Pearson Edexcel Level 3 Advanced GCE in Further Mathematics (9FM0)

Decision Mathematics 2

Scheme of Work

For first teaching from September 2017

ALWAYS LEARNING

Contents

Introduction

Decision Mathematics 2 Overview

Decision Mathematics 2 Units

INTRODUCTION

This scheme of work is based upon a five-term model over one year for A level Further Mathematics students and it is to be delivered **after completing A level Mathematics** (consecutive delivery model). It can be used directly as a scheme of work for the A level Further Mathematics specification (9FM0).

The scheme of work is broken up into units and sub-units, so that there is greater flexibility for moving topics around to meet planning needs.

Each unit contains:

- contents, referenced back to the specification (A level Further Mathematics Specification)
- prior knowledge (references to A level Mathematics Specification and SoW)
- keywords

Each sub-unit contains:

- recommended teaching time, though of course this is adaptable according to individual teaching needs
- objectives for students at the end of the sub-unit
- teaching points
- opportunities for problem-solving/modelling etc.
- common misconceptions/examiner report quotes (from legacy Specifications)
- notes

AS content is indicated in this document using bold black font in the overview, specification references and objectives

Teachers should be aware that the estimated teaching hours are approximate and should be used as a guideline only.

Our free support for AS and A level Further Mathematics specifications can be found on the Edexcel mathematics website (http://qualifications.pearson.com/en/home.html) and on the Emporium (www.edexcelmaths.com).

Overlap, links and dependencies with other Units' content is clearly indicated throughout.

Decision Mathematics 2

Unit		Title	Estimated hours
1		**Transportation Problems**	
	a	North-west corner method	2
	b	Stepping stone method	7
	c	Formulation as a linear programming problem	2
2		**Allocation (Assignment) Problems**	
	a	**The Hungarian Algorithm**	**12**
	b	Formulation as a linear programming problem	2
3		**Flows in Networks**	
	a	**Cuts**	**2**
	b	**The labelling procedure**	**7**
	c	**Maximum Flow-Minimum Cut Theorem**	**4**
	d	Multiple sources and sinks	3
	e	Optimal flow rates	2
4		**Dynamic Programming**	
	a	Network and table form for Dynamic Programming	9
5		**Game Theory**	
	a	**Two-person zero-sum games**	**2**
	b	**Play-safe strategies**	**3**
	c	Dominance	1
	d	**Optimal mixed strategies using graphical methods**	**6**
	e	Optimal mixed strategies using the Simplex algorithm	4
6		**Recurrence Relations**	
	a	**Modelling using recurrence relations**	**3**
	b	**Solving first order recurrence relations**	**6**
	c	Solving second order recurrence relations	5
7		**Decision Analysis**	
	a	Decision trees	5
	b	Expected monetary values (EMVs)	3
			45 (AS) 90 (A Level)

UNIT 1: Transportation Problems

Return to Overview

SPECIFICATION REFERENCES

1.1 The north-west corner method for finding an initial basic feasible solution.

1.2 Use of the stepping-stone method for obtaining an improved solution. Improvement indices.

1.3 Formulation of the transportation problem as a linear programming problem.

PRIOR KNOWLEDGE

A Level Further Mathematics - Decision Mathematics 1

1 Introduction to algorithms (See SoW Unit 1a)

KEYWORDS

Supply (points), demand (points), source, destination, unit cost, north-west corner method, initial solution, balanced problems, unbalanced problems, dummy location, degenerate solutions, shadow costs, improvement indices, stepping-stone method, entering cell, exiting cell, optimal solution, decision variables, objective function, constraints.

1a. North-West corner method (1.1)	Teaching Time 2 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- understand and use terminology used to describe and model transportation problems;
- be able to find an initial solution using the north-west corner method.

TEACHING POINTS

Students should be familiar with the terminology supply/source, demand/destination, and unit cost, and be able to read a cost matrix.
Students will then need to find an initial solution using the north-west corner method.
In the **north-west corner method,** the upper left-hand cell is considered first and as many units as possible sent by this route before moving towards the lower right-hand cell.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

It is worth being aware of, and investigating the different solutions produced by the north-east, south-west and south-east methods, but for the examination only the north-west corner method will be tested.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

When the row and column requirements are both met within one cell, students can be unsure where to move next.

NOTES

Problems will be restricted to a maximum of four sources and four destinations.

1b. Stepping Stone Method (1.2)	**Teaching Time** 7 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- be able to find shadow costs;
- be able to find improvement indices;
- be able to use the stepping-stone method;
- understand what is meant by a degenerate solution and know how to deal with it;
- be able to understand what is meant by an unbalanced transportation problem and how to use dummy locations.

TEACHING POINTS

The **shadow costs** R_i , for the ith row, and K_j , for the jth column, are obtained by solving $R_i + K_j = C_{ij}$ for **occupied cells**, taking $R_1 = 0$ arbitrarily.

The **improvement index** I_{ij} for an **unoccupied cell** is defined by $I_{ij} = C_{ij} – R_i – K_j$.

The **stepping stone method** is an iterative procedure for moving from an initial feasible solution to an optimal solution. It is usual to use the cell with the most negative improvement index as the **entering cell** (ie the start of the stepping stone route). The cell which has the number of transportation units reduced to zero is the **exiting cell**. Students should be reminded not to enter a 0 in this cell, but instead to leave it blank, so that they treat it as an unoccupied cell when testing for optimality.

The solution is optimal when there are no negative improvement indices.

Degeneracy occurs in a transportation problem, with m rows and n columns, when the number of occupied cells is less than $(m + n – 1)$. To deal with a degenerate solution, we place a zero in a currently unused cell, and treat it as occupied.

When total supply ≠ total demand, the problem is **unbalanced**. To deal with an unbalanced problem, we add a dummy demand point or supply point so that total supply = total demand, with transportation costs of zero.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Ask questions such as:

- When is a degenerate initial solution reached?
- If you have a degenerate solution, try placing your “0” in different cells. What difference does this make?
- Rather than always using the most negative improvement index to indicate your entering cell, try using other cells as the entering cell. What difference does this make?

Students could attempt, as a group, solutions for problems with more than 4 supply or demand points.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

Most students are able to find a simple stepping stone route, though some add an extra ϑ in one cell, not understanding the requirement for balance both across rows, and down columns. When a stepping stone route involves “skipping” cells, students find this much harder to accurately identify. Once a correct stepping stone route is found, most students go on to give the correct improved solution in part (b), although some have an additional zero in the exiting cell, which is a costly mistake as it results in them missing an improvement index in the next iteration.
If students are told the entering cell in the question, it is not necessary to calculate the shadow costs and improvement indices to verify this, unless requested to do so. Students can get confused and use the supply pattern instead of the costs, as the basis for their shadow cost/improvement index calculations. Some students lose marks for failing to state the correct exiting cell.

NOTES

The ideas of dummy locations and degeneracy are required.
Students should identify a specific entering cell and a specific exiting cell.

1c. Formulation as a linear programming problem (1.3)	**Teaching Time** 2 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- be able to formulate a transportation problem as a linear programming problem.

TEACHING POINTS

Students should be familiar with:

- defining decision variables
- writing the objective function
- writing all constraints.

This is a straight-forward section to learn, however, students should be reminded to take care and be precise with their solution. It is common for students to define their decision variables in one way and then use different variables for their constraints or objective function.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Any questions attempted with the stepping stone method could be formulated as a linear programming problem. It is also interesting to ask students to identify and model local business transportation problems – collect data from a local company, then simplify and model their scenario.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

Students often fail to make a clear definition of their x_{ij} and then use it consistently throughout the question. Common errors include omitting the word "number" or using P, Q, R in the definition and 1, 2, 3 elsewhere. Some students will get confused with the transportation and allocation problems and define x_{ij} as being equal to 1 or 0 as in an allocation formulation.

Most students are able to correctly state the objective function and "minimise", although a small number make slips, either with the coefficients or suffices. There are often variety of errors made with the constraints, with some not having unit coefficients and commonly the non-negativity constraint for x_{ij} being absent. Other errors include errors with suffices or values and a mixture of equations and inequalities. A small number of students incorrectly write all their constraints as $\geq$. Some students change their notation between the objective function and constraints, for example using PA in the objective function and x_{PA} in the constraints.

UNIT 2: Allocation (Assignment) Problems

Return to Overview

SPECIFICATION REFERENCES

2.1 **Cost matrix reduction. Use of the Hungarian algorithm to find a least cost allocation. Modification of the Hungarian algorithm to deal with a maximum profit allocation.**

2.2 Formulation of the Hungarian algorithm as a linear programming problem.

PRIOR KNOWLEDGE

A Level Further Mathematics - Decision Mathematics 1

1 Introduction to algorithms (See SoW Unit 1a)

KEYWORDS

Allocation, assignment, cost matrix, row/column reduction, unbalanced problem, element, Hungarian algorithm, dummy location, incomplete data, minimisation (cost) allocation, maximisation (profit) allocation, decision variables, objective function, constraints

2a. The Hungarian algorithm (2.1)	Teaching Time 12 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- **be able to reduce cost matrices;**
- **be able to use the Hungarian algorithm to find a least cost allocation;**
- **be able to adapt the Hungarian algorithm to use a dummy location;**
- **be able to adapt the Hungarian algorithm to manage incomplete data;**
- **be able to modify the Hungarian algorithm to deal with a maximum profit allocation.**

TEACHING POINTS

In allocation problems we are seeking a one-to-one solution where each worker does one task, and each task must be done by just one worker, so we need the same number of workers and tasks.

Consider initially a balanced problem, ie where the number of workers = number of tasks, where the cost is to minimised.

Students first need to understand how to reduce cost matrices by subtracting the row minimums and the column minimums from all elements in the corresponding row/column so that relative costs can be compared.

Students then need to use the Hungarian algorithm to find a least cost allocation.

If students do not have an $n \times n$ problem, students should be able to enter a dummy row or column with zero elements to enable the Hungarian algorithm to be used.

If there is incomplete data in a matrix (ie it is not possible to assign a task to a given person), a large value should be entered in this cell to make it an inefficient allocation to make.

To solve a maximisation problem (ie where the profit is to be maximised), students should subtract every number in the original matrix from the largest number in the matrix, and then proceed as usual.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Investigate drawing the lines in different places during the Hungarian algorithm – what difference does this make to reaching a final solution?

Why does the modification for a maximisation problem work? What is happening?

Students could attempt, as a group, solutions for problems with a larger number of workers and tasks.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

Students can be confused by the combination of a maximisation problem with empty cells. On such a question, some students entered large values in the empty cells before minimising, replaced the empty cells with zeros after converting to a minimisation problem, or dealt with them midway through their solutions (usually after row/column reduction). Some students did not fully undertake row and column reduction, in some cases prematurely augmenting their tables.

There were many students who began their solution as a minimisation problem and then realised their mistake and restarted.

The number of arithmetic errors is small but significant, sometimes caused by students misreading their own handwritten numbers.

NOTES

This topic can be linked to matchings and bipartite graphs, which is not part of this specification. However, students may find it useful to visualise an allocation problem using a bipartite graph.

Students should reduce rows first. The idea of a dummy location is required.

The adaption of the algorithm to manage incomplete data is required.

To deal with a maximum profit allocation, students should subtract all the values (in the original matrix) from the largest value (in the original matrix).

2b. Formulation as a linear programming problem (2.2)	**Teaching Time** 2 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- be able to formulate an allocation problem as a linear programming problem.

TEACHING POINTS

Students should be familiar with:

- defining decision variables
- writing the objective function
- writing all constraints.

This is a straight-forward section to learn, however, students should be reminded to take care and be precise with their solution. It is common for students to define their decision variables in one way and then use different variables for their constraints or objective function.

Parallels should be drawn with the transportation problem (ie the allocation problem is a special case of the transportation problem), however, students should be clear about the differences between them and how this relates to the formulation of the linear programming problem.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

It is useful to discuss the use of binary coding here, which could lead in to some investigative work on binary coding.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

Some students have completely misunderstood a question to formulate a problem as a linear programming problem, instead routinely setting about using the Hungarian algorithm. Many students make a confused attempt at defining their variables or define them in a way that is inconsistent with their later work, where a minority lose a variable altogether, using the subscripts only (see also Unit 1 Transportation Problems above). With maximisation problems, those who modify their matrix generally used these values in their objective function, although a number state that they were maximising instead of minimising. Others used the original matrix when writing the objective function. A significant number of candidates made errors with their constraints, by writing them as inequalities or using coefficients other than 1 or with inconsistent notation.

UNIT 3: Flows in Networks

Return to Overview

SPECIFICATION REFERENCES

3.1 Cuts and their capacity.

3.2 Use of the labelling procedure to augment a flow to determine the maximum flow in a network.

3.3 Use of the max–flow min–cut theorem to prove that a flow is a maximum flow.

3.4 Multiple sources and sinks. Vertices with restricted capacity.

3.5 Determine the optimal flow rate in a network, subject to given constraints.

PRIOR KNOWLEDGE

A Level Further Mathematics - Decision Mathematics 1

1 Introduction to graph theory (See SoW Unit 1c)

KEYWORDS

Capacitated directed networks/graphs or capacitated digraphs, capacity, source, sink, flow, feasibility condition, conservation condition, saturated, cut, initial flow, labelling procedure, flow-augmenting route, backflow, maximum flow-minimum cut theorem, supersource, supersink.

3a. Cuts (3.1)	**Teaching Time** 2 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- **understand and use the terminology used in analysing flows through networks;**
- **be able to calculate the capacity (value) of a cut through a network.**

TEACHING POINTS

Students should be familiar with finding cuts and their values, and evaluating the values of given cuts, dealing correctly with the direction of arcs across the cut.

A **cut**, in a network with source *S* and sink *T*, is a set of arcs (edges) whose removal separates the network into two parts *X* and *Y*, where *X* contains at least *S* and *Y* contains at least *T*.

The **capacity of a cut** is the sum of the capacities of those arcs in the cut which are directed from *X* to *Y*.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Students can investigate the value (and position) of all cuts for a network – how can they be sure they have found them all?

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

Many errors are often seen in calculating cuts, either by including the flow on arcs rather than the capacity, or included capacities of arcs crossing from sink to source.

NOTES

Only networks with directed arcs will be considered.
Problems may include both upper and lower capacities.

3b. The labelling procedure (3.2)	**Teaching Time** 7 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- **be able to find an initial flow through a capacitated directed network;**
- **be able to start from an initial flow and use the labelling procedure to find flow-augmenting routes to increase the flow through the network.**

TEACHING POINTS

Students should be familiar with the vocabulary associated with flows through networks, and interpret the information given on networks.

A flow through a network needs to satisfy two conditions:

- The feasibility condition – the flow along each arc cannot exceed the capacity of that arc;
- The conservation condition – for all vertices except the source and sink vertices, the total flow in to a vertex = total flow out of a vertex.

If flow = capacity for an arc, then that arc is saturated.

Students will need to find an initial flow through a network, even though it is often given in an examination. From there, they should be familiar with labelling a network by drawing two arrows on each arc, known as the labelling procedure:

- The "forward" arrow (in the same direction as the arc) identifies the spare capacity on that arc;
- The "backward" arrow (in the opposite direction to the arc) identifies the current flow on that arc.

Flow-augmenting routes from source to sink can then be found. In a flow-augmenting route, it is possible to "travel" backwards down an arc, as long as there is capacity in the direction in which you wish to move (this is known as backflow). The practical interpretation of this is that the flow down an arc is reduced and redirected elsewhere, but this is done holistically by dealing with a complete route from source to sink – students should not consider "re-directing" flow on single arcs, as they would not then be using a flow-augmenting route.

Whilst the labelling procedure is very useful for finding the maximum flow through a network, it can often produce messy diagrams which can be very difficult for an examiner to check. Encourage students to put one neat line through each number that is changed, and to always list their flow-augmenting routes and values outside of their diagram.

The final flow diagram should only have one number on each arc (the flow). It is useful to check that the conservation condition holds at every vertex to ensure that errors have not been made when changing numbers during the labelling procedure.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Using the feasibility and conservation conditions to find missing flows/capacities on arcs.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

The vast majority of students can find an initial flow and complete a flow diagram. Whilst most can find one or more flow augmenting routes, a significant number fail to obtain the maximum flow. Sometimes students try to increase the flow down an arc by too much, not taking account of the direction of the arc. Students also commonly incorrectly make statements about decreasing the flow in particular arcs, rather than using “backflow” correctly as part of the labelling procedure. When students are asked to show their final flow pattern, it is common to see either one arc left blank or an inconsistent flow pattern, so it is advisable to methodically check every node for ‘flow in = flow out’.

NOTES

The arrow in the same direction as the arc will be used to identify the amount by which the flow along that arc can be increased. The arrow in the opposite direction will be used to identify the amount by which the flow in the arc could be reduced.

3c. Maximum Flow – Minimum Cut Theorem (3.3)	**Teaching Time** 4 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- **be able to find a cut equal in value to that of the maximum flow;**
- **be able to confirm that a flow is maximal using the maximum flow-minimum cut theorem.**

TEACHING POINTS

Students should be familiar with finding cuts as small as possible – it is advisable to look at the position of saturated arcs to help to find a cut which is equivalent in value to the maximal flow. The minimum cut passes through saturated arcs going through the direction from source to sink, and empty arcs going from sink to source. It is useful to consider practical applications of network flows to consider why the maximum flow – minimum cut theorem is valid.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Practical scenarios are interesting to consider – for example, could students model a local road network and estimate the number of cars that could pass along each road in a minute, to evaluate the effectiveness of the local road network. An extension to this would be to consider the effect of introducing a one-way system.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

Many candidates can attempt to find a cut, but often not equal to the maximum flow. When asked to confirm that a flow is maximal, students must quote the ‘maximum flow – minimum cut’ theorem. It is also advisable for students to draw the cut on the diagram showing their maximal flow pattern rather than just stating the arcs that the cut passes through in order to make their working clear. Those that quote the theorem without finding a cut will gain no marks, as they are not actually using the theorem. Students should be reminded to refer to the original diagram containing the flow capacities, when considering possible cuts, rather than their optimal solution.

3d. Multiple sources and sinks (3.4)	**Teaching Time** 3 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- be able to adapt the model to deal with networks with multiple sources and/or sinks;
- be able to adapt the model to deal with vertices with restricted capacity.

TEACHING POINTS

Students need to be able to add a supersource and supersink to a network with multiple sources/sinks, before using the labelling procedure as before to find a maximum flow through a network.

If a network has several sources S_1 , S_2 , . . ., then these can be connected to a single **supersource** S. The capacity of the edge joining S to S_1 is the sum of the capacities of the edges leaving S_1.

If a network has several sinks T_1 , T_2 , . . ., then these can be connected to a **supersink** T. The capacity of the edge joining T_1 to T is the sum of the capacities of the edges entering T_1.

If a vertex has a **restricted capacity** then this can be replaced by two unrestricted vertices connected by an edge of the relevant capacity.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Present students with the scenario of multiple sources and sinks, and see if they can develop the idea of supersource and supersink, including the restrictions upon the arcs joining these.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

When directed, students can comfortably add a supersource and supersink. However, often, students do not carry this through to the rest of the question, meaning that a lot of subsequent marks are lost.

NOTES

Problems may include vertices with restricted capacity.

3e. Optimal flow rates (3.5)	**Teaching Time** 2 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- be able to calculate upper and lower capacities through a network;
- be able to determine the optimal flow rate in a network subject to given constraints.

TEACHING POINTS

If a problem contains both **upper** and **lower capacities** then the capacity of the cut is the sum of the upper capacities for arcs that cross the cut in the direction from *S* to *T* minus the sum of the lower capacities for arcs that cross the cut in the direction from *T* to *S*.

The maximum flow – minimum cut theorem can therefore be adapted to verify the value of a minimum cut.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Logical deduction can be used to establish the minimum and maximum flow through a network by considering the upper and lower capacities of each arc.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

Students can forget to take in to account the lower capacities of the arcs going from *T* to *S* when calculating a cut. Students who do not fully understand problems with upper and lower capacities just ignore the lower capacities and only deal with the upper.

NOTES

Problems may include both upper and lower capacities.

UNIT 4: Dynamic Programming

Return to Overview

SPECIFICATION REFERENCES

4.1 Principles of dynamic programming. Bellman's principle of optimality.

Stage variables and State variables. Use of tabulation to solve maximum, minimum, minimax or maximin problems.

PRIOR KNOWLEDGE

A Level Further Mathematics - Decision Mathematics 1

1 Introduction to algorithms (See SoW Unit 1a)

1 Introduction to graph theory (See SoW Unit 1c)

2.2 Dijkstra's algorithm (See SoW Unit 2b)

KEYWORDS

Bellman's principle of optimality, stage, state, action, destination, value, minimax, maximin

4a. Network and table form for dynamic programming (4.1)	**Teaching Time** 9 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- understand the terminology and principles of dynamic programming;
- understand Bellman's principle of optimality;
- understand and use stage variables and state variables;
- be able to use dynamic programming to solve maximum and minimum problems, presented in network or table form;
- be able to use dynamic programming to solve minimax and maximin problems, presented in network or table form.

TEACHING POINTS

Bellman's principle for dynamic programming is 'Any part of an optimal path is optimal.' Dynamic programming is a versatile technique for solving multistage decision making problems.

Students must understand that the basic principle of dynamic programming is that you work backwards through a network, or problem presented in tabular form. Students should be familiar with both forms, being able to model a situation using either form. They should then be able to interpret their solution in the practical context.

Stage – a move along the path from the initial vertex to the final vertex

State – the vertex being considered

Action – a directed arc from one stage to the next

Destination – the arrival vertex having taken an action

Value – the sum of the weights on the arcs used in a sequence of actions

If presenting the solution in tabular form, students should be encouraged to use the headings of

"stage – state – action – destination – value"

Optimal values at each stage are denoted by * and it is these which should be carried through to the next stage (a key part of the dynamic programming procedure).

Dynamic programming can be used for a wide variety of problems, including to find a shortest path, so it is useful to draw parallels with this topic from Decision Mathematics 1.

The **minimax route** is the one in which the maximum value of the individual arcs used is as small as possible.

The **maximin route** is the one in which the minimum value of the individual arcs used is as large as possible.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Many practical problems can be solved using dynamic programming, and one of the hardest things students find is how to model a practical situation correctly using dynamic programming. Students should therefore have experienced a wide variety of practical problems – for example, shortest path, allocation, production planning, stock control, investment planning. Any multistage problems which aim to optimise time, profit, cost or resources could be considered. Students should also be encouraged to construct a diagram where appropriate to visualise each stage.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

This is a question that students either do very well or very badly on. If students are able to show a clear grasp of how to use the given information to work backwards through their table, from one stage to the next using the correct relevant values at each stage, they usually obtain a completely correct final solution. Some students will make errors when choosing the correct elements to include in calculations or in the arithmetic, and a few will forget to carry forward previous optimal values but most will continue correctly. Many students cross working out and then attempt to squeeze in alternative answers, making it very difficult for examiners to actually mark their work.

Sometimes extra rows are seen but a significant number of students will lose marks if they are deleting or omitting a state. If students are doing this, it is because they seem to be considering the demand at an earlier time, and concluding it would not be possible. In essence these students are working forwards and not backwards; this is a common error when applying the principles of dynamic programming. Students should be advised that in Decision Mathematics they must rigorously apply the algorithm, rather than introduce their own logic or common sense. A small minority of students will start to apply the algorithm going forwards, and would score no marks as they are not using the principles of dynamic programming.

NOTES

Students should be aware that any part of the shortest/longest path from source to sink is itself a shortest/longest path, that is, any part of an optimal path is itself optimal.

Both network and table formats are required.

UNIT 5: Game Theory

Return to Overview

SPECIFICATION REFERENCES

5.1 **Two person zero-sum games and the pay-off matrix.**

5.2 **Identification of play safe strategies and stable solutions (saddle points).**

5.3 Reduction of pay-off matrices using dominance arguments.

5.4 **Optimal mixed strategies for a game with no stable solution by use of graphical methods for $2 \times n$ or $n \times 2$ games where $n = 1, 2, 3$ or 4**

5.5 Optimal mixed strategies for a game with no stable solution by conversion of higher order games to linear programming problems that can then be solved by the Simplex algorithm.

PRIOR KNOWLEDGE

A Level Further Mathematics - Decision Mathematics 1

1 Introduction to algorithms (See SoW Unit 1a)

5 Linear Programming (See SoW Unit 4)

KEYWORDS

Two- person, zero-sum, pay-off matrix, play safe, stable solution (saddle point), equilibrium, mixed strategy, value of the game, dominance, Simplex

5a. Two-person zero-sum games (5.1)	**Teaching Time** 2 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- **understand what is meant by a two-person game;**
- **understand and use pay-off matrices;**
- **understand the meaning of a zero-sum game.**

TEACHING POINTS

A **two-person game** is one in which only two parties can play.

A **zero-sum** game is one in which the sum of the losses for one player is equal to the sum of the gains for the other player.

Each element *Cij* of a pay-off matrix shows the gain to the row player (player A) when they play row *i* and the column player (player B) plays column *j*.

Students should be able to change a pay-off matrix from player A's point of view to a pay-off matrix from player B's point of view, by transposing the matrix and multiplying each term by −1.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

The Prisoner's Dilemma is a famous classic example of game theory which is a good way to introduce this topic.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

When converting a pay-off matrix to the other players' point of view, many students either multiply by −1 or transpose, but not both as they should do.

NOTES

A pay-off matrix will always be written from the row player's point of view unless directed otherwise.

5b. Play-safe strategies (5.2)	**Teaching Time** 3 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- **understand what is meant by play safe strategies;**
- **be able to determine the play safe strategy for each player.**

TEACHING POINTS

To consider what “play safe” means, suppose each player works out what is the worst that could happen for each choice in turn. He then selects the choice which represents the least worst choice.

To determine the play-safe strategy for player A, find the row maximin, and for player B, the column minimax.

If the row maximin = column minimax, then the game is said to be in equilibrium – there is a stable solution/saddle point.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Consider a pay-off matrix – what entries could be changed so that there is a stable solution? What happens to a stable solution if one entry in the matrix is changed?

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

Students can often correctly find row minimums and column maximums and then deduce that the row maximin was not equal to the column minimax, without clearly identifying these values. A few students do calculate row maximums and column minimums instead of the correct values.

NOTES

Students should be aware that in a zero-sum game there will be a stable solution if and only if the row maximin = the column minimax.

The proof of the stable solution theorem is not required.

5c. Dominance (5.3)	**Teaching Time** 1 Hour

OBJECTIVES

By the end of the sub-unit, students should:

- be able to reduce a pay-off matrix using dominance arguments.

TEACHING POINTS

If one row X (or column) is always a better option than another row Y (or column), then row X (or column) is said to dominate row Y (or column). This means that the player would always choose the play the dominant X row (or column) in preference to the other, so the Y row (or column) can be deleted from the pay-off matrix.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Consider what the values in a row or column would need to be for it to be dominated and therefore eliminated.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

When students are using the dominance argument, it is common to see the wrong column/row eliminated.

5d. Optimal mixed strategies using graphical methods (5.4)	Teaching Time 6 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- **be able to determine optimal mixed strategies for a game with no stable solution by use of graphical methods for $2 \times n$ or $n \times 2$ games where $n = 1, 2, 3$ or 4.**

TEACHING POINTS

Students should always check to see if there is a stable solution. If not, students should be able to determine a mixed strategy.

For games that are, or can be reduced to, $2 \times n$ or $n \times 2$ games, a graphical method can be used. The expected winnings for each player can be calculated, by forming probability equations (use "p" and "$1 - p$" for the player that has a choice of two strategies). To find the optimal strategy, the probability equations can be graphed to find the intersection that gives the highest minimum point (if considering from player A's perspective), and then solved algebraically.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Students should carefully consider whether they are solving the problem from the perspective of player A or player B, and how this affects the graphical approach.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

When using the graphical approach, candidates will score no marks if they do not interact with the graph, and instead just solve 3 pairs of simultaneous equations. Students should draw their probability graph and use this to select the relevant pair of equations to solve algebraically. Many graphs:

- are drawn poorly without rulers,
- go beyond the axes at $p = 0$ and $p = 1$,
- have uneven or missing scales on the vertical axes,
- are cramped so that it is difficult to identify the correct optimum point.

When a game needs to be modified to consider the game from the point of view of the other player, students will usually attempt to modify their matrix but may either just change signs, or transpose, rather than correctly doing both.

5e. Optimal mixed strategies using the Simplex algorithm (5.5)	Teaching Time 4 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- be able to determine optimal mixed strategies for a higher order game with no stable solution by converting the game to a linear programming problem that can be solved by the Simplex algorithm.

TEACHING POINTS

Students should always check to see if there is a stable solution. If not, students should be able to determine a mixed strategy.

For games that cannot be reduced to two strategies for one player, e.g. a 3×3 game, an algebraic method must be used. The game should be formulated as a linear programming problem and then the Simplex method can be used. First, all elements in the pay-off matrix are made positive by adding n to each element, so that the feasible region is in the positive quadrant. Then the usual steps for formulating a linear programming problem are used:

- defining decision variables
- writing the objective function
- writing all constraints.

Constraints are written as equations with slack variables introduced so that the information can be put in to a Simplex tableau. This is then solved using the methods learnt in Decision Mathematics 1.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

A detailed discussion should be had to ensure that students understand how a problem can be translated in to constraints and then solved using the simplex algorithm.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

When setting up a 3×3 game for solution by simplex many students forget to make all values in the matrix positive before writing down their constraints.

UNIT 6: Recurrence Relations

Return to Overview

SPECIFICATION REFERENCES

6.1 **Use of recurrence relations to model appropriate problems.**

6.2 **Solution of first** and second **order linear homogeneous and non-homogeneous recurrence relations.**

PRIOR KNOWLEDGE

A Level Mathematics - Pure content

2.10 Partial fractions (See SoW Remaining A level content Unit 2b)

A Level Further Mathematics – Core Pure content

2 Complex numbers (See SoW Remaining A level content Unit 1)

GCSE (9-1) in Mathematics at Higher Tier

A23, A24, A25 Sequences

A18 Solve quadratic equations

KEYWORDS

Recurrence relation, iteration, first-order, second-order, linear, homogeneous, non-homogeneous, general solution, particular solution, complementary function, auxiliary equation.

6a. Modelling using recurrence relations (6.1)	**Teaching Time** 3 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- **be able to detect problems in context which could be modelled with a recurrence relation;**
- **recognise and describe associated sequences using a recurrence relation;**
- **be able to classify a recurrence relation by its order.**

TEACHING POINTS

A recurrence relation is an equation of the form:

$$f(n) = u_n + a\,u_{n-1} + b\,u_{n-2} + \cdots$$

The order of a recurrence relation is the difference between the highest and lowest subscript in the equation.

Students should be familiar with modelling sequences from GCSE, so a good introduction to this topic is to start with the sequences that they do know, and then see if they can also be expressed as a recurrence relation.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

There are lots of opportunities here to model real-life applications of recurrence relations, for example with the Fibonnaci sequence, population growth, mortgage repayments or similar economic situations.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

This topic has not been on the previous Pearson A Level Specification, so no examiner report comments exist.

6b. Solving first order recurrence relations (6.2)	**Teaching Time** 6 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- **be able to solve a first order recurrence relation;**
- **be able to apply solutions to first order recurrence relations to problems in context.**

TEACHING POINTS

First order linear homogeneous recurrence relations with constant coefficients are of the form $u_{n+1} = k\,u_n$ with k constant and $n \geq 1$, and the general solution is of the form $u_n = k^{n-1}\,u_1$.

First order linear non-homogeneous recurrence relations are of the form $u_{n+1} = k\,u_n + c$ and for k and c constant, $k \neq 1$, the general solution is $u_n = k^{n-1}\,u_1 + \frac{c(k^{n-1}-1)}{k-1}$.

If a first order linear non-homogeneous recurrence relation has $k = 1$, then we have $u_{n+1} = u_n + c$ which has general solution $u_n = u_1 + (n-1)c$.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

There are lots of opportunities here to investigate real-life applications of recurrence relations, for example with the Fibonnaci sequence, population growth, mortgage repayments or similar economic situations.

It can be interesting to investigate the values of k for which $u_{n+1} = k\,u_n + c$ converges.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

This topic has not been on the previous Pearson A Level Specification, so no examiner report comments exist.

Students should be able to deal with alterations in notation, for example starting a sequence with u_0 rather than u_1.

NOTES

Students should be able to solve relations such as,

- $u_{n+1} - 5u_n = 8,\ u_1 = 1$

Students must be able to use recurrence relations to model real life applications e.g. population growth.

6c. Solving second order recurrence relations (6.2)	**Teaching Time** 5 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- be able to obtain the solution of any linear homogeneous second order recurrence relation;
- be able to use generating functions to solve non-homogeneous second order recurrence relations;
- be able to apply solutions to second order recurrence relations to problems in context.

TEACHING POINTS

Second order linear homogeneous recurrence relations are of the form $au_{n+2} + b\,u_{n+1} + c\,u_n = 0$, with *a*, *b* and *c* constant and $n \geq 1$, and the general solution is found by solving the associated auxiliary equation $aX^2 + bX + c = 0$.

If X_1 and X_2 are solutions of the auxiliary equation, then the general solution is of the form

$u_n = A\,X_1{}^n + B\,X_2{}^n$ if X_1 and X_2 are distinct

or $u_n = A\,X_1{}^n + Bn\,X_1{}^n$ if $X_1 = X_2$.

Second order linear non-homogeneous recurrence relations are of the form $f(n) = au_{n+2} + b\,u_{n+1} + c\,u_n$, and the general solution is $u_n =$ complementary function + particular solution.

The complementary function is the general solution of the associated homogeneous equation $au_{n+2} + b\,u_{n+1} + c\,u_n = 0$, and the usual form of the particular solution is similar to the form of *f(n)*.

Students should be familiar with the use of generating functions to obtain solutions.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

There are lots of opportunities here to investigate real-life applications of recurrence relations, for example with the Fibonnaci sequence, population growth, mortgage repayments or similar economic situations.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

This topic has not been on the previous Pearson A Level Specification, so no examiner report comments exist.

NOTES

Students should be able to solve relations such as,

- $2u_{n+2} + 7u_{n+1} - 15u_n = 6$

  $u_1 = 10,\quad u_2 = -17$

The terms, particular solution, complementary function and auxiliary equation should be known.

UNIT 7: Decision Analysis

Return to Overview

SPECIFICATION REFERENCES

7.1 Use, construct and interpret simple decision trees.

7.2 Use of expected monetary values (EMVs) and utility to compare alternative courses of action.

PRIOR KNOWLEDGE

GCSE (9-1) in Mathematics at Higher Tier

P1-8 Probability

P9 Tree diagrams

KEYWORDS

Decision tree, expected monetary value (EMV), chance node, decision node, end pay-off

7a. Decision trees (7.1)	**Teaching Time** 5 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- be able to construct a decision tree to analyse a more complicated decision situation;
- understand that decision trees have three types of node: end pay-offs, chance nodes and decision nodes

TEACHING POINTS

Decision analysis is a procedure that can be used to help make decisions as to whether or not to do something, where there is a mixture of decision making and chance element that are involved in the decision. In game theory, it gives us the opportunity to decide whether or not to go ahead and play or not.

A decision tree is used to summarise all of these choices/chance outcomes. It is conventional to use triangles for end pay-offs, circles for chance nodes and rectangles for decision nodes. Decision trees are constructed from left to right with values shown at end pay-offs.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Decision analysis is often used in business scenarios to consider whether to embark upon a new venture, so parallels could be made here with work students may be covering in Business/Economics, or to consider local business examples.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

This topic has not been on the previous Pearson A Level Specification, so no examiner report comments exist.

However, it is worth noting that when asked to draw a decision tree, it is crucial to correctly establish the order of decisions and chances. Some students omit a stage which includes decision nodes, meaning that it is impossible to accurately process the EMV.

NOTES

Students should be familiar with the terms decision nodes, chance nodes and end pay-offs.

7b. Expected monetary values (EMVs) (7.2)	**Teaching Time** 3 Hours

OBJECTIVES

By the end of the sub-unit, students should:

- be able to calculate the average profit per game in a simple game of chance;
- be able to use the EMV algorithm to evaluate the EMV from a decision tree.

TEACHING POINTS

The EMV algorithm makes decisions to maximise the EMV at each stage. To analyse a decision tree, we work from right to left calculating the EMV at each chance node. This enables us to decide which strategy would maximise our expected monetary value.

OPPORTUNITIES FOR REASONING/PROBLEM SOLVING

Students can consider a variety of game situations and try to model them using decision trees.

COMMON MISCONCEPTIONS/EXAMINER REPORT QUOTES

This topic has not been on the previous Pearson A Level Specification, so no examiner report comments exist.

NOTES

Students should be able to use the decision tree to identify the optimal strategy for a game and state the associated optimal EMV.

For more information on Edexcel and BTEC qualifications please visit our websites:
www.edexcel.com and www.btec.co.uk

Pearson Education Limited. Registered in England and Wales No. 872828
Registered Office: 80 Strand, London WC2R 0RL.
VAT Reg No GB 278 537121