# Course Scheduling as a Constraint Satisfaction Problem

**Micah Brown** and **Ben Wiley**
{msbrown,bewiley}@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

**Abstract**

The current system for selecting courses at Davidson College is known as "Webtree". Webtree selects courses for students, in order of class seniority, in a random order within their class. This outdated system has been in place for decades, and is very static. We attempt to redesign Webtree such that it selects courses for students in a more efficient way. By framing the problem as a constraint satisfaction problem, we have created a more optimal solution to assigning courses than the current system.

## 1 Introduction

With over 400 courses offered every semester at Davidson College, assigning students courses based on their preferences is not a trivial task. For years, the Office of the Registrar has used "Webtree", a system designed to be fair yet give students preferential treatment according to seniority. Webtree, untouched in the past several decades, remains a constant source of confusion, and later in the course selection process, frustration. At first, students are overwhelmed by the non-intuitive user interface. Once Webtree has run, students are then disappointed by the courses they receive. Due to the static nature of Webtree, it is not uncommon for students to receive only one or two courses after an entire run of Webtree (with four courses being a full course-load).

Course scheduling is a common problem for artificial intelligence researchers and much work has gone into efficient ways to assign courses. However, Davidson College's Webtree is a unique problem that has not been studied yet, to our knowledge. Much of the prior research into course scheduling involves placing courses in appropriate classrooms and at appropriate times. Davidson College's course selection is unique because course ceilings are strictly enforced, to ensure that class enrollment sizes are kept small to give students the necessary attention to foster a positive learning environment. Additionally, room selection is rarely an issue at Davidson College because very few classes exceed an enrollment of 30 students, and so most classes can fit into most rooms.

Recently, researchers have been framing the university course selection problem as a constraint satisfaction problem (Sampson, Freeland, and Weiss 1995). Constraint satisfaction problems maximize or minimize an objective function over a set of constraints with variables in a given domain. By setting constraints as actual limitations that exist, such a course must obey the given course ceiling, the course scheduling process can be made equitable and processed quickly.

The rest of the paper is organized...

## 2 Background

The current system of course selection, Webtree, was designed to allow students to customize their course selections to deal with various contingencies. For example, a student might want a different second course option depending on whether or not he or she successfulling recieved his or her first option. Webtree consists of three binary trees (depth of 3), which are traversed according to an elaborate algorithm. Students' fourth class is ranked in order $1-4$, and is selected according to availability in that order. For a visual represention of Webtree, as well a chance to imagine the confusion that many students feel when filling it out, see Figure 1. Interested readers are referred to the Davidson College Registrar's Office for a detailed description of the Webtree algorithm[1].

Alternatively, we chose to approach this problem as a constraint satisfaction problem. A constraint satisfaction problem takes the inputs *Varibles*, *Domains*, and *Constraints* (Russell and Norvig 2003). We chose to specifically implement this problem as a psuedoboolean constraint satisfaction problem. A psuedoboolean problem has its domain limited to True, False, or 0,1. Thus, we set up our problem as a set of constraints of inequalities, and an objective function that we attempt to minimize. Several computer scientists have devoted time to creating psuedoboolean solvers and allowing the public to have full access to them.

[1] http://www.davidson.edu/offices/registrar/course-registration-and-webtree/how-to-use-webtree

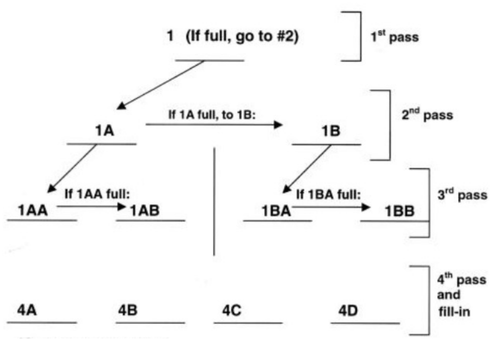


Figure 1: The complicated nature of Davidson College's Webtree. Image courtesy of Davidson College Registrar.

To determine whether one method of scheduling is "better" than another, we must devise a way of ranking a schedule. Since we are given preferences in Webtree format, we must be able to assign an individual request a certain rank. For our purposes, we have defined rank of a scheduling assignment to be the sum of ranks of all the courses assigned in that given assignment:

$$Rank_{scheduling} = \sum_{i=0}^{|Variables|} rank(i) * x_i \tag{1}$$

A course is assigned a given rank depending on the tree and branch that it was placed in the original Webtree. If a request was in the first "branch" (i.e. the root node of a Tree), then the course is given a rank 1-4, according to which Tree it is in. For Trees 1-3, the rest of the courses are then ranked according to whether they are a left child or a right child. Left children are a higher priority than right children and given a rank of 1; right children are the second option and thus given a rank of 2. Requests on Tree 4 are ranked 4-7, in the order that they are placed.

## 3 Experiments

Each variable $x_i$ represented one full Webtree request by a student. Thus, setting that variable true indicates that that course is granted to that particular student. Our constraint set consisted of four subsets: ensuring that each student would not receive the same course (of any and all sections) more than once, ensuring that each student would not receive more than four courses, ensuring that enrollment for each course fell under or equal to the course ceiling, and ensuring that each student received $n$ number of courses. The last constraint was varied from 0-4, which allows us to treat it as a "soft constraint", as setting $n = 0$ is similar to saying that the constraint does not need to be followed.

To create our minimization function, we subtracted our rank from 8 (so that a score of 7 was now the maximum), then negated. This was done because we want the best courses to minimize the function the greatest.

We used Sat4j's psuedoboolean solver to solve this constraint satisfaction problem, a free open-source java implementation (Berre et al. 2010).

## 4 Results

| | Min. courses | Avg. rank | Avg. courses granted | Students with 4 courses |
|---|---|---|---|---|
| Webtree | N/A | -6.30 | 3.74 | 1396 |
| Constraint Satisfaction Problem | 0 | -6.47 | 3.72 | 1440 |
| Constraint Satisfaction Problem | 2 | -6.46 | 3.72 | 1427 |
| Constraint Satisfaction Problem | 3 | -6.43 | 3.75 | 1354 |

## 5 Conclusions

This study sought to establish a novel method for assigning courses at Davidson College by framing the problem as a constraint satisfaction problem. After a comparative analysis of the old system of Webtree and our new method, using a psuedoboolean SAT solver, we conclude that our solution assigns courses in a manner that gives more students all four courses (i.e. a full course-load), as well as three courses. While the old Webtree algorithm assigns a higher average of courses to each student (3.735) than our approach without minimum courses as a constraint (3.716), adding a constraint that requires each student to receive a minimum of 3 courses was found to result assigning the maximum average number of courses to each student (3.746). Additionally, while not adding constraints that require a minimum number of courses to be fulfilled per student resulted in a lower average courses per student, students can easily add more courses after the algorithm has run during the add/drop period.

Future research should examine...

## 6 Acknowledgements

Thanks to Jackson Spell for his tip on using pseudobooleans, as well as his guidance on interpreting the output of the Sat4j solver.

## References

Berre, D. L.; Parrain, A.; Lille, U.; and France, N. 2010. The sat4j library, release 2.2 system description. *Journal on Satisfiability, Boolean Modeling and Computation* 7:pp. 59–64.

Russell, S. J., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education.

Sampson, S. E.; Freeland, J. R.; and Weiss, E. N. 1995. Class scheduling to maximize participant satisfaction. *Interfaces* 25(3):pp. 30–41.