Assignment Four: CSPs

Due: 11:59pm, Monday 12 October 2020.

Solving following problems requires using the `http://AIspace.org` applets and/or the Python code at `http://aipython.org/aipython_322_as3.zip`. You will need to download the AIspace applets to use them. An AIPython representation of the scheduling problem from question 2 of assignment 3 is in the zip file, and AISpace representation is at: `https://www.cs.ubc.ca/~poole/cs322/2020/as3/as3csp.xml` (copy this url into "File" ⟶ "Load from URL" in the applets.)

This can be done in groups of size 1, 2 or 3. Working alone is not recommended. All members of the group need to be able to explain the group's answer.

Submit your answers using Canvas. Use proper sentences (not note form) in your answer.

Ask questions on Canvas discussion board. Feel free to answer them too.

Question One

Show how arc consistency can be used to solve the scheduling problem of question 2 of assignment 3. To do this you need to

(a) For the first 5 instances of arc consistency, explain which elements of a domain are deleted at each step, and which arc is responsible for removing the element. You need to explain it at a level for one of your peers to understand if this is the first time they have seen arc consistency. Select arcs to cover the interesting cases.

(b) Show explicitly the constraint graph after arc consistency has stopped.

(c) Show how splitting domains can be used to solve this problem. Draw the tree of splits and show the solutions.

(d) Based on this experience, discuss how much arc consistency saves over the backtracking for this problem (include the best tree that you found and an arbitrary tree). Make sure there is something quantitative in your answer and there is evidence for your claims.

Question Two

Show how stochastic local search can be used for the scheduling problem of question one. Make sure that it tries 2000 steps before termination. You can use the AIspace applet or the Python code.

(a) For one particular run, where you select a variable that is involved in the most conflicts, and either select a value at random or that results in the minimum number of conflicts, 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). Again, explain this at a level for one of your peers to understand if this is the first time they have seen local search. (Try to make the example interesting as if you were explaining it.)

(b) Compare and explain the result of the following settings:

  i) select a variable involved in the maximum number of unsatisfied constraints, and either its best value or a random value (but tell us which one)

ii) select any variable which is involved in unsatisfied constraints, and either its best value or a random value

iii) a probabilistic mix of the (i) and (ii), such as with probability 0.4 select a variable involved in the most conflicts, otherwise select variable in any conflict, and either select its best value or a random value. Try a few probabilities and report on the best one found.

You must show and explain the runtime distributions and describe what you observe (in particular, tell us which curves correspond to which settings, when one algorithm better than another, and how often each setting does not find solutions; be specific). You should use multiple runs for each setting in the comparisons.

(c) Does a random restart help for any of these cases? Explain.

(d) Based on this experience suggest good settings for the parameters that control which algorithms is run (e.g., which method is chosen, and the probabilities that different choices are made). Justify your choice.

Question Three

For each question, specify how long you spend on it, and what you learned. How was the work in the team allocated? Was the question reasonable? (This questions is worth marks, so please do it!)