M22/4/COMSC/HP3/ENG/TZ0/XX/M

Markscheme

May 2022

Computer science

Higher level

Paper 3

10 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 (;). This does not apply to Question 4.
- 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]***

A heuristic does not guarantee an optimal solution/finds an approximate solution;
Heuristics produce a solution relatively quickly/sacrifices optimisation accuracy for speed;
Used when exact algorithms (e.g. brute force) are impractical/used for computationally intractable problems;

[2]

(b) *Award* ***[2 max]***

Small (initial) population/Lack of diversity in (initial) population;
Failure to preserve population diversity/Genes of high rated individuals dominate the population early in the process;
Low/zero mutation rate;
Stopping condition occurs too soon;
Selection strategy that reduces diversity/Elitism;
Crossover strategy that reduces diversity;

[2]

2. (a) *Award **[4 max]***
*Award **[1]** for two values correctly entered;*
*Award **[2]** for three values correctly entered;*
*Award **[3]** for four values correctly entered;*
*Award **[4]** for five values correctly entered;*

		H	D	E	C	J			

		H	D	E	C	J	B		

		H	D	E	C	J	B	A	

		H	D	E	C	J	B	A	I

G		H	D	E	C	J	**B**	**A**	**I**

G	F	H	D	E	C	J	B	A	I

Award full marks for a correct last stage, even if the stages are not shown.

[4]

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

Initial population size
A large population size gives enough diversity (that makes an optimum solution more likely)/ A small population size lacks diversity (that makes an optimum salutation unlikely);
A large population takes too long to calculate (very long conversion) / A small population is calculated quickly (converges early);

Mutation rate
A very low mutation rate gives little diversity / A high mutation rate provides high diversity;
A low mutation rate tends toward exploitation (narrow search and local optima) / A high mutation rate tends towards exploration (search for too long and never converge);

Example answers with mutation and population size interaction
A low initial population size will give limited diversity/fewer good routes;
So the mutation rate will need to be higher/more cities swapped;

A high initial population size gives plenty of diversity/many good routes;
So a low mutation rate will be needed;

A mid-point sized initial population size and low(ish) mutation rate is necessary;
To stop the GA from converging on a local optima/to ensure the GA converges to a near-global optimum;

To explore the search space a balance between exploration and exploitation is needed;
A medium-sized population size with a low(ish) mutation rate will achieve this;

*Mark as **[2]** for population size + **[2]** mutation.*
*You can also award up to **[4]** if comparisons only are made.* **[4]**

3. *Award* ***[6 max]***

Roulette Wheel Selection (RWS)
Award ***[1 max]***
Solutions are mapped to a roulette wheel / occupying space that is proportional to their fitness;
Better solutions have a greater probability of being selected / Worse solutions have a lower probability of being selected;

Truncation
Award ***[1 max]***
The best / top N solutions are selected (for entry into the mating pool)/The worst / other solutions are not selected (for entry into the mating pool);
The number chosen for N (truncation point) affects the speed of convergence;

Comparison
Award ***[4 max]***
Both solutions rank/sort the solutions from best to worst OR truncation must be sorted but RWS can be implemented without sorting (e.g. rejection sampling);
Thus, the chosen sorting algorithm OR approach affects the solution time (because it is done every cycle);

In truncation, weakest solutions will never be chosen/only strongest solutions can be chosen;
Whereas in RWS every solution has a chance of being selected/even the worst solution has a chance of being selected;

In truncation, once selected the top N solutions have a probability equal to that of any of the other top N solution (e.g. best and 4th best have an equal chance of being selected);
whereas in RWS proportionality is used so the top N percent are differentiated (e.g. best has a high chance of being selected than 4th best);

RWS is more likely to preserve diversity than truncation;
Because roulette wheel is more likely to avoid local minima than truncation;

Both truncation and roulette wheel make their selections based on fitness scores;
So need to consider the processing required to calculate fitness;

Both methods work well with large populations;
So if population diversity is necessary they are good choices/should not be chosen if population sizes are small;

[6]

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

Better essays will be structured to include the following areas:
- An introduction of genetic algorithms (GAs) and route optimization and the problem of vast search space / computational intractability.
- Advantages and disadvantages of GAs.
- Discussion of the characteristics that influence GA outcomes (population size, population diversity, selection method, crossover technique, mutation rate, stopping conditions, and hyperparameters).
- How GAs overcome limitations of premature convergence.
- A final measured conclusion in which the candidate links together the various points (good and bad) in evaluating the extent GAs can solve route optimization problems.

Introduction
- A GA is a search heuristic that a near optimal solution.
- Based on the principle of natural selection.
- Designed for non-polynomial (NP) / computationally intractable combinatory optimization problems.
- GAs sacrifice accuracy for speed by using a combination of exploration and exploitation.
- GAs are useful in problems where an optimal solution is not critical (like route optimization).
- The route optimization in the scenario is the kind of problem where brute force is unsuitable.
- That is because the search space is factorial 20 / 2 or 1.216451×10^{18} different routes.

Discussion points
GAs are suitable when:
- An optimal solution is not required.
- A deterministic algorithm is unavailable or impractical.
- Other heuristic methods are difficult to implement.

Advantages
- Relatively easy to understand and implement.
- Can be easily configured using initialisation parameters.
- Can find near-optimal solutions in a much shorter time than other methods.
- Can use parallel sampling of the search space.
- Variety of approaches can be used including selection method, crossover technique, simulated annealing, and novelty search.
- Different techniques can be combined to improve performance (e.g. truncation selection may be used to provide an initial mating pool then switched to roulette wheel selection).
- A terminating condition can be used (e.g. time, number of generations) to limit the algorithms to use only the available time (e.g. before Lotte leaves).
- Techniques such as simulated annealing can combine early exploration and later exploitation of the problem space.
- Many implementations of GAs are readily available and require no specific algorithm development.

Disadvantages

- Not guaranteed to find the optimum solution.
- Prone to converge prematurely on a local optimum, exploring only a small part of the solution space.
- The population may not converge at all.
- Because of the random aspect, the algorithm will miss good solutions and, even when it finds them, may destroy them with mutation.
- The performance of the algorithm may be highly sensitive to initial conditions (initial population, mutation rate) or other arbitrary choices, such as selection or crossover methods.
- The selection of these initial conditions is mostly by trial and error.
- They only take consider the quantitative data which is available (in this case distance between cities) and do not consider qualitative data (eg. driving conditions, road closures etc).
- They require some expertise to run effectively (ie. correctly set up initialisation parameters, choose appropriate selection strategy etc).
- Due to the “trial and error” nature, a GA is not able to “explain” how it arrived at the near optimum solution.

Characteristics that might be discussed

- Since GAs are so dependent on initial conditions, it often happens that they are run with a range of different sets of initial conditions, in order to find fruitful ones.
- All of the initial conditions affect the others so the balance between them is difficult to determine.
- One GA can be used to find a set of initial conditions for another.
- Initial populations can be 'seeded' with better-than-average solutions.
- It is difficult to choose the ideal population size (too large and the algorithm runs slowly but too small and it lacks diversity).
- The mutation rate can be varied during the run. For example, mutation can be high at the beginning (exploration) and decrease as the algorithm progresses (exploitation), or it can be tied to an indicator of the current diversity of the population.
- The method of calculating the fitness value can slow down GAs.
- The method of terminating the algorithm may affect the solution.
- Stopping conditions include a set number of iterations/specified run time/ minimum target distance/frequency of better solutions found or a combination of these.
- Novelty search can be used to preserve diversity in the population.
- The balance between exploration and exploitation is the key to a successful algorithm;
- Both necessary but fundamentally opposed to each other.
- Functional Programming.

Conclusion

Any reasoned conclusion is fine. For example:

In this scenario with only 20 cities, a GA is not appropriate because alternative methods such as the route inspection algorithm would be more suitable. Besides, the distance covered may not be the most important aspect of the tour. Particular sites of interest or events occurring on a particular day might be more important than a near-optimal route based on distance only.

A GA is a suitable solution because a brute force approach on a home laptop would be computationally intractable and an optimal route is not required, any near-optimal one is fine. Since GAs are likely to be easier to implement than other heuristic approaches, a high school student may find it the best solution for their purpose.

Please see markband below.

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.

[12]

Total: [30]