

---

# ECMM409 - Coursework exercise

---

**Deadline: 15th November 2023 12:00 (midday, Exeter time)**

Lecturers: Dr. Ayah Helal (a.helal@exeter.ac.uk)

Dr. David Walker (d.j.walker2@exeter.ac.uk)

---

This assessment (CA) is worth 40% of the overall module assessment. This is an individual exercise and your attention is drawn to the faculty and University guidelines on collaboration and plagiarism, which are available from the faculty website.

**Referencing, and academic conduct** You are required to cite the work of others used in your solution and include a list of references, and must avoid plagiarism, collusion and any academic misconduct behaviours.  
<https://ele.exeter.ac.uk/course/view.php?id=1957>

**Coursework submission** The submission of this coursework is via **EIE2**. The implementation can be in the programming language of your choice, such as Python, Java, C#, etc., and it must run without errors. Please ensure that your code is clear and well-commented, and that your answers are clearly labelled. The implemented code should have a README file to support the execution of the code.

In **EIE2**, you should submit a **single compressed (zipped) folder** which contains all the relevant files for this coursework exercise. The compressed folder can contain a PDF file with your report and your code. The report should have a **maximum of 4 pages (4 sides of A4, references do not count towards the limit)** which should include a description of your experiments where tables and/or graphs of results should take up no more than 2-3 pages, and your answers to the questions/descriptions in the remaining space. **Your code should be clearly commented and submitted as a zip file.**

**Marking scheme** The marking scheme for this assessment is as follows.

Correct and efficient implementation of the algorithm	15%
Quality of the experiments design	15%
Documentation of code	5%
Correct results from the EA runs	20%
Quality (e.g. readability & usefulness) of tables and graphs	20%
Answers to Questions	20%
Conclusions	5%
Overlength submissions (per page)	-10%

## Question 1 | Evolutionary Algorithm.

**The Task** is to implement an Evolutionary Algorithm (EA) to optimize the Travelling Salesman Problem. Your objective is to conduct a series of experiments to determine the most suitable parameters for the algorithm when applied to this problem. You have the flexibility to choose any programming language for your implementation.

In the following sections, you will find information about the problem itself, followed by essential details about the EA algorithm, and lastly, a description of the experiments that you should carry out.

### Design Problem

Your company conducts sales operations across various international cities, and you have been provided with maps for two specific sets of cities, one in Brazil (brazil.xml) and the other in Burma (burma.xml).

These maps include details about the cities and the associated travel costs between them, all provided in an XML format, which is further explained in the attached “TSP XML format” file. To address this optimization challenge, you can make use of a distance matrix denoted as  $D$ . In this matrix, each element represents the distance between two locations. Your goal is to minimize the total cost, which is defined as:

$$cost = \sum_{i=1}^{n-1} D[C[i]][C[i+1]] + D[C[n]][C[1]]$$

You will achieve this by finding the most efficient route using a vector, denoted as  $C$ , where each entry  $C[i]$  corresponds to a unique integer value ranging from 1 to  $n$ , representing the location identifiers. Your objective is to minimize this cost, thus optimizing your travel routes in both the Brazil and Burma cities.

### The Evolutionary Algorithm

The evolutionary algorithm should be a steady state implemented as follows:

1. Generate an initial population of  $p$  randomly created solutions and assess the fitness of each individual in the population.
2. Use tournament selection twice to select two parents, denoted as  $a$  and  $b$ .
3. apply a single-point crossover on these selected parents to generate two children, referred to as  $c$  and  $d$ .
4. Run a mutation on  $c$  and  $d$  to give two new solutions  $e$  and  $f$ . Evaluate the fitness of  $e$  and  $f$ .
5. Run replacement function, firstly for  $e$ , then  $f$ .
6. If a termination criterion has been reached, then stop. Otherwise return to step 2.

**Termination Criterion:** The termination criterion is simply reaching a maximum of 10,000 fitness evaluations, as detailed in the implementation and experimentation section below.

**Tournament Selection:** In tournament selection,  $N$  chromosomes are randomly chosen from the population, and their fitness is recorded. The fittest of these  $N$  individuals, breaking ties randomly, becomes the selected parent.

**Single-Point Crossover:** For single-point crossover, a ‘crossover point’ is randomly selected, which should be smaller than the total length of the chromosome. A crossover function is applied to ensure that the problem’s constraints are not violated.

**Swap Mutation:** Swap mutation involves selecting two distinct genes randomly and swapping their positions within the chromosome.

**Replacement:** Choose and implement one of the replacement functions discussed in the lectures to determine how the newly generated solutions are introduced into the population during the evolution process

## Implementation and Experimentation

Design a series of experiments to explore how varying different parameters and operators affects the performance of the EA. Consider varying one parameter at a time while keeping others constant. Some parameters/operators to explore include:

- Crossover operator (crossover with fix, ordered crossover).
- Mutation operator (single swap mutation, inversion, multiple swap mutation).
- Population size (e.g., 50, 100, 200).
- Tournament size (e.g., 5, 10, 20).

## Experiment Execution

Execute the designed experiments for both the Brazil and Burma TSP instances by varying the selected parameters/operators. For each experiment, run multiple trials with different random number seeds.

## Data Collection and Analysis

Collect data on the performance of the EA for each experiment, such as the best solution found, convergence curves, and execution time. Analyze the results to understand how changes in parameters/operators impact the algorithm's performance in terms of solution quality and convergence speed.

## Draw Conclusions

Based on the results, draw conclusions about which parameter settings and operators work best for solving the TSP instances (Brazil and Burma). Identify trends and insights that can inform the choice of parameters and operators for similar optimization problems. Remember to keep a record of all experimental details and results to ensure a thorough analysis of the EA's performance. Answer those questions on the experiment you implemented.

**Hint:** You should think carefully about how best to present your results to show the behaviour of the algorithm during your trials.

**Question 1:** Which combination of parameters produces the best results?

**Question 2:** What do you think is the reason for your findings in Question 1?

**Question 3:** How do each of the parameter settings influence the performance of the algorithm?

**Question 4:** Can you think of a local heuristic function to add?

**Question 5:** Can you think of any variation for this algorithm to improve your results? Explain your answer.

**Question 6:** Do you think of any other nature inspired algorithms that might have provided better results? Explain your answer.

In your answers, describe your observations of the results and explanations or conclusions you can draw from your results.