Course Scheduling as a Pseudoboolean 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 This outdated system has been in place for decades, and is inflexible—always following the same iterative method of assigning courses to students. We attempt to redesign Webtree such that it selects courses for students in a more efficient and flexible 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 yet been studied, 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 as courses must obey the given course ceiling, the course scheduling process can be made equitable and processed quickly.

The rest of the paper is organized beginning with a background section explaining Webtree as well as the type of pseudoboolean constraint problem we use to solve Webtree more efficiently. Following that, our experiments section details our pseudoboolean constraints and minimization function, as well as the process by which we ranked course requests for maximum student satisfaction. Our results section shows how request fulfillment and satisfaction vary across the different constraint schemas we tested, and finally, we conclude with recommendations on further research.

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 successfully received his or her first option. Webtree consists of three binary trees (of depth 3), which are traversed according to an elaborate algorithm. A student's fourth class is ranked in order, 1 – 4, and is selected according to availability in that order. For a visual representation 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 choose to approach this problem as a constraint satisfaction problem. A constraint sat-

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

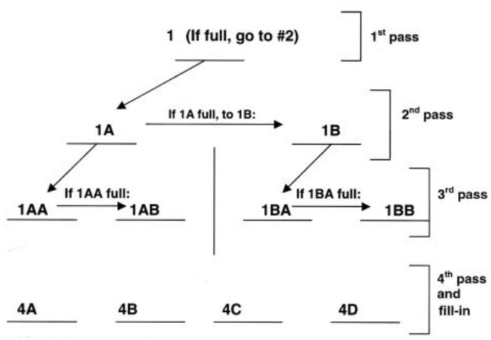

Figure 1: The complicated nature of Davidson College's Webtree. Image courtesy of Davidson College Registrar.

isfaction problem takes the inputs *Varibles*, *Domains*, and *Constraints* (Russell and Norvig 2003). We choose to specifically implement this problem as a pseudo-boolean constraint satisfaction problem. A pseudo-boolean problem has its domain limited to True, False, or 0,1. An example of a pseudoboolean constraint is the following:

$$x_1 + x_2 + x_7 + x_8 \geq 2 \tag{1}$$

In this example, no less than two of the variables x_1, x_2, x_7 and x_8 must be set to true in order for the constraint to be satisfied.

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 pseudoboolean solvers and allowing the public to have full access to them.

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:

$$Rank_{scheduling} = \sum_{i=0}^{|Requests|} rank(i) * x_i \tag{2}$$

Where $rank(i)$ is the following for a given request i:
If branch # of i is 1: The tree # of i
Else, if tree # of i is 1, 2 or 3: The branch # of i
Else (if tree # of i is 4): The branch # of i + 3

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 consists 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 at least n courses. The last constraint is varied for n 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 create a new function $rank'(i)$:

$$rank'(i) = -8 + rank(i) \tag{3}$$

The function $rank'(i)$ is used so that it can be minimized for optimal results, with -7 being the most desirable rank in this scheme and -1 being the least desirable. Thus our minimization function minimizes the summation in figure 2, though with $rank'(i)$ substituted in for $rank(i)$.

We solve course scheduling given a set of roughly 40,000 requests from the Spring 2015 Webtree data set. Specifically, we use Sat4j's pseudoboolean solver, a free open-source Java implementation, to solve this satisfaction problem (Berre et al. 2010).

4 Results

The results from our experiment are presented in Table 1. We see the comparison between Webtree's algorithm and our Constraint Satisfaction pseudoboolean algorithm, for three different trials with a different minimum number of courses. Webtree assigned an average course rank of -6.30, which is worse than any of the average course ranks for our algorithm (recall that since this is a *minimization* problem, a smaller rank is better). Our model yielded average course ranks of -6.47, -6.46. and -6.43 for minimum course constraints of 0, 2, and 3, respectively. This is unsurprising, since adding a constraint to give each student a minimum number of courses likely will reduce the rank of the courses that students are given, a sacrifice that is made in order to ensure students receive courses.

However, unless we add a constraint to give each student a minimum of three courses, Webtree assigns an average of more courses. This is a tradeoff that we are willing to accept, since the difference is trivial (Webtree assigns an average of 3.74 courses, and ours assigns an average of 3.72 courses), and the average rank of the courses is much better in our model.

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

Table 1: Our results of course assignment, compared to Webtree as a control. CSP stands for Constraint Satisfaction Problem, the method used in this experiment.

Interestingly, the number of students that receive four courses decreases as we increase the number of courses that students are required to receive. As we increase the number of forced required courses, we take courses away from those who had all four and give them to those who had less than the minimum number. Even though less students receive four courses, the average number of courses received increases. Essentially, we are making a trade off that promotes more equality while still ensuring that students receive courses that they rank highly on their Webtree form.

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 pseudoboolean 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), while giving all students more courses that they want than Webtree does. Our best end state was obtained in the absence of the additional constraint on minimum number of courses granted for each student. Without this state, our model performs better than Webtree by providing a just and appealing distribution of courses for all students. Therefore, if an overhaul to the method of assigning courses at Davidson College is in the near future, framing it as a constraint satisfaction problem should be a strong consideration of the Office of the Registrar.

Future research should establish a new method and interface for ranking courses. It should be simplified from the current state, while still giving students a chance to customize different contingencies depending on whether or not they receive certain courses. Importantly, more research should examine what sorts of additional constraints or constraint modifications could result in improved scheduling, as well as suitable ways to incorporate desired hierarchical schemas such as class and lottery number. Does there exist a constraint schema that can supply a higher number of completed schedules than vanilla Webtree, while simultaneously returning a higher number of fulfilled requests and a higher score of course satisfaction?

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.
We are appreciative to Alden Hart and Rich Korzelius for their feedback on this manuscript.

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.