

School of Computing Assessment Brief

Module title	Artificial Intelligence
Module code	COMP2611
Assignment title	Search Algorithms
Assignment type and description	Report detailing investigation, evaluation and customisation of algorithms (based on use of provided software).
Rationale	The assignment is designed to consolidate students' understanding of search algorithms and develop an appreciation of how different algorithm variants perform in relation to characteristics of the problem being addressed.
Length limit and guidance	8–10 pages (+ cover page). It is expected that your answer for each of the 8 parts (A1–4, B1–4) should fit on one page. You may go over to a 2nd page if you think you really need to.
Weighting	20% of total module grade
Submission deadline	3rd March
Submission method	Gradescope (one member of pair submits then adds partner)
Feedback provision	Feedback will be provided via Gradescope.
Learning outcomes assessed	<ul style="list-style-type: none"> • Be able to employ AI techniques to solve problems • Evaluate the effectiveness of AI techniques. • Identify limitations of AI techniques.
Module lead	Brandon Bennett B.Bennett@leeds.ac.uk
Other Staff contact	Arash Rabbani: A.Rabbani@leeds.ac.uk

COMP2611 Coursework 1

Search Algorithms

Overview

Your task is to investigate the effectiveness of a variety of different search algorithms (and option settings) for solving: (A) the 8-puzzle problem; and (B) a simple task planning problem that might be assigned to a robot worker.

You will work on and submit your assignment as a *pair* (i.e. two students working together). You must both contribute to work on both parts of the assignment. It is not acceptable to work completely separately on different tasks (and this would be likely to result in lower marks as the parts are closely related.)

Provided you have kept up with the lectures and gone through the Search Exercises, I would expect that working alone you can produce a report that will get you a pass mark will require both students to spend around 5 hours on this assignment. Of course, gaining a high grade is likely to take considerably longer.

Preliminaries

Your preliminary work for before starting this assignment should be to study the Search Exercise interactively executable notebook files that I have provided and can be accessed via a folder within the Learning Resources area of the module's Minerva pages.

Assignment Requirements and Marks

The overall structure of your assignment task is as follows:

A. 8-Puzzle Search Investigation

This can be done working within the Search Exercise 6 notebook. You can use the notebook directly online, but *take care to save your own code and results*.

Report sections and associated marks:

- A1.** Design a sequence of tests that show the strengths and weaknesses of different search algorithms when applied to the 8-Puzzle problem. [2 marks]
- A2.** Present your results in the form of one or more tables. [6 marks]
- A3.** Give a list of key observations as concise bullet points. [4 marks]
- A4.** Define your own `misplaced.tiles` and `manhattan` heuristic functions. [4 marks]

Extra Marks are available for exceptional features of the work done in addressing the requirements (e.g. particularly well-presented results or interesting observations). [6 marks]

B. Robot Worker Experimentation

Starting with the code given in the Search Exercise 5 notebook create and investigate your own robot worker scenario. A relatively small modification of the given scenario is sufficient. Take care not to increase the complexity of the problem too much (If there are too many choices, the algorithms may fail to solve even apparently simple problems.)

Report sections and associated marks:

- B1.** Design and describe a robot worker scenario. [3 marks]
- B2.** Describe and implement a heuristic suitable for your scenario. [3 marks]
- B3.** Investigate algorithms and present results in table form. [3 marks]
- B4.** Give key findings as concise bullet points. [3 marks]

Extra Marks: Available for exceptional work going beyond the basic requirements (e.g. more complex scenarios, clever heuristics, interesting results). [6 marks]

[40 marks total]

Read on for further details, suggestions and tips.

What and How to Investigate

In order to obtain high marks, you will need to carry out a thorough investigation of search algorithms within the limitations of the search algorithm software with which you have been provided. You should aim to cover the following different types of search algorithm and options, as well as different levels of difficulty of the problem itself:

- Different basic search algorithms
 - breadth-first,
 - depth-first, with fixed action choice ordering,
 - depth-first, with randomised action choice ordering,
 - best first,
 - A*.
- Different heuristics for the informed search algorithms (best first and A*):
 - The misplaced tiles heuristic,
 - The Manhattan distance heuristic.

(Note: If you have coded these heuristics yourself and they seem to work well, you should use your own heuristic functions. If you have some trouble using your own heuristics you may use the ones defined in the `crazy8heuristics` module instead.)

- Whether or not loop checking is used.
- How many moves is the initial state away from the goal state. (Start by trying an initial state just a couple of moves away from the goal. Then try a moderately difficult case and a completely scrambled up case.)

Tips

- When experimenting with different search configurations, proper testing will require adjustment of the `max nodes` parameter. Start with a small number such as 5000. If the search attempt terminates quickly without finding the solution you should raise the number to see if a solution can be found with a higher limit. If you still do not get a solution raise the limit further until, either you get a solution, or the algorithm runs for too long for you to wait. You can decide your own time limitations. Around 1 to 5 minutes would be a reasonable time to wait.
- Try problems of different difficulty by altering the value of the 8-puzzle initial state or the robot workers goal. I suggest you concentrate on 2 or 3 case, but you may have to try a few possibilities to identify ones that correspond to widely different levels of difficulty. Your hardest problem should preferably be one that is not solvable without using one of the heuristic search algorithms.

- Systematically record the results of your tests in tables. Each row should correspond to a particular search configuration and the columns should give the path length, total nodes tested and time taken. Also, indicate whether the search was successful. In the case where no solution has been found, there will be no path found and the number of nodes tested will be determined by the maximum node limit.

In order to present results clearly, you may wish to use a separate tables for different problem cases and/or different sets of values. However, it is up to you to judge what is the clearest and most informative way to present your results.

- Do **not** try to cram every possible test and result number you can think of into your report. Beyond giving a reasonable coverage of possibilities, you will not gain extra marks for sheer volume of results. You need to present your results in a clear way and could actually lose marks if your results tables are cluttered and difficult to understand.

See next page for submission instructions.

Submission Instructions

The submission for this assignment will be in the form of a document in **PDF** format, to be uploaded to **Gradescope**.

Your answer for each of the eight parts should be presented on a separate page (for ease of marking). It is expected that answers for each part would fit on one page but you could go onto a second page if you feel you really need it (e.g. for results tables).

A PDF file illustrating the expected layout of your submission is provided and can be found the Assessment area of the module's Minerva pages. Note: this is just a guide for the layout, not an actual template file. You will need to create your own report file using the software of your choosing (e.g. Word or LaTeX/Overleaf). You will need to generate a PDF version of your file for upload to Gradescope.