Stephen Chambers
CS830
October 28, 2015

Project Proposal

The Chemistry department needs to take approximately 300 students and place them into study groups. The time of these study groups is based upon both the student's availability and the department TA's availability. Each student submits times that work for them, times that they could make work if needed, and times that do not work for them. The department then manually places all students into study groups, taking into account their time choices. This is an interesting CSP because of the massive state space. If 300 students need to be placed into ten study groups, the state space is a massive 300^{10} placement options.

Each solver takes in the same input format consisting of student info, class info, and the objective function. The student info is populated from student responses from a survey sent out by the Chemsitry department. The class info is set by the Chemistry department when running the program. It contains a list of TA's, their email, and the times of possible study groups. Finally, all search algorithms will try to optimize an objective function, which is a function that approximates the cost of a placement based on customer requirements. For example, there could be a penalty for not maintaining a certain gender ratio, not having a bell curve of GPAs, being over a maximum group limit, etc. The evaluation of a placement will be the cost received from running the objective function over the placement. The two main families of algorithms that will be used are tree search and local search. The tree search will utilize variable and value heuristics, as well as bounding on the objective function cost. The local search will start with a random placement of students, and modify that placement in place iteratively until a solution is found.

The program will output results in a programattically readable format. Another part of the project will be to take this output and make it human readable, resembling something like the following:

Study Group 1		
TA:	Mary Foo	
Email:	foo@gmail.com	
Time:	12:40-2:00	
John Bar	Stephen Chambers	Prof Ruml
Michaela Tremblay	Bence Cserna	John Cena

Pair programming is being utilized for this project to make sure the code is of high quality and maintainable. Michaela and I are working on tree search algorithms, while John and Joel are working on the local search. Additionally, a senior project group is working with us by creating a validator and an input converter. The input converter takes in different mediums of input, such as a csv file from blackboard or a result from Google forms, and converts it into a general format that is readable from both solvers.