Diploma Programme
Programme du diplôme
Programa del Diploma

Markscheme

November 2021

Computer science

Higher level

Paper 3

8 pages

Subject details: **Computer science HL paper 3 markscheme**

Mark allocation

Candidates are required to answer **all** questions. Total 30 marks.

General

A markscheme often has more specific points worthy of a mark than the total allows. This is intentional. Do not award more than the maximum marks allowed for that part of a question.

When deciding upon alternative answers by candidates to those given in the markscheme, consider the following points:

- Each statement worth one point has a separate line and the end is signified by means of a semi-colon (;).
- An alternative answer or wording is indicated in the markscheme by a "/"; either wording can be accepted.
- Words in (…) in the markscheme are not necessary to gain the mark.
- If the candidate's answer has the same meaning or can be clearly interpreted as being the same as that in the markscheme then award the mark.
- Mark positively. Give candidates credit for what they have achieved and for what they have got correct, rather than penalizing them for what they have not achieved or what they have got wrong.
- Remember that many candidates are writing in a second language; be forgiving of minor linguistic slips. In this subject effective communication is more important than grammatical accuracy.
- Occasionally, a part of a question may require a calculation whose answer is required for subsequent parts. If an error is made in the first part then it should be penalized. However, if the incorrect answer is used correctly in subsequent parts then **follow through** marks should be awarded. Indicate this with "**FT**".
- Question 4 is marked against markbands. The markbands represent a single holistic criterion applied to the piece of work. Each markband level descriptor corresponds to a number of marks. When assessing with markbands, a "best fit" approach is used, with markers making a judgment about which particular mark to award from the possible range for each level descriptor, according to how well the candidate's work fits that descriptor.

General guidance

Issue	Guidance
Answering more than the quantity of responses prescribed in the questions	• In the case of an "identify" question read all answers and mark positively up to the maximum marks. Disregard incorrect answers. • In the case of a "describe" question, which asks for a certain number of facts *eg* "describe two kinds", mark the first two correct answers. This could include two descriptions, one description and one identification, or two identifications. • In the case of an "explain" question, which asks for a specified number of explanations *eg* "explain two reasons …", mark the first two correct answers. This could include two full explanations, one explanation, one partial explanation *etc*.

1. (a) *Award **[2 max].***
(Search) heuristic / Finds a near optimal solution;
Based on the theory of evolutionary biology / natural selection;
Uses convergence;
Uses a stochastic approach;
Uses a mating pool;
Uses elitism;
Uses a stopping condition;
Uses selection;
Uses crossover;
Uses mutation;
Uses a fitness function;
Uses offspring/generations;
Uses stochastic universal sampling; **[2]**

(b) *Award **[2 max].***
Taking only the fitness members of a population to the mating pool;
Based on a calculated fitness value;
Unfit population members are discarded / fit members are carried forward; **[2]**

2. (a) *Award **[4 max].***

Solution 1

C1: **B** C **G** E J **D** A H **F** I

*Award **[1]** for B D in correct place*
*Award **[1]** for G I F J in correct place*
*Award **[1]** for C E in correct place*
*Award **[1]** for A H in correct place*

Solution 2

C2: **D** A **I** C B **G** H E **J** F

*Award **[1]** for D G in correct place*
*Award **[1]** for I F J B in correct place*
*Award **[1]** for A C in correct place*
*Award **[1]** for H E in correct place* **[4]**

Note to examiners: allow follow through errors, so that if a candidates' logic is correct from the point of an error, they are awarded marks that point onwards.

(b) *Award [**4 max**].*

Offspring 1
PMX algorithm randomly selected sub-sequence of P1 is copied into the first offspring;
P2 contributes the remaining cities to the first offspring;
To ensure that P2 cities are not duplicated, they are mapped against P1;

Offspring 2
The remaining cities in P1 (i.e. not copied to offspring 1) are copied into the second offspring;
P2 contributes the remaining cities to the second offspring;
To ensure that P2 cities are not duplicated, they are mapped against P1;

*Award **[1]** for this alternative offspring 2 solution if offspring 1 is correctly described*

Alternative Offspring 2
The process for offspring 1 is repeated but with a different P1 random subsequence;

*Mark as **[2 max]** for Offspring **1** and **[2 max] for** Offspring 2*

Accept a correct example that explains the process but not replicated from the case study. **[4]**

3. *Award **[6 max].***
A heuristic is an algorithm that does not guarantee an optimal solution;
Instead, a stopping criterion is determined before running;
Heuristic sacrifices accuracy (optimization) for speed;
Heuristics use randomness to escape local extrema / for exploration;
Unlike heuristics, brute-force does not build off previous success/is entirely explorative and novel;
Given enough processing power/time, a brute force approach will always produce an optimal solution;
A brute-force approach is not suitable / heuristic only suitable approach for non-polynomial (computationally intractable) problems;
The number of different permutations if the number of cities is high will make the problem difficult to solve computationally;
In some situations, an optimal solution may be necessary ethically (*e.g.* where people's;
lives are at risk) / In other situations, a non-optimal solution can be used;

[6]

4. *Award **[12 max].***

Roulette wheel selection

- Effective with large-sized problems.
- Low fitness parents of the population have a chance for be selected.
- This is important in hill climbing/avoiding local minima.
- When a population contains only individuals with scores of large, nearly equal values, the selection probability of all individuals becomes nearly identical.
- Other types of selection may be more suitable for smaller search spaces.
- Prevents overly quick convergence.
- Populations must be sorted on every cycle.

Cyclic crossover (CX)

- Allows for two parents chromosomes/cities to be combined/preserved.
- And not be duplicated during crossover.
- Ensures diversity of chromosomes/cities from parents.
- Accelerates the search process in genetic algorithms.

Population size

- It is difficult to decide upon a population size.
- Too large a population will increase the likelihood of a near-optimal solutions but will increase the length of time to run through a generation.
- A large enough population may have enough diversity to arrive at a near-optimal solution without any mutation.
- Too small a population may mean that you may never arrive at a near-optimal solutions;
- Because there is not enough variation in the parents.
- This may require you to increase the mutation rate, which will add greater variation.
- But may also not arrive at a near-optimal solution.

Initial routes

- A random generation of cities would not be an optimal initial population.
- Some form of optimization of the initial population could be done.
- A scale map of the cities could be created and people attempt an optimal route by drawing the path between.
- These human attempts at the optimal solution could be used for the initial population.
- Along with some random routes to ensure variation.

Mutation rate

- Typical mutation rate is around 1%.
- A higher value will result in a more random population.
- Too high and the population will not evolve towards a near-optimal solution.
- Too low and there may not be enough diversity in the population to find a near-optimal solution.
- Mutation allows you to escape from local minima optimisation.

Stopping criterion

- A set number of iterations (generations). This could be based on run for 100 iterations or run for a set period of time. This solution would rely on chance for estimating the number of iterations required to ensure convergence.
- A set number of iterations (generations) when the best route does not change. For example, if after 100 generations the value doesn't change for another 30

generations (*ie*, 30%) then the algorithm stops. This would increase the likelihood of convergence.
- Set a target distance for the route and run the algorithm until that distance is reached.

Conclusion
- A final measured conclusion is included in which the candidate links together the various points in evaluating the probability of success.

Please see markband. **[12]**

Marks	Level descriptor
No marks	• No knowledge or understanding of the relevant issues and concepts. • No use of appropriate terminology.
Basic 1–3 marks	• Minimal knowledge and understanding of the relevant issues or concepts. • Minimal use of appropriate terminology. • The answer may be little more than a list. • No reference is made to the information in the case study or independent research.
Adequate 4–6 marks	• A descriptive response with limited knowledge and/or understanding of the relevant issues or concepts. • A limited use of appropriate terminology. • There is limited evidence of analysis. • There is evidence that limited research has been undertaken.
Competent 7–9 marks	• A response with knowledge and understanding of the related issues and/or concepts. • A response that uses terminology appropriately in places. • There is some evidence of analysis. • There is evidence that research has been undertaken.
Proficient 10–12 marks	• A response with a detailed knowledge and clear understanding of the computer science. • A response that uses terminology appropriately throughout. • There is competent and balanced analysis. • Conclusions are drawn that are linked to the analysis. • There is clear evidence that extensive research has been undertaken.

Total: [30]