CS 6360: Advanced AI
Assignment 3

Due: 03/14/2017, at midnight

General Instructions:

If anything is ambiguous or unclear:

1. **Discuss possible interpretations with other students, your TA, and instructor**
2. **Make assumptions, state them explicitly, and then use them to work out the problem**
3. **Use Piazza for discussions among yourselves, and also for questions, also for questions and clarifications you need from the instructor and the TA. Piazza levels the playing field because the responses to questions asked are of interest to all, and shared by all. If you have very specific questions, you may email the TA**

Remember that after general discussions with others, you are required to work out the problems by yourself. All submitted work must be your own work. Please refer to the Honor code for clarifications.

Constraint Satisfaction Problems (CSPs)

Programming Assignment. (100 points)

For this assignment you will generate random instances of the map coloring problem as follows: scatter n points in a unit square; select a point X at random, connect X by a staright line to its nearest point Y such that X is not already connected to Y, and the line crosses no other line. Repeat the above steps till no more connections are possible. The points represent regions on a map, and the lines represent connected neighbors.

Your assignment is to find the k-colorings for each map, for values of $k = 3$ and $k = 4$.

You will implement the following algorithms:

1. Backtracking along with the MRV, degree, and LCV heuristics
2. Backtracking with the Maintaining Arc Consistency (MAC) algorithm. The MAC procedure is explained on page 218 of Russell and Norvig, chapter 6 (also see the section on Full Look Ahead (page 31) of the Bartak report). After a variable, X_i is assigned a value, the arc-consistency algorithm (either AC-3 or AC-4) is invoked for all arcs (X_j, X_i) (all neighbors of X_i) whose values are unassigned.
3. Backtracking with conflict-directed backjumping.

Along with the implementation, you will run a series of experiments, and report your results as a table of average run times for each algorithm, starting with values of $n = 20$, up to the largest that you can manage. You will then write a report discussing your implementation of the algorithms, their time and memory requirements, and a discussion of the results that you have obtained.

Submission: You will submit your code (source and executable), you documentation, and your report that includes the results of your experiments as zipped file on Blackboard.