Luke Newman

ICS 340 Algorithms and Data Structures

Local Search Report

April 20, 2018

I used a variation of two stage and any conflict to implement the local search. After my initial random assignment, I store the courses in a Max Heap priority queue. It uses the number of conflicts or constraint violations for a course as the key. In the two stage strategy a variable is chosen and then a value for that variable is chosen. I take the course with the maximum number of conflicts and seek to improve its semester assignment to the optimum value. If the course's semester assignment is already optimum meaning any other assignment will not improve the heuristic value of the total assignment then I choose any course that is involved in a constraint violation at random instead. I try to improve its semester assignment in the same way. If this course also has a current optimum semester assignment, then I again pick a random course involved in a constraint violation and assign it a random semester value 1 - 7.

I found that this algorithm will solve the problem in less than 150 iterations about 40% of the time. After 150 iterations I'm better off restarting with a new random sample and trying again in the same manner. I basically just hope for a better hand dealt to me with a random restart. The algorithm has solved the problem in as little as 16 assignments, but it has also taken as much as 1000 assignments to solve. On average it takes about 300 to 400 iterations.