UNIVERSITY OF CAMBRIDGE

MATHEMATICAL TRIPOS Part III

Thursday 1 June 2006 1.30 to 4.30

PAPER 38

MATHEMATICS OF OPERATIONAL RESEARCH

*Attempt **FOUR** questions.*

*There are **SIX** questions in total.*

The questions carry equal weight.

STATIONERY REQUIREMENTS
Cover sheet
Treasury Tag
Script paper

SPECIAL REQUIREMENTS
None

You may not start to read the questions printed on the subsequent pages until instructed to do so by the Invigilator.

1 Consider the linear programming problem:

$$\begin{aligned}
&\text{maximize } 4x_1 + 2x_2 + x_3 \\
&\text{subject to} \\
&x_1 \qquad\qquad\qquad\quad \leqslant 5 \\
&4\,x_1 \;+\; x_2 \qquad\quad \leqslant 25 \\
&8\,x_1 \;+\; 4\,x_2 \;+\; x_3 \qquad \leqslant 125
\end{aligned}$$

and $x_1, x_2, x_3 \geqslant 0$. Solve this with the simplex algorithm, starting from $x = (0, 0, 0)$, and using the rule that whenever there is a choice as to which variable should next enter the basis it should selected as the one that produces the greatest increase in the objective function per unit increase in that variable.

Is there a pivot selection rule under which the problem would have been solved more quickly?

Discuss the worst-case running time of the simplex algorithm.

2 Explain what is meant by the minimum-cost flow problem.

A project of n tasks is to be completed as quickly as possible. We may work on more than one task at the same time, but we are subject to precedence constraints expressed in the matrix $a = (a_{ij})$, such that if $a_{ij} = 1$ we may not start task j until task i is complete; otherwise $a_{ij} = 0$ and there is no precedent constraint between i and j. Task i has processing time τ_i, $i = 1, \ldots, n$. Let t_i be the earliest time at which task i can be started. Formulate as a linear program the problem of minimizing $t_{n+1} - t_0$, where t_0 and t_{n+1} are the times at which the project starts and finishes.

Show that the dual LP can be expressed as an uncapacitated minimum cost flow problem on a graph (N, A) where $(i, j) \in A$ if and only if $a_{ij} = 1$ and the cost on arc (i, j) is $c_{ij} = -\tau_i$.

Illustrate an algorithm that can be used to solve any minimum cost flow problem by applying it to the project of 6 tasks defined by the following data. Start your explanation at a basic feasible solution corresponding to the tree with arcs $\{(0,1), (2,4), (4,5), (5,7), (4,6), (1,3), (3,5)\}$.

$$a = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \tau = (3, 4, 5, 6, 4, 3).$$

3 Explain in terms of the theory of computational complexity what it means to say that a problem Π is no harder than another problem Π'.

Let A be the $m \times n$ payoff matrix for a two-person zero-sum game in which players 1 and 2 have m and n pure strategies respectively. Given A and a number V, let Π be the problem of determining whether the value of the game equals V. Sketch an argument to show that if $\mathcal{P} \neq \mathcal{NP}$ then Π is not $\mathcal{NP}$-hard.

In an instance of the Subset Cover Problem, SCP, we are given subsets $S_1, \ldots, S_m$ of $S = \{1, \ldots, n\}$, and a number $k < m$, and ask 'Are there k of these subsets whose union is S?' Consider the non-zero-sum game in which if player 1 plays pure strategy $i \in \{1, \ldots, m\}$ and player 2 plays pure strategy $j \in \{0, 1, \ldots, n\}$, the payoff is

$$((e_1(i,j), e_2(i,j)) = \begin{cases} (1,1) & j = 0, \\ (1,0) & j \geqslant 1 \text{ and } j \in S_i, \\ (0, \frac{k}{k-1}) & j \geqslant 1 \text{ and } j \notin S_i. \end{cases}$$

Show that if and only if the answer to the SCP instance is 'Yes' does there exist an equilibrium in which player 1 randomizes with positive probabilities over just k of his pure strategies.

Comment on the difficulty of computing all equilibria of a two-person non-zero-sum game.

4 List the assumptions of a symmetric independent private values (SIPV) auction.

State the revenue equivalence theorem.

Let $e(p)$ denote the minimal expected payment that a bidder can make if he wishes to win an SIPV auction with probability p. Suppose that when a bidder with valuation v seeks to maximize his expected profit he does so by choosing $p = p(v)$ as a stationary point of his expected profit function. Show that

$$\frac{de(p)}{dp}\Big|_{p=p(v)} = v, \quad \frac{de(p(v))}{dv} = v\frac{dp}{dv}, \quad \text{and} \quad e(p(v)) = vp(v) - \int_0^v p(w)\,dw\,.$$

Consider a 'lowest-price auction' amongst n bidders in which the highest bidder wins but he pays only the lowest bid. Assume that bidders' valuations are independent and uniformly distributed on $[0, 1]$. Show that, at the equilibrium, the seller's expected revenue is $(n-1)/(n+1)$.

Suppose that when $n = 3$ there exists a contant A such that it is optimal in equilibrium for a bidder with valuation v to bid Av. Find A.

5 Define the meaning of an equilibrium in a multi-person game.

In a small town there are just 3 residents. A proposition regarding taxes is favoured by residents B and C, but opposed by A. It will be passed in a ballot if and only if it receives more votes in favour than against. Each resident has a cost of going to the polls to vote of $3c$. If the proposition passes, each of B and C will gain by $4c$ and A will lose $8c$. Suppose A decides to go to the polls with probability α and each of B and C go independently with probability β.

Find a condition that must be satisfied by β if there is to be an equilibrium with $0 < \alpha < 1$.

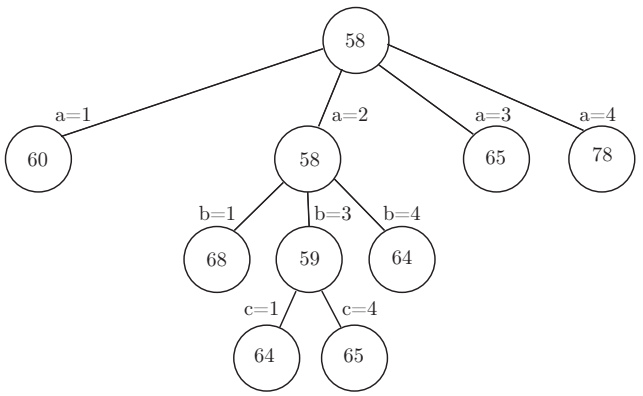
Show that there is an equilibrium of $\alpha = 1$, $\beta = 3/4$.

Is this the only equilibrium?

6 Describe the methodology of branch and bound algorithms.

Consider an assignment problem in which four machines, a, b, c, d are to be assigned to four tasks 1, 2, 3, 4 at minimal cost. The costs of assigning machines to tasks are given in the matrix below. At the first stage of a branch and bound algorithm there are four branches, in which machine a is assigned to either task 1, 2, 3 or 4. If machine a is assigned to task 2, then a lower bound is computed by adding this cost (12) to the minimum costs with which each of the unassigned jobs, 1, 3, 4, can be assigned to some unassigned machine, e.g., 11, 13, and 22, respectively, for a total lower bound of 58. Branches from this node, are those for which machine b is assigned to jobs 1, 3, or 4. In the branch in which machine b is assigned to task 1 we would have a lower bound of $12+14+19+23 = 68$. Using a branching rule that branches on the node with least lower bound, complete the branch and bound algorithm that is begun in the figure below, and find the optimal assignment.

	1	2	3	4
a	11	12	18	40
b	14	15	13	22
c	11	17	19	23
d	17	14	20	28



END OF PAPER