INF2D Coursework 1: CSPs and A* Search

Ameer Saadat-Yazdi, Nadin Kökciyan

January 23, 2023

1 Introduction

The aim of this assignment is to familiarise you with Constraint Satisfaction Problems (CSPs) and the basic algorithms that are used to solve them, as well as A* search. Submission details can be found in Section 2. There will be a lecture discussing this assignment on Friday 27th of January 2023 at the standard lecture time and venue. There will also be weekly drop-in labs on Fridays 11am-1pm (weeks 4-9), in Appleton Tower 6.06, where a demonstrator will be available to answer your questions and provide appropriate help. The assignment is divided into two parts:

- In Part A, you are asked to represent an airplane seat allocation problem as a CSP and use the Backtracking algorithm to find a solution (50%).
- In Part B, you will use A* search to find a path for passing a set of forms to a number of passengers (50%).

2 Submission Instructions

For this coursework, you will have to submit **a single PDF file** with your answers. Please ensure that each question is answered on a separate page to make marking easier.

Your submission must be uploaded to GradeScope via the link given in Learn (Assessment > Assignment Submission > Coursework 1 > Coursework 1: Constraint Satisfaction Problems and Search). You can simply drag and drop your pdf file into the submission window. After uploading your file, take the time to associate each question to the relevant pages in your PDF, this will significantly speed up marking and ensure that everyone gets feedback on time. **Your file must be submitted by 12pm on the 7th of March 2023.** You can submit more than once up until the submission deadline. All submissions are timestamped automatically. We will mark the latest submission that comes in before the deadline.

Late submission policy: Gradescope will accept late submissions up to 7 days past the on-time deadline. Do not submit (or re-submit) after the on-time deadline unless you have an approved extension or are prepared to accept a late penalty. If you have been approved to submit more than 7 days late (this is only possible if you have both an extension and a learning adjustment), please notify the lecturer in charge of the assignment, and we will tell you how to submit. If you do not submit anything before the deadline, a late penalty will be applied to this submission, unless you have received an approved extension.

Do not email any course staff directly about extension requests. For information about late penalties and extension requests, see: `http://web.inf.ed.ac.uk/infweb/student-services/ito/admin/coursework-projects/late-coursework-extension-requests`.

Good Scholarly Practice: Please remember the University requirement as regards all assessed work for credit. Details about this can be found at:

- `https://www.ed.ac.uk/academic-services/staff/discipline/academic-misconduct`
- `http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct`

Furthermore, you are required to take reasonable measures to protect your assessed work from unauthorised access. For example, if you put any such work on a public repository then you must set access permissions appropriately (generally permitting access only to yourself, or your group in the case of group practicals).

3 Important Information

You should submit **only one PDF file** to Learn containing written solutions to each question. Each question should be answered on a separate page. **The deadline for this coursework is 12pm on the 7th of March 2023**. Note that for questions requiring you to "use" or work through an algorithm, you should show working out and present your steps on paper. You may wish to implement the algorithms in a programming language of your choice to verify your results, however, only the written solutions in the PDF will be marked.

There are the following sources of help:

- Attend lab sessions: Fridays 11am-1pm (weeks 4-9), in Appleton Tower 6.06
- Read "Artificial Intelligence: A Modern Approach" Third Edition, Russell & Norvig, Prentice Hall, 2010(R&N) Chapter 6 or Pearson New International Edition, Pearson, 2014 (NIE) Chapter 7.
- Ask any questions on Piazza/during lab sessions.

4 Part A: Constraint Satisfaction Problems (CSPs) (50%)

In this section, you will be asked to find a seat for 18 passengers on a plane based on a number of constraints. The plane layout for this part is depicted in Figure 1.

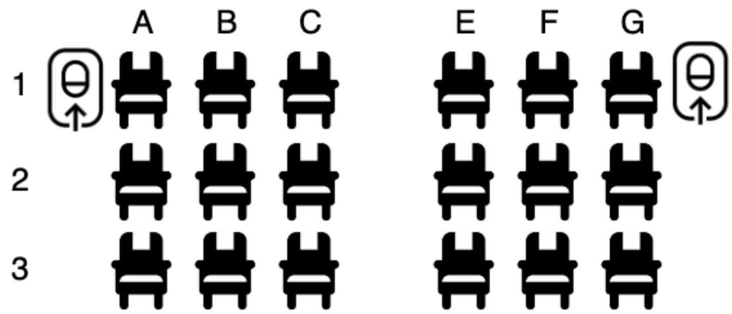

Figure 1: Plane layout for Part A

1. A flight is fully booked from Edinburgh to Kirkwall with 18 seats as shown above. The passengers are as follows:

   - 1 family (consisting of 1 young couple and 3 children)
   - 3 young couples
   - 1 elderly couple
   - 1 wheelchair user
   - 2 elderly individuals
   - 2 young individuals

   There are also a number of rules on how individuals may be seated based on their age and requirements.

   [R1] Children must be sat immediately next to at least one parent. They cannot be separated by another passenger or the aisle.

   [R2] There are emergency doors on the first row; these seats cannot be occupied by children, elderly passengers, or passengers with physical disabilities.

   [R3] Couples without children must be sat immediately next to each other.

   [R4] Families must be assigned to the same row.

   [R5] Passengers requiring a wheelchair must be assigned to an aisle seat.

   [R6] The elderly individual passengers have both requested window seats.

   [R7] The elderly couple has requested to be sat on the third row.

   [R8] One member of the elderly couple has requested a window seat.

   (a) (5 points) Write down the variables and the domain (before applying any of the constraints) to translate this into a CSP problem.

   (b) (5 points) Of the above rules which ones can be represented as unary constraints? For each variable, write down the domain after applying the constraints.

   (c) (6 points) To define the constraints, write down pseudocode for functions that take a pair of seats and return a boolean value for the following relations **same_row** and **sat_adjacent**.

(d) (8 points) The constraint associated with one of the above rules contains a ternary relation. Identify this rule and define the ternary relation using the relations defined above. By introducing a new variable convert the ternary relation into a set of binary relations.

(e) (6 points) Write down the list of binary constraints and draw the corresponding constraint graph.

(f) (12 points) Use backtracking search with the Minimum-Remaining Values heuristic to find a solution. The domain values should be ordered row-wise and then aisle-wise. The search tree should be presented as in Figure 2. You should also show any additional working outs.

(g) (8 points) Discuss the pros and cons of using the Minimum-Remaining Values heuristic compared with the Degree heuristic combined with Least Constraining Values for this problem. Which would you choose for future applications to a similar problem?

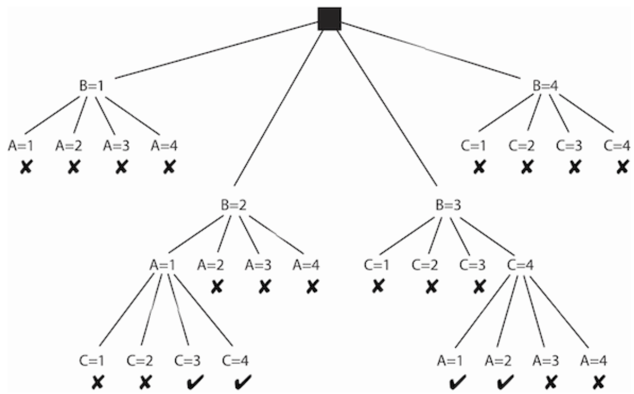

Figure 2: Example CSP search tree taken from `https://artint.info/html/ArtInt_78.html`. Suppose you have a CSP with the variables A, B, and C, each with domain {1,2,3,4}. Suppose the constraints are A<B and B<C. A possible search tree is shown above.

5 Part B: Search (50%)

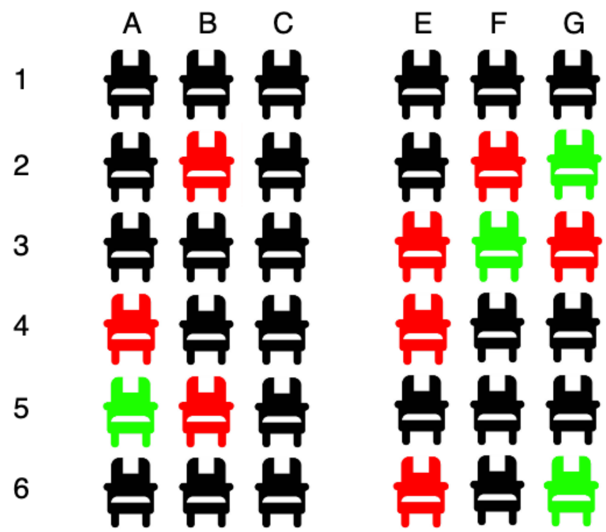

Figure 3: Plane layout for Part B. Red seats indicate sleeping passengers while green seats indicate the passengers who need to receive forms.

2. In this question you will be asked to find an efficient path to get a set of forms from the passenger seated in A1 to the passengers seated in green seats indicated in Figure 3. The forms can only be passed ahead, behind, left, or right, and not diagonally. In this problem, we will consider the distance between aisles C and E as double the distance between aisles B and C as the passengers.

   (a) (4 points) Define an appropriate measure for the distance between two passengers.

   (b) (6 points) A^* is designed to find a path from a start state to a goal, in this case, we have multiple goals. Write pseudocode to define an algorithm that uses the above metric to find a suitable order in which to achieve the goals. Your algorithm should ignore the sleeping passengers.

   (c) (10 points) Use the algorithm you defined in the previous part to find the order in which the goals should be met.

   (d) (12 points) Use A^* search to find a path from 1A to each of the goals in the order you determined in the previous part. Assume that the passengers in red seats are asleep and cannot be passed the forms. Write down the final set of paths, the nodes searched in each search run, and the final frontier as well as any additional working.

   (e) (4 points) What are some of the limitations of the approach outlined in parts (b)-(d). Can you suggest some potential improvements?

   (f) (6 points) Define an algorithm in pseudocode based on the suggestions in the previous parts.

   (g) (8 points) Compare how the two approaches would perform with different configurations of the same problem, i.e. with different goals and sleeping passengers. [***Hint****:* *A high level comparison is sufficient. You don't need to provide a search tree for each configuration.*]