COURSEWORK SPECIFICATION
ECM3412 – Nature-Inspired Computation
Module Leader: Alberto Moraglio
Academic Year: 2024/25

Title: **Programming Task & Report**

Submission deadline: **6 November 2024 (12:00 noon)**

This assessment contributes **40%** of the total module mark and assesses the following **intended learning outcomes**:

- demonstrate a clear understanding of the difficulties associated with certain intelligence-related tasks that we would wish to program computers to do;

- evaluate a difficult problem and determine the likely best algorithm selection;

- implement software for addressing real-world optimisation problems with nature-inspired methods;

- digest and communicate succinctly information from publications in the field to individuals unfamiliar with the material.

This is **individual** assessment and you are reminded of the University's regulations on collaboration and plagiarism. You must avoid plagiarism, collusion, and any academic misconduct behaviours. Further details about academic honesty and plagiarism can be found at https://ele.exeter.ac.uk/course/view.php?id=1957.

Use of GenAI tools

The University of Exeter is committed to the ethical and responsible use of Generative AI (GenAI) tools in teaching and learning, in line with our academic integrity policies where the direct copying of AI-generated content is included under plagiarism, misrepresentation and contract cheating under definitions and offences in TQA Manual Chapter 12.3. This coursework is categorized as **"AI-supported"** because it involves complex tasks that require students' original work and critical thinking, such as algorithm implementation, experimentation, and analysis. While AI tools can assist with literature review, proofreading, and data visualization, the core elements—including the ACO implementation, experimental runs, and interpretation of results—**must be the student's own work.** This approach allows for responsible AI use while ensuring the assessment meets its learning objectives and maintains academic integrity.

You can find further guidance on using GenAI critically, and how to use GenAI to enhance your learning, on Study Zone digital.

Instructions

Task

What you will do in this assignment is write a 'research paper' on the application of ant colony optimisation (ACO) to the problem of bin packing. You will need to research the various nature-inspired algorithms that have been applied to this problem, implement the ACO approach and then carry out a variety of experiments to help find out what parameters for the algorithm are best for this problem. The implementation can be in the programming language of your choice. In the remainder of this document, the following is provided: details of the problem, the basic details of the algorithm, the requirements for the literature review and a description of the experiments you should carry out. The final section indicates what should be in your report to be handed in.

The Bin-Packing Problem

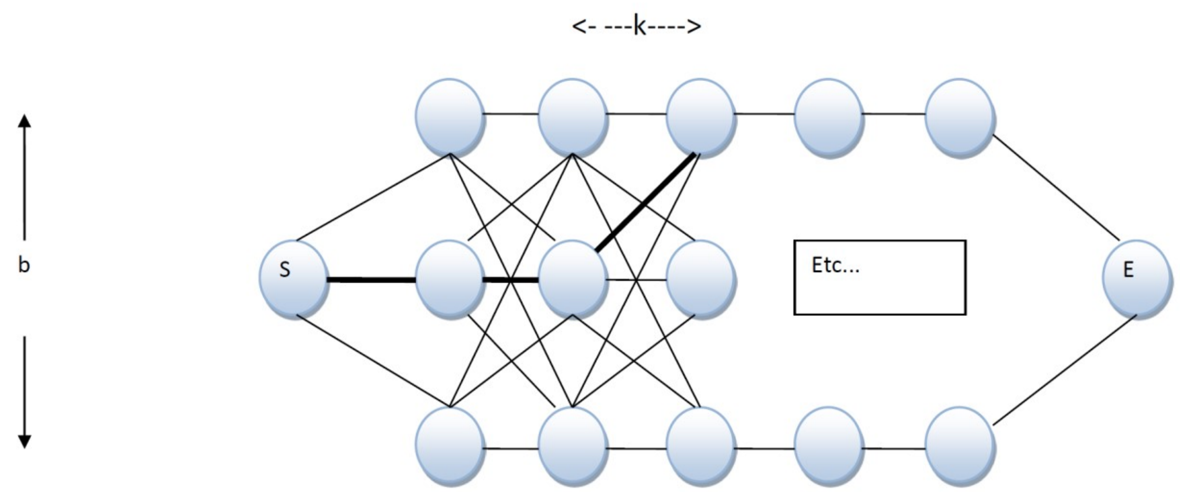
The bin-packing problem (BPP) involves *n* items, each with its own weight. Each item must be placed in one of *b* bins. The task is to find a way of placing the items into the bins in such a way as to make the total weight in each bin as nearly equal as possible. E.g., if there are 6 items with weights as follows: (17, 12, 19, 6, 4, 28) and we place them into 3 bins as follows indexing from 1:

(bin1: 1, 3) (bin2: 4) (bin3: 2, 5, 6)

then bin1 has weight 36 (17+19), bin2 has weight 6, and bin3 has weight 44 (12+4+28). These are quite different, so it is a poor solution. We can measure the solution quality by taking the difference *d* between the heaviest and lightest bins – in this case *d* = 38. What we want to do here is to minimise this difference. A far better solution in this case is:

(bin1: 1, 2) (bin2: 3, 4, 5) (bin3: 6)

with *d* = 1. You will need to create a construction graph to represent the problem. This should be a structure which contains the pheromone values for every possible decision made by the ants. For instance, a problem with *k* items and *b* bins will look like this:



Therefore an ant travelling from the start (S) to the end (E) will have to place each of the *k* items into one of the *b* bins. The route taken by an ant marked in bold shows that item 1 will be placed in bin 2, item 2 will also be placed in bin 2, and item 3 will be placed in bin 1.

The **fitness** of a route is worked out by calculating the difference *d* between the heaviest and lightest bins, where the goal is to minimize this difference. There are two specific bin-packing problems I want you to address. In each, there are 500 items to be packed into a number of (*b*) bins (either 10 or 50). In problem BPP1, $b = 10$ and the weight of item *i* (where *i* starts at 1 and finishes at 500) is ***i***. In problem BPP2, $b = 50$ and the weight of item *i* is $(i^2)/2$.

The Ant Colony Optimisation Algorithm

Implement the ACO like this:

1. Randomly distribute small amounts of pheromone (between 0 and 1) on the construction graph.
2. Generate a set of *p* ant paths from S to E (where *p* is a variable and specified below).
3. Update the pheromone in your pheromone table for each ant's path according to its fitness.
4. Evaporate the pheromone for all links in the graph.
5. If a termination criterion has been reached, then stop. Otherwise return to step 2.

Termination Criterion: When the algorithm has reached a maximum number of fitness evaluations (generated paths). The result is then the fitness of the best ant in the population at the end.

Generating Ant Paths: An ant will traverse your construction graph by making a decision at each new item it comes to (i.e. an ant at S can choose to go to bin 1, 2 or 3 in the illustration above). This selection is made at random, but biased by the amount of pheromone on the choices ahead (e.g. if an ant is placed at position S and bin 1 has a pheromone value of 0.5, bin 2 has a pheromone value of 0.8 and bin 3 has a pheromone value of 0.1, the ant should have a 5/14 chance of selecting bin 1, an 8/14 chance of selecting bin 2, and a 1/14 chance of selecting bin 3). This should be repeated for all *k* variables and *b* bins. There is no local heuristic for this implementation.

Pheromone Update: Once the fitness has been computed, the pheromone must be updated accordingly. With the bin packing problem, we want to reward paths that lead to bin packs with smaller differences. Therefore the pheromone update for the path will be *100/fitness*.

Pheromone Evaporation: Finally, the pheromone on all paths must be evaporated. This is achieved simply by multiplying all paths within the construction graph by the evaporation rate *e* (specified below).

Implementation and Experimentation

Implement the described ACO in such a way that you can address the BPP problems, and then run the following experiments and answer the subsequent questions. Note that, in all of the below, a single trial means that you run the algorithm once and stop it when 10,000 fitness evaluations have been reached. Different trials of the same algorithm should be seeded with different random number seeds.

Run five trials of the ACO with $p = 100$ and $e = 0.90$
Run five trials of the ACO with $p = 100$ and $e = 0.60$
Run five trials of the ACO with $p = 10$, and $e = 0.90$
Run five trials of the ACO with $p = 10$, and $e = 0.60$

Do all of the above, first on BPP1 and then on BPP2 (as defined in the Bin Packing Problem section). Record the best fitness reached at the end of each trial and any other variables you think will be important.

‘Conference Paper’ Report

Your submission should be a PDF in the style of a paper using one of the style files (Latex style file or Word document template) posted on ELE. **Please do not include your name in the ‘author’ section – simply replace the name with your candidate number.**

The paper should have a **maximum of 4 pages** (4 sides of A4, references do not count towards the limit), which should include a short literature review on nature-inspired approaches to the bin-packing problem and a description of the results of your experiments where tables and/or graphs of results should take up no more than 2 pages. In the remaining space, write a ‘Discussion and Further Work’ section including your answers to the following questions.

Question1: 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: Do you think that one of the algorithms in your literature review might have provided better results? Explain your answer.

In your answers, describe your observations of the results, and describe any tentative explanations or conclusions you feel like making, and describe any further experiments you felt it interesting or useful to do.

Your paper should therefore minimally include the following sections: 1) Literature Review, 2) Description of Results, 3) Discussion and Further Work, 4) References.

Marking criteria

Submission

In ELE2, you should submit a single compressed (**.zip**) folder which contains:

Paper: a PDF file with your report;

Code: a text file with **clearly commented code**;

GenAI: a MS word file with the **completed GenAI student declaration**;

Marking Scheme

Quality of Literature Review	15%
Correct and efficient implementation of the algorithm	10%
Quality of code documentation	5%
Correct results from the ACO runs	20%
Quality (e.g. readability & usefulness) of tables and graphs	15%
Answers to Questions 1-4	20%
Further Experiments	10%
Conclusions	5%
Over length submissions (per page)	-10%