

  $$\nabla f(x^*) = 0.$$
- Second order necessary condition

  If x^* is a local minimum then

  $$\nabla f(x^*) = 0$$

  and

  $$x^T \nabla^2 f(x^*)\, x \ge 0 \text{ for all } x \in \Re^n.$$

16

Unconstrained optimization: Optimality conditions

- Any point $x \in \Re^n$ satisfying first order optimality condition is called **stationary**.

- Second order sufficient condition

  If x^* is a stationary point and

$$x^T \nabla^2 f(x^*)\, x > 0$$

  for all $x \in \Re^n$, with $x \neq 0$, then x^* is a strict local minimum.

17

Typical Optimisation Procedure

- Aim at seeking **stationary points**.
- Based on idea of **Steepest Descent** for minimizing an unconstrained nonlinear function
- At a given feasible point, find a **direction** along which you can improve your objective value (search direction)
- Then **search along the direction** (line search) until you find a better point and re-iterate from this point

- **BUT** if that direction leads out of the feasible set then must “correct” it (reduced direction)

18

Steepest Descent for Unconstrained Optimization

- Let f be a continuously differentiable function of several variables, $f : \Re^n \rightarrow \Re$.
- Consider the unconstrained problem: min $f(x)$
- At a given point x, the steepest descent direction is $d = -\nabla f(x)$
  - gradient of f at x is the vector of partial derivatives:

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)(x) \in R^n$$

- **To decrease f :**
  - **search along this direction:** find a step size $t > 0$ such that $f(x+td) < f(x)$.
  - Replace x by $x + td$ and **repeat** with the new steepest descent direction

19

In-Class Exercise

Steepest Descent Directions

What is the steepest descent direction for

1. $\cos(\pi x)$ at x=0 ?
2. $x_1/(1+ x_2^2)$ at x=(1,0) ?
3. $5x_1 - 2x_2 + x_3$?

For case 1 above, perform 2 steepest descent steps starting from x = 1/2 with unit stepsize (t = 1).

20

Steepest Descent Method for Unconstrained Optimization

Consider the unconstrained problem: min $f(x)$

- Initially
  - Require a starting vector x^0
  - Set iteration counter $k = 0$
- At iteration k
  - **Descent direction:** Let $d = -\nabla f(x^k)$
  - **Linesearch:** find a step size $t = t_k > 0$ such that

$$f(x^k + td) < f(x^k).$$

  - **Update:** $x^{k+1} = x^k + t_k d$, $k = k + 1$ and **repeat** from above

What does Steepest Descent Method achieve ?

Steepest Descent Convergence Theorem

- Given an initial "iterate" (vector) x^0, and iteration counter k=0:
- If f is bounded below and the step size $t_k > 0$ is properly chosen† for each iterate x^k, then the steepest descent method produces an infinite sequence $\{x^k\}$ where
  - ➔ $\nabla f(x^k) \rightarrow$ zero vector, on some subsequence of $\{x^k\}$
  - if any subsequence of $\{x^k\}$ converges to a vector x^* (a **limit point**), then $\nabla f(x^*) = 0$

† The linesearch must not choose t_k too large or too small. Linesearch methods will not be covered here.

Steepest Descent Method for Unconstrained Optimization

- That is, Steepest Descent Method tends to find **Stationary Points** of f
  - Remind that all minimizers of f are stationary points
- What happens if $\nabla f(x^k)$ = zero vector for some iteration k ?

23

Constrained optimization

The constrained minimization problem:

$$\min_{x \in R^n} \quad f(x)$$

$$\text{subject to} \quad g(x) \le 0$$

$$h(x) = 0$$

where $f : \Re^n \to \Re,\ g : \Re^n \to \Re^m,\ h : \Re^n \to \Re^p$.

- $x^* \in F = \{x \in \Re^n : g(x) \le 0 \text{ and } h(x) = 0\}$ is a **global minimum** if $f(x^*) \le f(x)$ for all $x \in F$.
- $x^* \in F$ is a **local minimum** if there exists an open neighborhood $B(x^*)$ of x^*, such that $f(x^*) \le f(x)$ for all $x \in F \cap B(x^*)$.

24

Constrained optimization: Optimality conditions

Need to define feasible descent directions at any point $x \in F$.

?

Will see next lecture!

Let's focus first on the practical side.

Constrained optimization: Optimality conditions and descent methods

- Any local minimum satisfies some necessary conditions, called **stationarity conditions**.
- Any point satisfying stationarity conditions is called **stationary**.
- Iterative (descent) methods look for stationary points.

Constrained Steepest Descent: Projected Gradient Method

- Let $f: R^n \rightarrow R$ be a continuously differentiable
- Consider the problem with LP constraints:

  min $f(x)$ subject to $Ax = b$, x nonnegative
- The **reduced gradient** or **projected gradient method** is just a version of steepest descent for optimization problems with constraints
  - Given a feasible x, let $d = -\nabla f(x)$
  - Motivation: want to decrease f by moving along ray $x + td$ where t starts at 0 and increases
  - **Problem:** $x + td$ may be **infeasible**

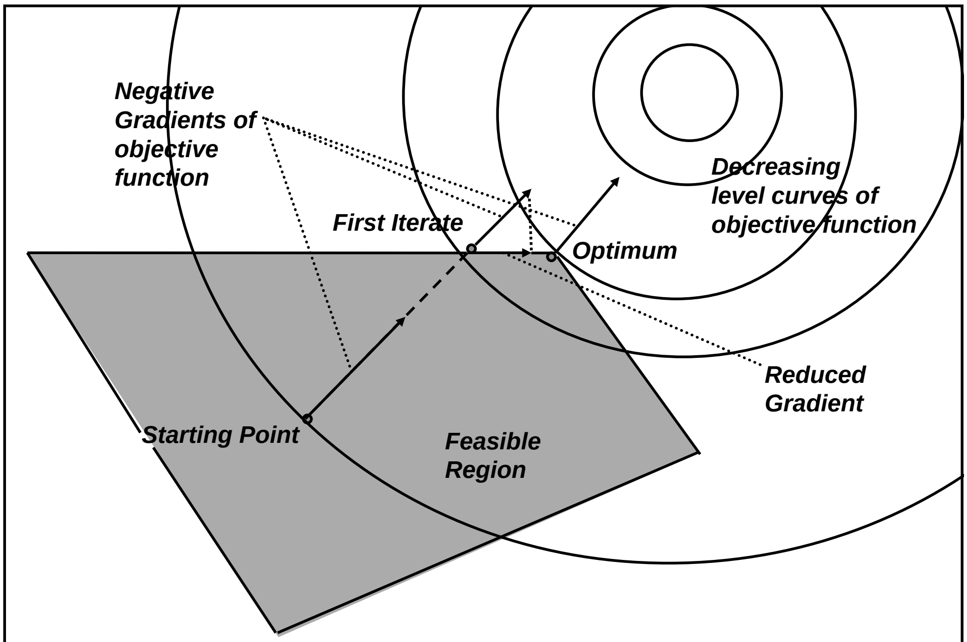
27

Constrained Steepest Descent: Projected Gradient Method

Solution is to convert any point of ray $x + td$ to a feasible point by <u>projection</u> onto feasible set:

- For any step size $t > 0$, find <u>nearest point</u> $y(t)$ in feasible set (polyhedron) to $x + td$
- Of course, $y(t)$ is feasible

28



Projected Gradient Method

29

Constrained Steepest Descent: Projected Gradient Method

In-class exercise

Let feasible set be the box $0 \le x \le 1$ in R^2,

$x = (1,0)$, and $f(x) = 5x_1 - 2x_2$.

1. Sketch the steepest descent ray $x + td$ and the projected gradient path $y(t)$.

2. Give a formula or formulae for $y(t)$.

30

Why does Projected Gradient Method Work ?

Result I: Projected Gradient Descent

Either $y(t) = x$ for all $t > 0$

Or $f(y(t)) < f(x)$ for small $t > 0$

Result II: Projected Gradient at a minimum

If x is a local minimum then $y(t) = x$ for all $t > 0$

These results say why the projected gradient path $y(t)$ is useful.

A useful by-product:

Definition:

A point x is **stationary** for our constrained minimization problem if $y(t) = x$ for all $t > 0$.

31

Formally:

Projected Gradient Method

- Consider the problem with LP constraints:

  min $f(x)$ subject to $Ax = b$, x nonnegative
- Initially
  - Require a feasible starting vector x^0
  - Set iteration counter $k = 0$
- At iteration k
  - **Descent direction:** Let $d^k = -\nabla f(x^k)$
  - **Projected gradient linesearch:** find step size $t = t_k > 0$ such that $f(y^k(t)) < f(x^k)$

    where $y^k(t)$ is projection of $x^k + td^k$ onto feasible set
  - **Update:** $x^{k+1} = y^k(t_k)$, $k = k + 1$ and **repeat** from above

32

What does Projected Gradient Method achieve ?

Direct extension of Steepest Descent Convergence:

Projected Gradient Convergence Theorem

- Given an initial "iterate" (vector) x^0, and iteration counter k=0:
- If f is bounded below and the step size $t_k > 0$ is properly chosen for each iterate x^k, then the projected gradient method produces an infinite sequence $\{x^k\}$ where
  - $y^k(t_k) - x^k \rightarrow$ zero vector, on some subsequence of $\{x^k\}$
  - if any subsequence of $\{x^k\}$ converges to a vector x^* then x^* is (feasible and) stationary for the constrained problem

33

Summary of Projected Gradient Method

- min $f(x)$ subject to $Ax = b$, x nonnegative
- **reduced gradient** or **projected gradient** method
  - Given a feasible x, let $d = -\nabla f(x)$
  - For any step size $t > 0$, find nearest point $y(t)$ in feasible set (polyhedron) to $x + td$
- **Fact:**
  - Either $f(y(t)) < f(x)$ for all small enough $t > 0$
  - Or $y(t) = x$ for all $t > 0$ … and we say x is a **stationary point** of the above problem
  - Note that all minimizers of above problem are stationary
- So projected gradient method makes sense
  - can show limit points of iteration sequence are stationary

34

Doing it in Excel: `Solver`

- Solver uses a version of projected gradient method this called the Generalized Reduced Gradient (GRG) algorithm to solve NLPs.
- GRG will attempt to track nonlinear feasible sets, which is generally much more difficult than dealing with LP feasible sets
- GRG can also be used on LPs (if you don’t select “assume linear model”) but is generally slower than the Simplex method.

35

Starting Points

- The produced local solution depends on the starting point
- If you think there may be a better point, try another starting point
- Instead of assigning an arbitrary value to variable cells, **start with values which are representative of the values you expect in the solution**
- Start close to where you expect the optimum to be if that information is available

36

A note on convergence

- In contrast to the simplex method, NLP methods often produce an **infinite** sequence of iteration points that converge to a (local) optimum
- The sequence needs to be stopped !!
- In Excel 8.0 & beyond, the convergence field in the Solver Options dialog box can be adjusted
  - increase to avoid **squillions** of iterations
  - decrease to avoid early stopping at sub-optimal solutions

37

Sensitivity Analysis

- Sensitivity report provides
  - Reduced Gradient
  - Lagrange Multipliers
- The reduced gradient information is the equivalent of reduced costs in LP
- Lagrange multipliers are the equivalent of shadow prices in LP

38

Lagrange Multipliers

- Amount by which the objective function would improve if the RHS of the constraint was relaxed by one unit
- For equality constraints: Amount by which the objective function would improve if the RHS of the constraint was increased by one unit
- Again: For NLP this is derivative information and therefore it isn't possible to give a range of RHS values for which Lagrange multiplier (shadow price) is fixed.

Lagrange Multipliers & Shadow Prices I

- Suppose x^* is a stationary point of the NLP

$$\min f(x) \quad \text{subject to} \quad Ax = b, \; x \geq 0$$

- Then
  - x^* solves the LP

    $$\min \nabla f(x^*)^T x \quad \text{subject to} \quad Ax = b, \, x \geq 0.$$

  - Lagrange multiplier for NLP at x^* coincides with shadow cost for LP
  - (if one is unique then so is the other)

40

Lagrange Multipliers & Shadow Prices II

- **Suppose** x^* is a stationary point of the NLP:

  min $f(x)$ subject to $h(x) = 0$, $x \geq 0$

  where $f, h : R^n \rightarrow R$ (single equality constraint)
- Define $c = \nabla f(x^*)$, $a = \nabla h(x^*)$, $b = \nabla h(x^*)^T x^*$ ($\in$ R)
- **Then**
  - x^* solves the following LP with a single equality constraint:

    min $c^T x$ subject to $a^T x = b$, $x \geq 0$
  - Lagrange multiplier for NLP at x^* coincides with shadow cost for LP
- **Technical requirement:** $\partial f(x^*)/\partial x_i \neq 0$ for some nonzero component of x^*

41

Lagrange Multipliers & Shadow Prices III

- **Suppose** x^* is a stationary point of the NLP:

  min $f(x)$ subject to $h(x) = 0$, $x \geq 0$

  where $h : R^n \rightarrow R^m$ (m equality constraints),

  i.e. $h(x) = (h_1(x), \ldots, h_m(x))$.

Q: How do we define the associated LP at x*?

A: Define

$c = \nabla f(x^*)$,

A = matrix of m rows $\nabla h_1(x^*)^T, \ldots, \nabla h_m(x^*)^T$

$b = \nabla h(x^*)^T x^*$ ($\in R^m$)

42

Automatic Scaling

- Use automatic scaling option since poor scaling can result in breakdown of the method, in particular in NLP
- For automatic scaling to work properly it is important that your initial variables have "typical" values

43

Solver Message 1

"***Solver found a solution. All constraints and optimality conditions are satisfied***."

- This means Solver found a "stationary" solution at which **necessary** optimality conditions are satisfied
- In most cases such stationary solutions will be locally optimal solutions but there is no guarantee
- It is not at all guaranteed that the solution is a globally optimal solution
- Run Solver from several different starting points to increase the chances that you find the global optimal solution to your problem.

44

Solver Messages 2-4 relate to various difficulties

- Will not go over these
- See next 3 slides for descriptions

45

Solver Message 2

"**Solver has converged to the current solution. All constraints are satisfied**."

- This means the objective function value changed very slowly for the last few iterations.
- If you suspect the solution is not locally optimal, your problem may be poorly scaled.

46

Solver Message 3

"**Solver cannot improve the current solution. All constraints are satisfied**."

– This rare message means that your model is degenerate and the Solver is "cycling"
– Degeneracy can sometimes be eliminated by removing redundant constraints in a model.

47

Solver Message 4

"**Solver could not find a feasible solution.**"

– In NLP this can occur in feasible problems.
– Solver is not able to reduce the sum of infeasibilities
– Possible remedies: try another starting point, include automatic scaling, decrease convergence tolerance

48

Lecture 7 *3E4 Homework*

1. Find all stationary points of
   a) $\cos(\pi x)$
   b) $x_1/(1+ x_2^2)$
   c) $5x_1 - 2x_2 + x_3$

2. What is the steepest descent direction for
   $5x_1 - 2x_2 + x_3$?

3. Perform two steepest descent steps on the function $\cos(\pi x)$ starting from 1/2 with unit stepsize ($t = 1$). I.e. take $x^0 = 1/2$ and calculate x^1 and x^2.

Lecture 7 *3E4 Homework*

4. Suppose we want to minimise $f(x)= 5x_1 - 2x_2$ subject to x in the box $0 \le x \le 1$ in R^2. Starting from $x^0 = (1,0)$, take two steps of the projected gradient method with stepsize t=1/3. Show x^0, x^1, x^2 on a sketch of the feasible set.

Lecture 7 *3E4 Homework*

5. Consider minimising $5x_1 - 2x_2 + x_3$ subject to nonnegative variables and $x_1^2 + x_2^2 + x_3^2 = 4$.

a) Verify that $x^* = (0,2,0)$ is optimal for the LP associated with this point.

b) Find the Lagrange multiplier for x^* by calculating the shadow price of the LP in part (a) at this point.

c) How will the optimal value of the NLP change to if the RHS value is changed from 4 to 4.5?

d) Check your answer to part (b) using the Excel sensitivity report.

e) Check your answer to part (c) by re-solving with Excel using RHS 4.5. Is there a difference? Comment briefly on this.

51

Lecture 7 *3E4 Homework*

Hints for Q5, parts (a), (b):

- Write down the linear program corresponding to x^*, as described previously
- Rewrite LP as a max problem, and analyse it as described in Lecture 3:
  - find reduced cost of x^* to show optimality
  - find shadow cost
- (How is shadow cost of max LP related to shadow cost of min LP? hence to Lagrange multiplier of NLP at x^* ?)

52