F29AI Artificial Intelligence and Intelligent Agents
Coursework – Search, Knowledge Representation, and Planning

Handed out during Week 2. Coursework demos will take place during the Week 8 lab. Details to follow.

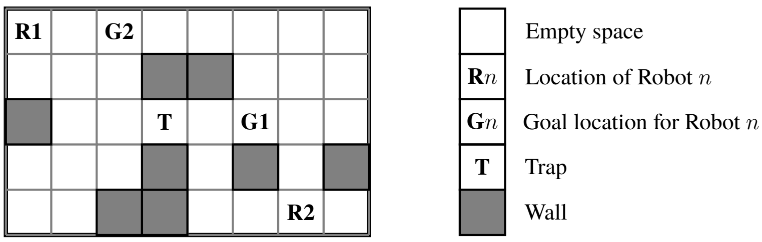
Please read these instructions **carefully**. The coursework will count for **22%** of your overall course assessment. You will work in **pairs** for the coursework. Group names will be collected during the Week 3 lab. Once assigned, groups cannot be changed. You will demonstrate your coursework during the Week 8 lab.

Part I: Search

In this part, you will write a program that uses A* search to find solutions to the scenario described below. You should work in C++ (or another appropriate language, see below). Starter code will be provided to guide you. You will be assessed on the quality and correctness of your program, and your understanding of the solution.

Problem description

Two parcel delivery robots must carry packages from their starting locations to specific delivery locations. Each robot can move independently of the other robot but the two robots cannot occupy the same space at the same time. The area in which the robots operate may contain walls which block their progress. The area may also contain traps which the robots *can* pass through but at an additional cost. The goal is to find paths for the two robots to move from their starting locations to their assigned delivery locations. For example:



Additional details about the scenario:

- The delivery area in which the robots operate is a grid of $N \times M$ spaces. Each robot occupies one space in the grid. The robots are blocked from moving outside the grid.
- A robot can move to any grid space immediately adjacent to it that is not occupied by another robot or a wall. A robot cannot move diagonally. Robots are permitted to move through traps.
- Moving to an adjacent space that is empty costs 1 unit of energy. Moving to a trap costs 5 units of energy.

What to implement

Write a program that uses A* search to solve the robot parcel delivery problem. You should design and implement an appropriate state space, transition function, cost function, and goal encoding that models the problem description. The design of the state space is particularly important for this problem. You should also make sure that your program displays the results of the search.

As part of your program, you should implement **two** different heuristic functions to be used with A* search. You should also be prepared to explain the choice and operation of these heuristics.

Test your implementation with a number of different grid sizes, robot locations, wall configurations, and trap positions. Your solution should be general enough that such conditions can be easily changed. A number of example configurations that your program should support will be distributed before the coursework is due.

You do not have to implement A* search yourself unless you wish to do so. Starter code will be distributed with a version of A* search implemented in C++. You are free to use this code and you should not need to modify it. If you would like to work in a different language (C, Java, or Python) you may do so but you are responsible for writing/converting/finding your own version of A* search. If you want to use a language that isn't listed above, please discuss this this with course instructor, the earlier the better.

Part II: Knowledge Representation and Planning

In this part, you will model two small planning domains in PDDL, using the STRIPS subset of the language. You should develop and test your domains using the planning tools available at `http://planning.domains/`. English descriptions of the domains, along with example planning problems are given below. You will be assessed on the quality and correctness of your domains, and your understanding of the knowledge representation.

(a) Parcel delivery revisited

A delivery robot can move parcels between the Earl Mountbatten, David Brewster, and Lyell buildings. Assume that the robot can carry an unlimited number of parcels and the only relevant locations are the three buildings. When travelling between Lyell and the other two buildings, the robot must go outside. In contrast, the robot can travel indoors between Earl Mountbatten and David Brewster, provided the door in the corridor between the buildings is open. If the door is closed, the robot can open the door by using one of its parcels to keep the door propped open as it passes through. If the robot goes outside when it's raining, it will get wet unless it has an umbrella. It can get an umbrella in the Earl Mountbatten building. Goals in this domain can specify requirements about the location of various parcels, the robot's location, and the robot staying dry. For example:

Initial state: the robot is in Earl Mountbatten, parcel1 is in David Brewster, it's raining, the robot is not wet, the umbrella is in Earl Mountbatten, and the robot is not carrying any parcel nor the umbrella.

Goal: parcel1 is in Lyell, the robot is not wet, and the robot is holding the umbrella.

(b) Mars exploration

Heriot-Watt's Space Agency needs to plan a mission for a group of robot probes that will be sent to Mars, as a prelude to human exploration of the planet. Two types of probes will be sent, a single *comsat* and a group of *rovers*. The comsat (communications satellite) remains in orbit and establishes a communication network, relaying messages between Earth and the rovers on the planet. The comsat is fixed in one of two *orientations*, facing Earth or facing Mars. When facing Earth, the comsat may send test results back to mission control; when facing Mars it may send or receive messages from a rover. Rovers explore the planet, perform tests, and send test results back to the comsat. Rovers may move between *test sites* on the planet—locations that have been identified as areas of scientific importance. At a test site, a rover may perform a *soil composition* test. A rover will only perform a test if it receives a message from a comsat instructing it to do so. Rovers are always monitoring comsat signals and immediately receive any messages that the comsat might send it. However, rovers must explicitly send messages back to the comsat. Due to memory limitations in the rovers, if a rover has performed a test it cannot perform another test until it has sent the results to the comsat. Goals in this domain will typically require that the results of tests at particular sites be sent back to mission control. For example:

Initial state: rover1 is at site1, comsat is facing earth, rover1 has no test results in its memory.

Goal: mission control has received the results of a soil composition test at site2.

What to implement

- You should encode each of the above domains in PDDL by defining the predicates, objects, and actions that are needed to describe the domains. Your domain should be general enough to deal with any collection of objects (parcels, buildings, test sites, rovers, etc.). To do so, you should think about the information that must be supplied for each *type* of object. Test cases will specify precise collections of objects.
- You should test your domains on a *set* of planning problems (an example is given for each domain) to ensure they are working properly. Additional example planning problems will be distributed before the coursework is due.

Note that the domains are described at a high level. This may give you the impression that the domains are not described precisely enough for you to encode them, but this isn't the case. Try to decipher what is relevant and make decisions to design an appropriate encoding. There are many good encodings.

Master's students only

In addition to the problems in Parts I and II, Master's students should also answer the following questions.

1. Extend the Mars exploration domain in Part II to enable rovers to perform a second type of test, a *mineral analysis* test. While a single rover can perform a soil composition test, a mineral analysis test is a joint test, requiring two rovers to be at the same test site. Goals in this domain will also be extended to specify the types of test results that should be sent back to mission control for particular sites.
2. How well does the Mars exploration domain scale? Choose a planning problem and increase the number of rovers. Does this have an effect on the planning times or the plan length? Does the planner ever fail to generate a plan? Repeat the experiment and increase the number of test results. Write a short report (2 pages maximum) describing your results. Use tables and graphs to illustrate the data you collected. Speculate on how robust your domain is and how you might improve it to scale to larger problem instances.

Assessment

Your coursework will count for 22% of your overall course assessment. Your assessment will be based on a demonstration of the work you have done, the correctness of your solutions, and the quality of your code and domain descriptions. Master's students will additionally be assessed on their written report.

Demonstrations

Each student pair will demonstrate their search program from Part I and the two planning domains from Part II during the Week 8 lab. A time slot will be allocated to you in advance of this lab. Although you are working in pairs, you will each be asked questions during your demo to test your knowledge of what you have implemented. You should think about how you are dividing up the work and understand the work that your partner has done.

If you have specific problems that prevent you from doing your demonstration as scheduled, please let the course instructor know **in advance**. Any demos not completed at the end of the demo session will receive no marks unless a different demo slot has been previously agreed.

What to hand in

In addition to your demonstration, all student pairs should hand in their code from Part I and their two planning domains from Part II **before** their demonstration. Master's students should additionally hand in their reports before the demonstration.

Mark distribution

Task	Undergraduates	Master's students
Part I		
Code handed in	2	2
Demo	8	6
Part II		
Domains handed in	2	2
Demo	10	7
Master's student report		5
TOTAL	**22**	**22**