N22/4/COMSC/HP3/ENG/TZ0/XX/M

Markscheme

November 2022

Computer science

Higher level

Paper 3

9 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]***

You are trying to find an (near) optimal solution/route (combinatorial optimization definition);
to a problem with a finite set of possibilities/city tours (combinatorial optimization requires);

The number of possible routes is so large/factorial problem;
it makes an exhaustive search computationally intractable;

Combinatorial optimization works by reducing the number of possible solutions;
which accelerates search time/makes finding a solution feasible;

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

It is easy to get trapped by local optimum routes/local convergence problems/low exploration;
because it is only looking at adjacent points/new routes are worse than the best current one;
Unable to break free of local/see the global optimum;
To escape from local optima, worse routes need to be selected/greater variation is needed/alternative approach chosen (e.g. novelty search);

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

GA can produce multiple potential routes more quickly than the organizers doing so manually;
GAs calculate distances automatically whereas the organizers may make mistakes;
GA only looks for the fittest(shortest) route, whereas the organizers may have other requirements (e.g. surroundings)/do not need a short route;
The fitness function algorithm may bias the process towards easily measurable properties rather than the more holistic subjective properties that humans make.
GA planned routes might cause unacceptable disruptions (e.g. stop traffic flow), whereas the organizers may consider this;
GA is not biased/does not favour any rider, whereas the organizers may plan the route to favour riders specific riders;
GA won't consider the suitability of road surface/safety whereas the organizers may plan the route for safety/ensure roads are suitable;
GA will not consider spectators whereas the organizers may plan the route to ensure spectators have a good view / see tourist attractions;
GA is likely to only plan in 2D whereas the organizers might consider the terrain depth (e.g. hills, mountains);
GA is unlikely to consider local knowledge (e.g. holidays) whereas the organizers would consider cultural/religious events;

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

The algorithm is approaching convergence;
No way of knowing if this is local or global optimum (so mutation will look for better routes)/has fully exploited the mating pool options;
Mutation provides a method for escaping from local optimum/allows the algorithm to explore more diverse solutions/introduces more routes;
Mutation ensures that you don't have to start again with a new set of random routes;
Applying OX may not produce a new route (preserves features)/offspring route may be identical to one of the parents/ e.g. Offspring I A G C J D H E F B;
The search process will halt early (if only crossover is used);
Mutation ensures that the route inherited/sequence from P1 can change/be reordered;
Mutation does not alter which cities are inherited from P1 and P2, just their order because to do otherwise might result in invalid routes;

3. *Award **[6 max]***
*Award up to **[3]** for evaluating roulette wheel selection.*
*Award up to **[3]** for evaluating tournament selection.*
*Award up to **[1]** for a direct evaluation against each other.*

Roulette wheel selection
Routes are mapped to a roulette wheel with higher fitness routes occupying a greater area/parents occupy space that is proportional to their fitness;
The stochastic nature of RWS supports exploration/allows for diverse routes/any route has a chance to be selected;
RWS is computationally intensive (because the fitness of every solution must be calculated);
RWS can perform poorly when one parent has a very large fitness score;
RWS (native) is biased, favouring earlier routes;

Tournament selection
Several tournaments between routes chosen at random with the winners (best routes) going through to the next stage where they compete against other winners;
Tournament selection works well on parallel architectures allowing it to be scaled horizontally;
Large sample tournaments only select the very strongest members (routes)/promotes elitism;
Smaller tournaments give weaker members/routes a greater chance to move through (it limits diversity/exploration);
Tournament selection does not need to be sorted/computationally inexpensive;
Tournament selection can lead to premature convergence;

Direct evaluation against each other
Tournament selection has been shown to outperform RWS for large populations;

Note: Accept diagrams that evaluate the selection methods.

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

Convergence

Convergence causes evolution to halt because precisely every individual/route in the population is identical.
Ideally, this will be a near-optimal route.
Premature convergence is when a GA reaches convergence but not at a near-optimal route;
Premature convergence tends to occur when there is no mutation present (i.e. random swapping of routes), and only crossover has been used.
Novelty search does not define progress by performance and instead rewards being different, which may assist with overcoming local minima/maxima problems.
As the GA approaches convergence, a stopping condition may cause the algorithm to stop.
The nature of the stopping condition will affect convergence.
GAs tend to be slow so waiting until convergence may take too long.

Terminating before convergence

The algorithm may be programmed to stop after a set number of generations or a specific period of time.
Set a probability p of accepting a worse solution, where p decreases over time.
A set number of iterations (generations) when the best route does not change. For example, if after X generations, the value doesn't change, the algorithm terminates.
Specify a target distance for the route and run the algorithm until that distance is reached.

Exploration vs. exploitation

Every search algorithm addresses the exploration and exploitation of a search space (i.e. number of potential routes).
Exploration is the process of visiting entirely new regions of a search space (i.e. routes that have not been considered).
Exploitation is the process of visiting those regions of search space within the neighbourhood of previous routes (i.e. similar routes to the best solution so far).
A search algorithm needs to establish a good ratio between exploration and exploitation to be successful.
Exploitation can be achieved by choosing certain selection types (i.e. choosing the fittest parents).
Exploration can be achieved through crossover and mutation (i.e. swapping cities).
Exploration vs exploitation is a balance between the combinations of selection, crossover, and mutation.
For example, tournament selection with a large pool will ensure the strongest routes are chosen (exploitation). Tournament selection with a small pool may have a relatively weak route winning and move the balance towards exploration.
Exploration vs exploitation in the travelling salesman problem is which chromosomes (i.e. cities) to keep and which ones to change.
Novelty search moves the balance towards exploration because it rewards being different, which may assist with overcoming local minima/maxima problems.
When a GA finds a near-optimal local maxima a decision to carry these chromosomes (i.e. city routes) to the next generation is made. Keeping these cities is called exploitation, whereas discarding them for a potentially better global maximum is called exploration.
The crossover operator is used for exploring the sample space, so it is called the exploration operator, while the mutation operator is used to exploit the promising locations found by the crossover operator, so it is called the exploitation operator.
Algorithms may start off with higher exploration and increase exploitation over time.
By estimating the global optimum, it is possible to ensure that the GA does not suffer from premature convergence.

Simulated annealing (SA)
SA is a probabilistic technique for approximating the global optimum in a large search space (i.e. billions of potential routes).
Suited to NP-hard optimization problems.
Analogous to annealing in metallurgy (how the slow cooling is implemented), the simulated annealing algorithm is interpreted as a slow decrease in the probability of accepting worse solutions as the solution space (potential routes) is explored.
Accepting worse solutions/routes allows for a more extensive search for the global optimal solution.
The algorithm randomly selects a solution (i.e. route) close to the current best route;
It then measures its quality (i.e. total distance of random against total distance of best route so far).

Conclusion
A final measured conclusion is included in which the candidate links together the various points in evaluating convergence.

Note that simulated annealing is not required, but credit students who mention it.

Please see markband on page 11.

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]