This is a closed book test. You may not refer to any resources.

This is a 50 minute test.

Please write your answers in ink. Pencil answers will be marked, but will not be reconsidered after the test papers have been returned.

The test will be marked out of 50.

<table>
<thead>
<tr>
<th>Question</th>
<th>Mark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1</td>
<td>8</td>
</tr>
<tr>
<td>Question 2</td>
<td>5</td>
</tr>
<tr>
<td>Question 3</td>
<td>5</td>
</tr>
<tr>
<td>Question 4</td>
<td>30</td>
</tr>
<tr>
<td>Question 5</td>
<td>2</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>50</strong></td>
</tr>
</tbody>
</table>

*If we didn’t have genetic mutations, we wouldn’t have us.*
*You need error to open the door to the adjacent possible.*

---

*Steven Johnson*
Question 1 (8 Marks)

**Branch and Bound Algorithms** and **Genetic Algorithms** are both designed to find a solution to an optimization problem by searching the set of possible solutions. B&B proceeds by constructing good solutions step by step, while GA takes a set of possible solutions and modifies them to try to create better solutions.

(a) Under what conditions would it be preferable to use a Branch and Bound Algorithm rather than a Genetic Algorithm?

**Solution:** Genetic Algorithms cannot guarantee that the solution found will be optimal. If we absolutely need to find an optimal solution, Branch and Bound is preferable.

**Marking:** This is the most likely answer, but another acceptable response is that B&B would be suitable when the bounding functions are excellent or the number of decisions is small (which can limit the number of partial solutions to be considered). Any good answer that correctly identifies a situation where B&B is preferable should get 4/4. An answer that gives an invalid reason for preferring B&B but shows some understanding of the difference between B&B and GA should get 2/4.
(b) Under what conditions would it be preferable to use a Genetic Algorithm rather than a Branch and Bound Algorithm?

Solution: If we are dealing with a large instance of an optimization problem and we are satisfied with finding a good (but not necessarily optimal) solution, then a Genetic Algorithm will be preferable because it will find a solution more quickly.

Marking: For full marks, the student should make reference to GAs being appropriate when the optimal solution is not mandatory.

As with part (a), an incorrect answer that shows some understanding of the difference between B&B and GA should get at least 2/4
Question 2 (5 Marks)

What is the purpose of mutation in a Genetic Algorithm?

Solution:

If we only used cross-over operations, we would never be able to go beyond the elements of the solutions in the initial population. For example if our solutions consist of bit-vectors, then if there happens to be a bit that is 0 in all members of the initial population, then without mutation we will never see any solutions that have a 1 for that bit.

Another reason is that mutation can help us move away from a local optimum solution, which increases our chance of finding a global optimum.

Marking: the student’s answer should relate to the benefit(s) of occasionally introducing random elements into the population.

If they explain the process of mutation (randomly selecting and modifying some element of a solution in the current population) but cannot explain the purpose of it, they should get at least 2/5.
Question 3 (5 marks)

In a Genetic Algorithm we choose members of the current population to be the parents of new offspring, with preference given to the stronger members.

Why don’t we simply use every possible pairing of members of the current generation to produce new offspring, and then keep only the best ones?

(Hint: what if the current population has 1000 members – which is an entirely plausible number for real-world applications?)

Solution: If the population has 1000 members and we use every possible pair to produce new offspring, we will have at least \( \binom{1000}{2} \approx 500,000 \) new offspring in every generation. The time required to produce and process all of these solutions would negate the advantage of the rapid convergence of the genetic algorithm.

Marking: The answer above is the most natural one, but students may not state it exactly this way. Anything like “the number of offspring would be unmanageably large” is fine.

Students may also say something like “This would result in applying crossover to pairs of weak solutions, which will most likely result in even more weak solutions.” This is not as strong an answer as the computational issue already mentioned, and should get about 3/5
Question 4: (30 marks)

The League of Rational People has successfully landed on Titan, thanks to your excellent work on assigning people to space ships. Now comes the task of assigning people to living space. Nobody got left behind so there are still exactly n people.

It turns out that the surface of Titan holds a number of caves, all conveniently warm, well lit and filled with breathable air.

Each person is to be granted some number of square meters of floor space for their personal use. The number can be any value in the range [50 .. 100]. The more space a person has, the happier they will be – in fact there is a happiness function $f(p, s)$ that tells you how happy person $p$ is if they are given $s$ square meters. Unfortunately you can’t just give everyone 100 square meters because the total allocation of space must be $\leq k$, where $k$ is the total amount of available floor space (we know $50n \leq k < 100n$).

Design a genetic algorithm to search for a good solution to this problem. Your goal is to maximize the sum of the happiness of the population.

For most parts of this question there is more than one reasonable answer. Whatever answer you give, explain your reason for choosing it.

There are six parts to this question. Each part is worth 5 marks.
(a) Describe how you will represent each solution as a vector. Specify clearly what the individual elements of the vector represent.

Solution: I would use a vector of real numbers in which each element represents the amount of space allocated to a specific person. The people would be in a specific order, but it can be arbitrary. No sorting is required.

Marking: It is also possible to use a bit-vector representation, for example by allocating 7 consecutive bits to each person – with 7 bits we can represent every integer in the range [50..100]. If we want finer granularity we can use more bits per person.

Give full marks for any solution that represents the necessary information (how much space is allocated to each person) in a usable way.

Give 2/5 to anyone who can explain what an appropriate vector would look like, even if they can’t properly define one.

(b) Describe how you will compute the value of any given solution.

Solution: I would total the happiness of all people. If the people are numbered 1 to n and a particular solution $S = \{s_1, \ldots, s_n\}$ then the total happiness for $S$ is

$$\sum_{i=1}^{n} f(i, s_i)$$

Marking: If they say “the value is the total happiness of the population” they should get 3/5, even if they cannot come up with the formula.
(c) Describe how you will ensure that each vector of values in your population represents a feasible solution.

Solution: a solution is feasible iff each element is in the range \([50 \ldots 100]\) and the sum of all the elements is \(\leq k\)

Marking: The answer can be given descriptively (as above) or in pseudo-code. For full marks they must mention both criteria.

(d) Describe how you will generate the initial population of feasible solutions.

Solution: To generate a feasible solution I would start by allocating 50 square meters to each person. The remaining \(k - 50n\) square meters would be randomly assigned to the people, making sure that nobody is given more than 100 square meters in total. One way to do this would be to assign the remaining space one square meter at a time, until all had been allocated. When doing this, it would be necessary to make sure nobody gets more than 100 square meters.

Marking: Clearly there are many other ways and to some extent it depends on the representation chosen. For example if they have chosen a bit-vector representation then they could randomly decide which bits to set to 1, then test for feasibility. Give full marks to any method that will generate random feasible solutions.
(e) Describe the cross-over operation you will use.

Solution: I will use the simulated binary cross-over method. I could also use the arithmetic cross-over method, but that would require more mutations to properly cover the solution space.

Marking: As long as their chosen cross-over method works with their chosen representation (bit-vector, vector of reals, etc.) they should get full marks.

(f) Describe the mutation operation you will use.

Solution: I will randomly determine which of the new offspring will mutate. For each one that does, I will randomly select one element. Then I will randomly change it by a small percentage, either up or down. When doing this I would make sure that the element being modified does not go outside the [50 .. 100] range of legal values.

Marking: same as for part (e)
Question 5: (2 marks)

True or False: The Hamming Cliffs are located in the Rocky Mountains near Banff.

FALSE

Solution: False