Imperial College London

UNIVERSITY OF LONDON

BSc and MSci EXAMINATIONS (MATHEMATICS)

May-June 2005

This paper is also taken for the relevant examination for the Associateship.

M2OD Graphs, Algorithms and Optimisation

Date: Tuesday, 17th May 2005 Time: 2 pm – 4 pm

Credit will be given for all questions attempted but extra credit will be given for complete or nearly complete answers.

Calculators may not be used.

1. Consider the portfolio problem

To maximise $x_1 + 3x_2 + 4x_3 + 5x_4$,
subject to $x_1 + x_3 \geq 9$, $x_3 + x_4 \leq 7$, $x_1 + x_2 + x_3 + x_4 = 20$,
with $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$, $x_4 \geq 0$.

(a) Express the above portfolio problem as a linear programming problem in standard form using slack variables x_5 and x_6 and extra variable x_7, so that $\{x_2, x_6, x_7\}$ form a basic set of variables giving a non-degenerate starting vertex for applying the simplex algorithm.

(b) Use the simplex algorithm with the starting basic set $\{x_7, x_6, x_2\}$ to obtain an optimal vertex and the optimal value of the objective function.

You may wish to use that if $A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & j \end{pmatrix}$

then

$$\det(A) = aej + bfg + cdh - gec - dbj - ahf$$

and

$$A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} ej - fh & ch - bj & bf - ce \\ fg - dj & aj - cg & cd - af \\ dh - eg & bg - ah & ae - bd \end{pmatrix}$$

2. (i) (a) Prove that the dual of the dual linear programming problem (DLP)

$$\max_y b^\top y \quad \text{subject to} \quad A^\top y \leq c, \quad \text{with} \quad y \geq 0$$

is the primary linear programming problem (PLP)

$$\min_x c^\top x \quad \text{subject to} \quad A\,x \geq b, \quad \text{with} \quad x \geq 0\,.$$

(b) For the linear programming problem

To maximise $-x_1 + 2x_2 - 3x_3 + x_4$,
subject to $x_3 - x_4 \leq 0$, $x_1 - 2x_3 \leq 1$, $2x_2 + x_4 \leq 3$, $-x_1 + 3x_3 \leq 5$,
with $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$, $x_4 \geq 0$

express it as a PLP in the form given in (a), and hence write down its DLP in the form given in (a).

Question 2 is continued on Page 3

2. (ii) Let G be a graph. Define the following terms.

(a) G is *connected.*

(b) A *connected component* of G.

Given a graph G, a new graph D is constructed as follows. The vertices of D are pairs (v, w) where v, w are vertices of G, and there is an edge in D joining (v, w) to (v', w') in the following circumstances:

(1) $v = v'$ and there is an edge $w - w'$ in G;

(2) $w = w'$ and there is an edge $v - v'$ in G;

(3) there are edges $v - v'$ and $w - w'$ in G.

If G is connected, is D connected?

If D is connected, is G connected?

3. Let G be a connected graph. Define the following terms.

(i) An *Eulerian trail* in G.

(ii) An *Eulerian circuit* in G.

State *Euler's Theorem*, and show how the part relating to trails can be deduced from the part relating to circuits.

Say what is meant for a graph to be a *tree*.

Can a tree have an Eulerian circuit or trail?

A new graph H is constructed from a given graph G as follows. We add k new vertices $w_1, \ldots, w_k$ and kn new edges $w_i - v_j$, $i = 1, \ldots, k$, $j = 1, \ldots, n$, where $v_1, \ldots, v_n$ is the vertex set for G.

Given that G has a Eulerian circuit, determine, in terms of k, n when H has an Eulerian circuit.

If H does not have an Eulerian circuit, determine the smallest number of edges that have to be added to H to produce a graph that does have an Eulerian circuit.

4. Let G be a simple connected graph. Show how to construct a *metric* d on (the vertices of) G.

Suppose that at least one vertex of a graph G has valency less than $k = \max\{\delta(v) \mid v \text{a vertex of} G\}$. Prove that $\chi(G) \leq k$, where $\chi(G)$ is the chromatic number.

Show that, for every pair h, n of positive integers with $2 \leq h \leq n$, there is a simple connected graph with $\chi(G) = h$, $|V| = n$.

5. Define the following terms.

(i) A *source* in a directed graph.

(ii) A *sink* in a directed graph.

(iii) A *network*.

State the *Maximal Flow, Minimal Cut* Theorem.

The figure below is a network, with edge directions indicated by the arrows, and capacities given in brackets. Starting with the zero flow, find a maximal flow in the network. [*Give the flow augmenting path that you use in each step.*]

Confirm that your flow is maximal by finding the capacity of an appropriate cut.