Assignment 2 - Solution

CPSC 322 2016, Term 2

Due: Friday, Feb 10, 7pm

NOTE: For this assignment, you can only use 2 late days; submissions received after Sunday, Feb12, 7pm will not be accepted. This is so that we can give out the solutions to help you study for the midterm.

Name (Last, First):

Student ID:

Partner's Name (if worked with someone else):

Late Days:

Please include the above information at the top of your first page; failure to do so will lead to a deduction of marks.

Assignments are to be submitted via **Connect** by the specified deadline. **We do not accept assignments sent via email or handed in in-person.** Only PDF files will be accepted and marked.

Late Assignments
Every student is allotted **four** "late days" during the term, which allow assignments to be handed in late without penalty on four days or parts of days during the term. **If you want to use some of your late days** for this assignment (max 2 as described above), **write the number of late days in the space provided above.**

If an assignment is submitted late and a student has used up all of her/his late days, **or if late days are not specified,** 20% will be deducted for every day the assignment is late. For example, an assignment that is 2 days late and graded out of 100 points will be awarded a maximum of 60 points.

Collaboration
For this assignment, you may collaborate with *at most* one other student from this class. If you are working with someone else, you must submit **a single assignment for your pair**, writing the names and the student IDs for both of you at the top of your first page.

You may not, under any circumstances, submit any solution not written by yourself or your partner, or look at solutions of students not in your team or solutions from other sources. You may not share your own work with others outside your team.

Submission File Naming
Please name the submission PDF file with your student ID: number.pdf, (e.g., `12345678.pdf`), for solo submission or student IDs of both you and your partner for pair submission: number1-number2.pdf, (e.g., `12345678-23456789.pdf`).

Make sure that in your solution you clearly indicate the exact section you are answering.

Failure to follow all instructions above will result in a substantial deduction of marks.

- Please try to start answering each question in a new page.

Question 1 [21 points] Allocating Developments Problem

CSP techniques are useful in solving complex configuration and allocation problems. You are given the task of allocating four developments in a new site in Whistler. You have to place a housing complex, a big hotel, a recreational area and a garbage dump. The area for development can be represented as 3x3 grid (three rows 0,1,2 and three columns 0,1,2) and you need to place each development in one cell of the grid. Unfortunately there are some practical constraints on the problem that you need to take into account. In the following, A is close to B if A is in a cell that shares an edge with B.

- There is a cemetery in cell 0,0.
- There is a lake in cell 1,2.
- The housing complex and the big hotel should not be close to the cemetery.
- The recreational area should be close to the lake.
- The housing complex and the big hotel should be close to the recreational area.
- The housing complex and the big hotel should not be close to the garbage dump.

a) [16 points] Represent this problem as a CSP. Be as precise as you can in specifying the constraints. Also do not forget some basic constraints that are inherent in allocating objects in space but are not listed above.

b) [5 points] Draw a constraint graph for this problem. If a constraint/domain is too long to fit easily in the graph, use a label in the graph instead, and indicate which constraint/domain the label refers to.

Question 2 [40 points] CSP - Search

Consider a scheduling problem, where there are eight variables A, B, C, D, E, F, G, H each with domain {1, 2, 3, 4}. Suppose the constraints are: $A > G$, $A \leq H$, $|F - B| = 1$, $G < H$, $|G - C| = 1$, $|H - C|$ is even, $H \neq D$, $D > G$, $D \neq C$, $E \neq C$, $E < D - 1$, $E \neq H - 2$, $G \neq F$, $H \neq F$, $C \neq F$, $D \neq F$, $|E - F|$ is odd.

An AIspace representation for this graph named "as2csp.xml" is available in Connect. (To load this file in the applets, go to the **File** menu and select **Load from File**.)

a) [25 points] Show how search can be used to solve this problem, using the variable ordering A, B, C, D, E, F, G, H. To do this you should write a program to **draw the search tree generated, report all answers (models) found and the number of failing consistency checks**. You can use whatever programming language you like.

To indicate the search tree, write it in text form with each branch on one line. For example, suppose we had variables X, Y and Z with domains {t, f} and constraints X != Y, Y != Z. The corresponding search tree, with the order X, Y, Z, can be written as:

```
X=t Y=t failure
    Y=f Z=t solution
        Z=f failure
X=f Y=t Z=t failure
        Z=f solution
    Y=f failure
```

Submit the printout of your tree and your search code as appendices at the end of your submission, but indicate the model assignments found and the number of failures in your answer to the question.

b) [10 points] Is it possible to generate a smaller tree? Come up with a simple variable selection heuristic that results in as small a tree as you can find, and report the following:

- Your variable selection heuristic
- A variable ordering that you obtain using this heuristic
- How many failing consistency checks there are for the tree obtained from this variable ordering.

c) [5 points] Explain why you expect the tree resulting from this heuristic to be good. (A good explanation as to why your ordering is expected to be good is more important than finding the best ordering).

Question 3 (16 points) CSP – Arc Consistency

a) [6 points] Show how arc consistency can be used to solve the scheduling problem in Question 2. You can use AISpace and the as2csp.xml file provided in Connect. You need to:

- Show the initial constraint graph.
- For the first 4 steps of arc consistency show which elements of a domain are deleted at each step, and which arc is responsible for removing the element.
- Show explicitly the constraint graph after arc consistency has stopped.

b) [5 points] Use domain splitting to solve this problem. Draw your tree of splits and show the solutions.

c) [5 points] Constraint satisfaction problems can become extremely large and complex. Given the choice between DFS with pruning and arc consistency with domain splitting, which would you use in such cases, and why?

Question 4 (33 points) CSP – Stochastic Local Search

Show how stochastic local search can be used for the scheduling problem in Question 2.

AISpace hints:

- Open the **Algorithm Options** dialog, and set the search method to **Greedy Descent with All Options**. All of the settings you will need, such as variable/value selection methods, can be set here.
- Unless otherwise specified, make sure that SLS tries 2000 steps before terminating.
- Unless otherwise specified, prevent random resets from occurring.

a) [5 points] For one particular run, make SLS select any variable that is involved in an unsatisfied constraint, and select a value that results in the minimum number of unsatisfied constraints. Explain which element is changed at each step and what was the resulting number of unsatisfied arcs. (You only need to do this for 5 steps).

b) [20 points] Compare and explain the result of the following settings:

i. **[4 points]** Select a variable involved in the maximum number of unsatisfied constraints, and the best value.
ii. **[4 points]** Select any variable that is involved in unsatisfied constraints, and the best value.
iii. **[4 points]** Select a variable at random, and the best value.
iv. **[8 points]** A probabilistic mix of i. and ii.. Try a few probabilities and report on the best one found.

c) [4 points] How important is it to choose the value that results in the fewest unsatisfied constraints as opposed to choosing a value at random? Justify your answer with evidence.

d) [4 points] For the best variable/best value method from part (b(i)), allow random resets (for example, after 50 steps). Explain how this affects the performance of the algorithm, and why.