Artificial Intelligence Methods (G52AIM)
Examination – 2008-2009

Rubric

Answer any 4 from 6. Only the four questions indicated on the front of the script will be marked.
Scientific calculators are allowed.

Artificial Intelligence Methods (G52AIM)
Examination – 2008-2009

Question 1

a) With reference to Ant Colony Optimisation (ACO) describe these algorithmic elements.

- Evaporation

(2 marks)

- Visibility

(5 marks)

- Transition Probability

(6 marks)

b) Assume that the original ACO as presented by Dorigo is being used to solve the Travelling Salesman Problem. Discuss the changes that would have to be made to the algorithm to convert it to an ant algorithm based hyper-heuristic.

(12 marks)

Question 2

a) Show, in pseudo code, a simple genetic algorithm with a brief description of each of the main elements.

(12 marks)

b) Explain the difference between a genotypic representation and a phenotypic representation. Give an example of each.

(8 marks)

c) Using an example, show why it is important to have a mutation operator in a genetic algorithm.

(5 marks)

Artificial Intelligence Methods (G52AIM)
Examination – 2008-2009

Question 3

a) Describe the idea behind the simulated annealing algorithm making reference to its origins as an optimisation methodology.

(7 marks)

b)

i) Show a simulated annealing algorithm.

(9 marks)

ii) Indicate how you could change the algorithm so that it implements hill climbing.

(4 marks)

c) With regards to simulated annealing, what is the probability of accepting the following moves? Assume the problem is trying to maximise the objective function.

Current Evaluation	Neighbourhood Evaluation	Current Temperature
16	15	20
25	13	25
76	75	276
1256	1378	100

(5 marks)

Question 4

a) How could you cope with division by zero in a program being evolved by a genetic programming approach?

(2 marks)

b) According to John Koza there are five stages when planning to solve a problem using genetic programming. What are they? Give a short description of each.

(12 marks)

c) Several problems were presented in the lectures (in the video by John Koza). Suggest *another* problem that could be solved by genetic programming. Give your answer with reference to part b.

(11 marks)

Artificial Intelligence Methods (G52AIM)
Examination – 2008-2009

Question 5

a) The orthogonal rectangle packing problem requires that rectangles are packed into a bin with the aim of minimising the height of the packing.

i) Suggest two representations for this problem when using a genetic algorithm.

(2 marks)

ii) Present the advantage(s) and disadvantage(s) of each type of representation.

(6 marks)

b) In the Travelling Salesman Problem we could represent the problem as a sequence of cities, 1..*n*.

i) Using a genetic algorithm with 1-point crossover we could (and almost definitely would) get infeasible solutions. Assuming we could not change the crossover operator, how could we overcome this problem?

(4 marks)

ii) If we were able to use a different crossover operator, which one would you suggest? Give an example.

(6 marks)

c) If we were using simulated annealing to solve the Travelling Salesman Problem, suggest a neighbourhood function.

(7 marks)

Artificial Intelligence Methods (G52AIM)
Examination – 2008-2009

Question 6

a) In the lectures we considered a case study of minimising the distances travelled by football clubs over the Christmas/New Year period.

i) If we were to extend this problem to schedule all the fixtures for the entire season, what changes would have to be made to the model that was presented in the lectures? You do not have to present a formal model. A bullet point list of changes will suffice.

(10 marks)

b)

i) Describe a problem domain of your choosing, that could be presented as a search problem. Additional marks will be awarded for describing a problem that has not been covered in the course.

(6 marks)

ii) Define the hard and soft constraints for your problem. These do not have to be presented in a formal way.

(4 marks)

iii) Briefly describe what solution methodology you would choose to solve this problem, explaining why.

(5 marks)

Artificial Intelligence Methods (G52AIM)
Examination – 2008-2009

Question 1: Answer

a) With reference to Ant Colony Optimisation (ACO) describe these algorithmic elements

At the start of the ACO algorithm one ant is placed in each city (assuming we are modeling the TSP). Time, *t*, is discrete. *t*(0) marks the start of the algorithm. At *t*+1 every ant will have moved to a new city and the parameters controlling the algorithm will have been updated. Assuming that the TSP is being represented as a fully connected graph, each edge has an *intensity of trail* on it. This represents the pheromone trail laid by the ants. Let $T_{i,j}(t)$ represent the intensity of trail edge (i,j) at time *t*. When an ant decides which town to move to next, it does so with a probability (Transition Probability) that is based on the distance to that city and the amount of trail intensity on the connecting edge. The distance to the next town, known as the *visibility*, n_{ij}, is defined as $1/d_{ij}$, where, d_{ij}, is the distance between cities *i* and *j*. The distances are symmetric (i.e. $d_{ij} = d_{ji}$)

At each time unit *evaporation* takes place. This is to stop the intensity trails building up unbounded. The amount of evaporation, *p*, is a value between 0 and 1.

I am not looking for a description as the one shown above. It is provided for the internal/external examiners so that they have an understanding of the algorithm

i) Evaporation

(2 marks)

As stated above *evaporation* is simply a mechanism to stop pheromone levels building unbounded. The amount of evaporation, *p*, is a value between 0 and 1.

It is used in the pheromone updating formula. That is (where *p* is the evaporation term):

$$T_{ij}(t + n) = p \,.\, T_{ij}(t) + \Delta T_{ij}$$

2 marks, pro-rata.

ii) Visibility

(5 marks)

Visibility is, essentially a heuristic. In the TSP it is defined as the distance between one city and the next. That is, an ant should prefer a shorter distance but if this is done all the time, we simply get a greedy algorithm.

Visibility is used in the transition probability formula. That is:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[T_{ij}(t)]^{\alpha} \,.\, [n_{ij}]^{\beta}}{\sum k \in allowed_k \quad [T_{ik}(t)]^{\alpha} \,.\, [n_{ik}]^{\beta}} & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases}$$

The term for visibility is n_{ij}.

3 marks for describing visibility and 2 marks for the formula.

iii) Transition Probability

(6 marks)

The transition probability formula is shown above. It is used for an ant to decide which city to visit next (assuming we are modelling the TSP). It is a combination of visibility and the current pheromone levels (each having a weight). The values are normalised over all the allowed values (in the case of the TSP this is the cities that have not yet been visited).

3 marks for describing what it is, 3 marks for the formula (even if they have produced the formula in part ii – thus they can get a total of 5 marks for reproducing the formula).

b) Assume that the original ACO as presented by Dorigo is being used to solve the Travelling Salesman Problem. Discuss the changes that would have to be made to the algorithm to convert it to an ant algorithm based hyper-heuristic.

(12 marks)

In the lectures we presented ant algorithms and hyper-heuristics and in each lecture we made reference to a hyper-heuristic version, without spending a lot of time as to how this algorithm would operate. For the students to answer this question well they will have to have thought about the two algorithms and how the conversion could be done. It should also be noted that this is a current research area so this material is not covered in the course textbooks.

I am looking for the following observations.

- The nodes on the graph now represent heuristics (rather than cities), so that we are now looking for a good sequence of heuristics rather than a TSP tour (**2 marks**).
- Due to the problem, we can revisit the same node (as we can apply the same heuristic more than once in a tour), so we no longer need have the concept of tabu any more (**1 mark**).
- The concept of pheromone and evaporation remain (**2 marks**).
- The distance matrix (i.e. visibility) is replaced by the concept of *how well the heuristics work together, with respect to time*. That is, if a given heuristic is *fast* we are more likely to visit that heuristic as we will not have wasted so much time if the heuristic shows itself to decrease the quality of the solution (**3 marks**).
- I would like the answer to say that there would actually be limited changes to the algorithm (distance table changes to a dynamic heuristic measure, the ability to re-visit heuristics and the fact when we visit a node, we execute that heuristic (**2 marks**).
- **2 marks** will also be used at my discretion (e.g. demonstration of reading around the subject)

Artificial Intelligence Methods (G52AIM)
Examination – 2008-2009

Question 2: Answer

a) Show, in pseudo code, a simple genetic algorithm with a brief description of each of the main elements.

(12 marks)

1. Initialise population
   - *Could be done randomly, using constructive heuristics, choosing best known solutions etc.*

(1 mark)

2. Repeat

   2.1. Evaluate each member in the population
   - *Evaluates how well each member of the population solves the problem at hand. This could be the only domain knowledge we need*

(2 marks)

   2.2. Selection
   - *Selects which chromosomes to breed. Possible methods include roulette wheel and tournament*

(2 marks)

   2.3. Crossover
   - *Crossover the selected chromosomes using a suitable crossover operator (examples are given earlier in the question (1-point, 2-point, uniform, partially matched etc.)*
   - *Crossover probability is usually about 0.8. The other 0.2 the children will be the same as the parents.*

(2 Marks)

   2.4. Mutation
   - *Mutate each of the new chromosomes in order to add diversity to the population.*
   - *Mutation is usually carried out with low probability*

(2 Marks)

   2.5. Replacement
   - *Using a suitable replacement policy, ensure that the size of the population is maintained*

(2 Marks)

3. Until termination
   - *Termination could be based on the number of generations, time available, a suitable solution has been found etc.*

(1 Mark)

b) Explain the difference between a genotypic representation and a phenotypic representation. Give an example of each.

(8 marks)

This was not covered in the lectures (although was mentioned in passing) but for the student to give a good answer they would have had to read around the subject.

A genotypic representation is one where the underlying representation has to be decoded in order to determine the behaviour (or produce an evaluation function).

(2 marks)

An example could be a bit string, which represents a set of real numbers with the binary digits having to be decoded in order to determine the real numbers, which can then be evaluated by carrying out the relevant function.

(2 marks)

The underlying representation for a phenotypic representation represents the problem in a natural way. The behaviour is expressed directly into the representation.

(2 marks)

An example, could be a robot arm, where the coding is directly related to the movement of the arm. For example, move right, turn 21°, move up etc. etc.

(2 marks)

c) Using an example, show why it is important to have a mutation operator in a genetic algorithm

(5 marks)

Mutation is important in order to add diversity to the population
(**1 mark**)

The worst case scenario is that a certain bit position (assuming a bit representation) is the same in every parent. This bit, using an operator such as one point crossover, can never be changed.

For example, given two parents (which we will assume are the entire population)

P_1	0	0	1	0	0	0	1	0	1	1
P_2	1	1	0	1	0	1	0	1	0	0

The shaded bits can never get set to one unless we allow mutation.
(2 marks)

In the lectures I showed an example of a very simple function where we could not find the optimal unless we had a mutation operator. I will award 2 marks if they make reference to the example I gave in the lecture (or something similar).
(2 marks)

Artificial Intelligence Methods (G52AIM)
Examination – 2008-2009

Question 3: Answer

a) Describe the idea behind the simulated annealing algorithm making reference to its origins as an optimisation methodology.

(7 marks)

Simulated Annealing (SA) is motivated by an analogy to annealing in solids. The idea of SA comes from a paper published by Metropolis etc al in 1953 (Metropolis, 1953). The algorithm in this paper simulated the cooling of material in a heat bath. This is a process known as annealing.
If you heat a solid past melting point and then cool it, the structural properties of the solid depend on the rate of cooling. If the liquid is cooled slowly enough, large crystals are formed. However, if the liquid is cooled quickly (quenched) the crystals will contain imperfections.
Metropolis's algorithm simulated the material as a system of particles. The algorithm simulates the cooling process by gradually lowering the temperature of the system until it converges to a steady, *frozen* state.
In 1982, Kirkpatrick et al (Kirkpatrick, 1983) took the idea of the Metropolis algorithm and applied it to optimisation problems. The idea is to use simulated annealing to search for feasible solutions and converge to an optimal solution.

Marks will be awarded pro-rata but full marks will only be awarded if the answer contains reference to Metropolis 1952 and Kirkpatrick 1983.

b)

i) Show a simulated annealing algorithm.

(9 marks)

I will award marks (pro-rata) if the algorithm is correct. The ones presented in the textbooks, in my view, are overly complex and the algorithm can be presented a lot more concisely (and I would accept a more concise algorithm as long as I can follow it). However, I did present a number of algorithmic example in the lectures. These two were taken from two of the course textbooks.

From Artificial Intelligence: A Modern Approach

Function SIMULATED-ANNEALING(*Problem*, *Schedule*) **returns** a solution state
    Inputs : *Problem*, a problem
    Schedule, a mapping from time to temperature
    Local Variables : *Current*, a node
    Next, a node
    T, a "temperature" controlling the probability of downward steps
Current = MAKE-NODE(INITIAL-STATE[*Problem*])
For t = 1 **to** ∞ **do**
    T = *Schedule*[t] // @@@@ //
    If T = 0 **then return** *Current* // @@@@ //
    Next = a randomly selected successor of *Current*
    ΛE = VALUE[*Next*] – VALUE[*Current*] **if** ΛE > 0 **then** *Current* = *Next* // **** //
    else *Current* = *Next* only with probability exp(ΛE/T)

This one from the main course textbook

procedure SIMULATED ANNEALING;
begin
    INITIALIZE (*istart* , *c0, L0*);
    k := 0;

```
    i := i_start ;
    repeat
        for l := 1 to L_k do
        begin
            GENERATE ( j from S_i );
            if f ( j ) ≤ f (i ) then i := j
            else
            if exp   ( f (i)− f ( j )) / ck > random[0, 1) then i := j // **** //
        end;
        k := k + 1;
        CALCULATE LENGTH (L_k);
        CALCULATE CONTROL (c_k); // @@@@ //
    until stopcriterion
end;
```

ii) Indicate how you could change the algorithm so that it implements hill climbing.

(4 marks)

There are two ways to convert this algorithm to a hill climbing algorithm.

1) Only accept improving moves (indicated in red in the two algorithms above, and also indicated by the lines marked '// **** //').
   2 marks awarded (1 for stating it, the other for indicating where this is defined in their algorithm).
2) Setting the temperature of the system to zero (indicated in green in the algorithm, and also indicated by '// @@@@ //'; although the algorithms do not make this really clear). This has the effect of NEVER accepting worse moves.
   2 marks awarded (1 for stating it, the other for indicating how where this is defined in their algorithm).

c) With regards to simulated annealing, what is the probability of accepting the following moves? Assume the problem is trying to maximise the objective function.

Current Evaluation (a)	Neighbourhood Evaluation (b)	Current Temperature (c)	Probability
16	15	20	0.95122942
25	13	25	0.61878339
76	75	276	0.99638337
1256	1378	100	3.38718773[1]

[1] This is the value returned from the SA accept function. The probability of accepting the move is 1. Either answer is correct.

(5 marks)

The probability is given by exp((b-a)/c).

1 mark for each one that is correct and an **extra mark** for saying that the last row is an improving move so will always be accepted, either due to simulated annealing always accepting improving moves or because the acceptance function returns a value > 1.

Artificial Intelligence Methods (G52AIM)
Examination – 2008-2009

Question 4: Answer

a) How could you cope with division by zero in a program being evolved by a genetic programming approach?

It is usual to use a protected division by zero which returns 1 if a division by zero is attempted.

2 marks

b) According to John Koza there are five stages when planning to solve a problem using genetic programming. What are they? Give a short description of each.

1) Identify the terminal set. That is the “symbols” that can appear at the leaves of the parse tree. For example X, Y, Z (i.e. variables).

1 mark for stating, 1 mark for an example

2) Identify the function set. That is the “symbols” that can appear at the inner nodes of the parse tree. For example +, /, *, -, cos, sin, etc..

1 mark for stating, 2 marks for an example

3) Identify the fitness measure that is used to evaluate a given evolved program.
For example, how well does the evolved program fit a given function? Given, say, X^3 in the range (-10..+10) how far is each point in the evolved program away from the same point in X^3? The sum of these errors is the evaluation of the program, with the aim being to minimise that error.

1 mark for stating, 2 marks for an example

4) What are the control parameters of the algorithm? For example, what is the population size and how many iterations will be carried out?

1 mark for stating, 1 mark for an example

5) What is the terminating condition and what designates the result? For example, do we run for a given number of iterations or do we run for a certain amount of time. For the result designation, do we take the best program found in all the runs or do we take the best program from the final population?

1 mark for stating, 1 mark for an example

c) Several problems were presented in the lectures (in the video by John Koza). Suggest *another* problem that could be solved by genetic programming. Give your answer with reference to part b.

(10 marks)

In the video the problems that were described included regression, backing up a lorry, ant trail (food foraging), classification (points on interlocking spirals), robots pushing blocks, evolving wall following behaviour etc.

As long as the students do not simply replicate any of these problems (but I will accept variations as long as they vary enough), I will accept their problem domain. I will then award marks as follows:

Problem Description: **3 marks**

Identify the terminal set: **2 marks**

Identify the function set: **2 marks**

Identify the fitness measure: **1 mark**

Control parameters of the algorithm: **1 mark**

The terminating condition and what designates the result: **2 marks**

Question 5: Answer

This question covers a variety of topics focussed loosely on representation issues. It allows the students to pick up marks on a variety of problem domains and AI search methods.

a) The orthogonal rectangle packing problem requires that rectangles are packed into a bin with the aim of minimising the height of the packing.

i) Suggest two representations for this problem when using a genetic algorithm

(2 marks)

This problem was briefly discussed in the lectures and the students were also provided with a paper which discusses this problem.

Represent each rectangle as *x*,*y* coordinates which represent two corners of the rectangle.
1 mark

Represent each rectangle as an integer (or alpha character).
1 mark

ii) Present the advantage(s) and disadvantage(s) of each type of representation.

(6 marks)

***x*,*y* coordinates representation**

Advantage: It is the natural choice.
(1 mark)

Disadvantage: The representation does not lend itself to detecting overlaps.
If using a search method such as genetic algorithms standard crossover might not be applicable and custom operators would be required.
(2 marks)

Representing each rectangle as an integer (or alpha character).

Advantage: Simple representation.
We can avoid overlaps by the use of a heuristic.
(1mark)

Disadvantage: We need to provide a heuristic (such as Bottom Left Fill).
Operators, such as crossover, are easier and we could even use standard ones such as 1-point crossover.
(2 marks)

Other advantages/disadvantages are also possible, as described in the paper.

b) In the Travelling Salesman Problem we could represent the problem as a sequence of cities, 1..*n*.

i) Using a genetic algorithm with 1-point crossover we could (and almost definitely would) get infeasible solutions. Assuming we could not change the crossover operator, how could we overcome this problem?

(4 marks)

I would expect the student to give an answer saying that we would have to *repair* the solution in some way (**1 mark**). If they give an example of a chromosome that has duplicated entries and show a heuristic (e.g. identify missing entries, identify duplicate entries and carry out a simple replacement) I will award the other marks (**3 marks**).
The students could give an alternative answer, which I will judge on its merits.

ii) If we were able to use a different crossover operator, which one would you suggest? Give an example.

(6 marks)

The answer I would expect is Partially Matched Crossover (**1 mark**).

An example is shown below but, of course, the answer received will be different but the students have to demonstrate that they know how to use the crossover operator (**5 marks**).

For a genetic algorithm, we are given the following parents, P_1 and P_2, and the template T

P_1	A	B	C	D	E	F	G	H	I	J
P_2	B	D	J	H	C	E	I	A	G	F
T	1	0	1	1	0	0	0	1	0	1

Assume that C_n, are crossover points where $C_n = 0$ means that the crossover point is at the extreme left of the parent.

Partially Matched Crossover (using $C_2 = 2$ and $C_3 = 4$)

P_1	A	B	C	D	E	F	G	H	I	J
P_2	B	D	J	H	C	E	I	A	G	F

P_1'	A	B	J	H	E	F	G	D	I	C
P_2'	B	H	C	D	J	E	I	A	G	F

c) If we were using simulated annealing to solve the Travelling Salesman Problem, suggest a neighbourhood function.

(7 marks)

This has not been explicitly discussed in the lectures. I have made passing references to operators such as 2-opt, but have not shown it. Therefore, to answer this question the students will have had to read around the subject.
I will award **2 marks** for suggesting a suitable neighbourhood function (such as 2-opt) and a further **5 marks** for showing a demonstration.

Question 6: Answer

a) In the lectures we considered a case study of minimising the distances travelled by football clubs over the Christmas/New Year period.

i) If we were to extend this problem to schedule all the fixtures for the entire season, what changes would have to be made to the model that was presented in the lectures? You do not have to present a formal model. A bullet point list of changes will suffice.

(10 marks)

This was not specifically discussed in the lectures, other than saying that the case study was a subset of the larger problem. The students will have had to read the paper that I made available as part of the reading material for the course. The types of things I would expect are.

- We now have to ensure we have a complete double round robin tournament for each of the four leagues.
- Some of the fixtures would be fixed before the main scheduling (i.e. those that we had previously found for Christmas/New Year).
- Minimising distances are no longer a concern, as we cannot do this over the complete season.
- We would have to consider spreading out the return fixtures (e.g. we have to have a certain amount of time after Liverpool have played Chelsea before Chelsea can play Liverpool).
- We may want define some important matches such that they are scheduled at certain times. As an example, we may not want the top five teams to play too early in the season so that when they do play there is *something to play for*.
- If a team plays at home (resp. away) in the first game of the season, they must play away (resp. home) in the last game of the season.

These are only examples of the types of things that might have to be covered. I am not actually looking for a detailed knowledge of English football, but looking for evidence that the student can think how they would extend an existing model. Therefore, some of the extensions they suggest do not have to make absolute sense with regards to English football. I also know (by a show of hands) that many (the vast majority) of the students do not follow English football and it would unfair to examine this knowledge.

To get full marks the students would have to describe five suggested extensions. Therefore **2 marks** for each valid extension.

b)

i) Describe a problem domain of your choosing, that could be presented as a search problem. Additional marks will be awarded for describing a problem that has not been covered in the course.

(6 marks)

This is an opportunity for the student to show their knowledge of a problem that they are interested in. I will award **3 marks** for problem description and an additional **3 marks** if this problem has not been discussed in the course (and we have covered many problems including bin packing, knapsack, examination timetabling, sports scheduling, TSP, VRP etc.)

ii) Define the hard and soft constraints for your problem. These do not have to be presented in a formal way.

(4 marks)

This provides the student the opportunity to think more in depth about the problem. I will give **2 marks** (pro-rata) the hard constraints and **2 marks** (pro-rata for defining the soft constraints).

iii) Briefly describe what solution methodology you would choose to solve this problem, explaining why.

(5 marks)

There is no right/wrong answer to this. I just want the student to choose an algorithm (either one that has been presented in the lecture or one they have read about) and provide some justification. I will award marks pro-rata.