ANSWER BOOKLET
LIVRET DE RÉPONSES
CUADERNILLO DE RESPUESTAS

International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

4 PAGES / PÁGINAS

Candidate session number: / Numéro de session du candidat: / Número de convocatoria del alumno:

2	2	M	T	0	P	3	-	C	S	H	L

Candidate name: / Nom du candidat: / Nombre del alumno:

At the start of each answer to a question, write the question number in the box using your normal hand writing / Avant de répondre à une question, inscrivez son numéro à la main dans la case appropriée / Al comienzo de cada respuesta, escriba a mano el número de pregunta en la casilla.

Example / Ejemplo 27 | 2 | 7 |

Example / Ejemplo 3 | | 3 |

1

(a) A heuristic algorithm is a type of non-deterministic algorithm that uses a rule of thumb to return a good, or "good enough" solution for the problem at hand. It is not deterministic, in that the most optimal solution is not found, as speed is sacrificed for accuracy.

(b) Premature convergence may occur due to:

- not enough diversity in the population (likely due to too much exploitation)
- too small a population size, which means that not enough of the problem space is explored.

2

(a)

		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

(b) Initial population size: the number of solutions which will initially exist within the population, a subset of the entire problem space! This should also remain the same throughout generations.

- ~~Lag~~ Large population size means that the ~~initial~~ problem space is already reasonably well explored upon initialisation, enabling less fit solutions to progress throughout generations. This exploration will help escape local optima and better reach a good solution, but not THE OPTIMAL solution as described in the question.
- ~~Using set~~
- Too large a population will ~~decrease~~ increase the ~~time~~ amount of resources required however is ~~it~~ unlikely to detract from the probability of finding a good solution.

Mutation rate: the probability that a gene in a solution (chromosome) is randomly flipped (binary encoding) or swapped with another (other types of encoding like permutation encoding in the TSP).

- ~~Mutation rate well~~ A higher mutation rate is likely to increase diversity, meaning the algorithm can escape local optima and be more likely to progress to a global optima, hence increasing the chance of finding a good solution
- Too high a mutation rate will prevent any fit solutions from progressing through

generations, likely meaning convergence may never be reached. This way, the probability of finding an optimal solution is, in fact, reduced.

3 Roulette wheel selection involves assigning a probability of selection to each solution based on its fitness in relation to others:

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$$

Here, each chromosome is given a "slice" of a roulette wheel proportional to its probability, p_i.

→ this gives opportunity for exploration of the problem space as chromosomes that are less fit have a chance of being selected during the selection process, and being sent to the mating pool. By increasing such exploration, the chance of becoming stuck in local optima is reduced, hence the genetic algorithm may be more successful.

→ However, particularly with smaller population sizes, very fit solutions may still dominate the selection process, and other than mutation, there is little that can be done to reduce the exploitative aspect of the roulette wheel's biased

division of the wheel.

Truncation selection (also known as elitism selection) involves simply removing less fit individuals from the population, hence allowing fitter individuals to proceed to the mating pool (or be copied to the following generation).

→ this method is very computationally efficient as complex parameters like roulette wheel slices need not be generated
→ this method also allows fitter individuals to pass on their traits to following generations, ensuring that good genetic material is withheld within amongst the iterative generations, and that a good solution can be found.
→ However, this method, given it's name, is very exploitative, using the idea of elitism to remove less fit individuals. This likely would lead to premature convergence in local optima, especially if the initial population did not contain very fit individuals (which is likely in most scenarios). In most cases, therefore, truncation/elitism selection is seen as unfavourable.

→ RWS is rather heavy on the computer, as relative probabilities must be calculated for each solution

Candidate session number: / Numéro de session du candidat: / Número de convocatoria del alumno:

22 M TZ 0 P 3 - C S H L

Candidate name: / Nom du candidat: / Nombre del alumno:

At the start of each answer to a question, write the question number in the box using your normal hand writing / Avant de répondre à une question, inscrivez son numéro à la main dans la case appropriée / Al comienzo de cada respuesta, escriba a mano el número de pregunta en la casilla.

Example / Ejemplo 27 [2 7]

Example / Ejemplo 3 [3]

3

Therefore, while both methods have their advantages, the computa resource savings of truncation selection usually are not great enough to justify a purely exploitative approach. this way, RWS, & other methods like SUS and Tournament selection are generally seen as more favorable due to the use of exploration.

4

There are a ~~vast~~ variety of methods known to date to solve route optimization problems like the TSP. For a proble reasonably sized problem space, brute force is a good deterministic method of finding the most optimal route, however for ~~pr~~ combinatorial optimisation problems like the TSP presented in the case study, ~~heuristic solo~~ stochastic solutions, as well

heuristic methods like the GA may be used.

Route optimisation problems require a problem space which spans over ordered permutations of a given string of "cities" to create a chromosome, or "tour". Genetic algorithms, in this case, use a special type of encoding called permutation encoding, to adhere to the requirements of a solution. This way, the hyper-parameter of encoding ~~is ideal~~ makes the GA an appropriate method for solving route optimisation problems. Not only that, but other hyper parameters such as crossover and mutation (which will be discussed) can also be fine tuned for ~~the~~ the specific requirements of route optimisation problems, making GAs all the more ideal.

Due to the nature of the permutations of cities, traditional brute force methods do not work on most ~~of~~ real route optimisation problems, meaning it is, in most cases, impossible to find an algorithm that will find a deterministic solution to the problem (e.g. TSP). Furthermore, ~~most~~ traditional methods will aim to use heuristic methods, such as hill climbing, in order to optimise a solution (e.g always climb the hill, hence finding better solutions), however often fail in many regards. Importantly is the complex nature of combinatorial optimisation problems, where the problem space is a complex wave of "peaks" and "troughs" (maxima and minima, or optima).

The aforementioned hill climbing method would quickly become stuck at local optima, hence preventing a sufficiently good solution to be found. As the problem space is not known beforehand, it is impossible to have foresight into which is a local and global optima, rendering hill climbing algorithms relatively useless without some element of tweaking.

Genetic algorithms (GAs) aim to "strike a natural balance between exploration and exploitation" in order to both sacrifice accuracy for speed, but also ensure that enough of the problem space is explored (in contrast to hill climbing) to ensure local optima are not converged upon. The way in which GAs therefore reward novelty throughout the iteration of generations, means that both the benefits of brute force exploration and hill climing exploitation are combined into a single, better algorithm, the Genetic Algorithm.

However, GAs are rendered useless if their hyper-parameters are not set properly, such as the initialisation parameters, as well as other factors like selection and crossover strategy, as well as termination conditions, mutation rate and crossover rate. These parameters have complex interdependencies, and while there are research papers and accepted strategies, these vary widely for different applications of the route optimisation problem. While hill climbing

and route space are relatively straightforward, GAs must be set-up in such a way that they are not too exploitative as to ~~find~~ reach local optima, but not too explorative as to waste computer resources and never converge. Techniques like simulated annealing may be used to dynamically change the mutation rate parameter, but the fact remains that there is still no definitive method or accepted solution. In this sense, meta-heuristics may help aid in using general ~~heuristics~~ heuristics to decide upon parameters the ~~of~~ GA's hyper-parameters, however at their core, their meta-heuristics still fall back on imperfection- it is unlikely that they would ever provide the perfect solution.

~~In essence, essence,~~

All in all, the argument for genetic algorithms in solving route optimisation problems has no definitive or accepted solutions. GAs can be currently accepted as one of the better methods of finding "good" solutions, but whether they are "good enough" can be an ethical problem based on the scenario (e.g. does the length of the route impact people's lives negatively?). ~~After~~ After all, GAs sacrifice accuracy for speed, and the main question is of the trade-off between these two ideas. So, the question stated to what extent GAs may "solve" route optimisation problems, and the extent to which they may solve them are ~~limited~~ vast, however limited by the non-deterministic nature of heuristics and the difficulty of striking a balance between exploration and exploitation.