5.16 AS MODULE 4736: DECISION MATHEMATICS 1 (D1)

Preamble

Knowledge of the specification content of Modules *C1* and *C2* is assumed, and candidates may be required to demonstrate such knowledge in answering questions in Unit *D1*.

The specification content of this module is to be understood in the context of modelling real-life situations, and examination questions may ask for comment and interpretation, including where appropriate cross-checking between a model and reality.

Algorithms

Candidates should be able to:

(a) understand the definition of an algorithm;

(b) appreciate why an algorithmic approach to problem-solving is generally preferable to *ad hoc* methods, and understand the limitations of algorithmic methods;

(c) understand the meaning of the order of an algorithm, and determine the order of a given algorithm in simple cases, including the algorithms for standard network problems;

(d) interpret and apply simple algorithms defined by flow diagrams or given as a listing in words;

(e) show familiarity with simple algorithms concerning sorting and packing, including

   (i) bubble and shuttle sorts,

   (ii) first-fit methods (first-fit and first-fit decreasing).

Graph Theory

Candidates should be able to:

(a) understand the meaning of the terms ‘arc’ (or ‘edge’), ‘node’ (or ‘vertex’), ‘path’, ‘tree’ and ‘cycle’;

(b) use the orders of the nodes in a graph to determine whether the graph is Eulerian or semi-Eulerian or neither;

(c) solve simple problems involving planar graphs, both directed and undirected.

Networks

Candidates should be able to:

(a) recall that a network is a graph in which each arc is assigned a 'weight', and use networks as mathematical models;

(b) apply Prim's and Kruskal's algorithms in solving the minimum connector problem to find a minimum spanning tree (including the use of a matrix representation for Prim's algorithm);

(c) find a solution to the travelling salesperson problem in simple cases, and in other cases

(i) determine an upper bound by using the nearest neighbour method,

(ii) use short-cuts where possible to improve on an upper bound,

(iii) use minimum connector methods on a reduced network to determine a lower bound;

(d) use Dijkstra's algorithm to determine the shortest path between two nodes;

(e) solve simple cases of the route inspection problem for at most six odd nodes by consideration of all possible pairings of the odd nodes.

Linear Programming

Candidates should be able to:

(a) formulate in algebraic terms a real-world problem posed in words, including the identification of relevant variables, constraints and objective function;

(b) set up a linear programming formulation in the form 'maximise (or minimise) objective, subject to inequality constraints and trivial constraints of the form $variable \geq 0$', and use slack variables to convert inequality constraints into equations together with trivial constraints;

(c) carry out a graphical solution for 2-variable problems, including cases where integer solutions are required;

(d) use the Simplex method for maximising an objective function, interpret the values of the variables and the objective function at any stage in the Simplex method.