CSCI 446 — Artificial Intelligence

Project #1
Search, Constraint Satisfaction, and Graph Coloring

The purpose of this assignment is to give you an introduction to artificial intelligence from the perspective of search. Being that this is a team project, several elements will need to be implemented. This project is a variation and extension of the problem in the textbook (Russell & Norvig, 6.10).

For this project, you need to write a "problem generator" whereby you will create several graphs of various sizes to solve instances of the graph-coloring problem. For your graph generator, scatter n points on the unit square (where n will be provided as input), select some point X at random and connect X by a straight line to the nearest point Y such that X is not already connected to Y and line crosses no other line. Repeat the previous step until no more connections are possible. The points represent regions on the map, and the lines connect neighbors.

After building the graph, try to find k-colorings of each map for both $k = 3$ and $k = 4$ using min-conflicts, backtracking, backtracking with forward checking, and backtracking with MAC. You will also attempt to find such colorings using a genetic algorithm with tournament selection and a fitness function with a penalty term.

Here are the specific steps that need to be followed:

- Implement the graph generator.
- Generate at least 10 different graphs of sizes ranging from 10 to 100 vertices in steps of 10 (i.e., 10, 20, 30, 40, 50, 60, 70, 80, 90, 100).
- Implement a constraint solver to find 3-colorings and 4-colorings for each graph (if they exist), implementing the following variations:
  - Min conflicts
  - Simple backtracking
  - Backtracking with forward checking
  - Backtracking with constraint propagation (MAC)
  - Local search using a genetic algorithm with a repair function and tournament selection
- Write a design document that outlines the architecture of your software (a UML class diagram is encouraged but not required), design decisions made in implementing your algorithms, and the experimental design for testing the results.
- Run experiments, keeping track of the main decisions being made for each algorithm. The count of these decisions will be used to gauge run time since CPU time and wall clock time are not reliable estimators.
- Write a paper that incorporates the following elements, summarizing the results of your experiments. Make sure you explain the experimental setup, the tuning process, and the final parameters used for each algorithm.
  1. Title and author names
  2. A brief, one paragraph abstract summarizing the results of the experiments
  3. Problem statement, including hypothesis (how do you expect the algorithms to do?)
  4. Description of algorithms implemented
  5. Description of your experimental approach
  6. Presentation of the results of your experiments
  7. A discussion of the behavior of your algorithms, combined with any conclusions you can draw
  8. Summary

9. References (you should have at least one reference related to each of the algorithms implemented, a reference to the data sources, and any other references you consider to be relevant)

- Submit your fully documented code for the data converter, results of the runs of each algorithm, your design document, and your paper.

Due Dates:

- Design Document – September 16, 2016
- Program Code and Sample Runs – September 23, 2016
- Project Report – September 26, 2016