Analytics in the Pursuit of Optimizing Course-Student Placements for Sarah Lawrence College Registration [Preliminary Report]

Payton Howell, Class of ’23, Sarah Lawrence College, Bronxville, NY[1]
Daniel King, Professor of Mathematics, Sarah Lawrence College, Bronxville, NY

Abstract

Using methods in analytics, a binary linear programming model was constructed to provide course-student assignments in the two rounds of registration at Sarah Lawrence College that were optimal with regard to certain identified goals in the registration process. The analytics-driven approach to forging course-student assignments shows great promise in improving current registration practices at the College.

Introduction

In comparison to that of other undergraduate institutions, the registration process at Sarah Lawrence College (SLC) is rather unique. To start, SLC registration is a two-round, week-long process that typically occurs after all students arrive on campus for the semester.[2] The first round of registration starts with three days of students traveling around campus and interviewing professors about their course offerings for the upcoming semester. This interviewing process allows students to get to know their professors and the course(s) they are teaching. Students are able to ask about the workload associated with the course, examine the course syllabus, inquire about the professor’s teaching style, etc. The professors, in turn, have the opportunity to ascertain whether a student’s academic background and motivations are in line with the course’s content and learning goals.

---

[1] This research was funded through the Sarah Lawrence College Summer Science and Mathematics Research Program.
[2] In response to the Covid-19 pandemic, aspects of the registration process described herein have changed in recent academic years 2020-21 and 2021-22. These alterations primarily affect the student-faculty interview portion of the registration program and do not relate to the course-student assignment process, the focus of the current project.

Students are encouraged to interview for three times as many courses as they can actually take. Once interviews are completed and following careful consideration and detailed discussion with their don (academic adviser), students submit their list of desired courses to the administration. Students do not rank their choices of courses in this first round of registration.

Students must also follow a number of registration guidelines, and these rules of course selection are immutable. For example, students cannot request more than 18 credits. In addition, students cannot request any course for which they did not interview, list two creative writing courses in the same genre (e.g., poetry or non-fiction) or choose all three courses in the same subject (e.g., anthropology).

Once students have submitted their course requests at the end of the third day of first-round registration, various members of the administration are charged with the daunting task of assigning courses to students by the end of the fourth day of registration. Given the restrictive nature of course enrollment caps—most classes at SLC are capped at 15 students—and the inevitable appearance of some über-popular courses requested by a multitude of students, not all students get placed into each of their requested courses.

Students wait with baited breath at the end of the fourth day of registration to hear the results of first-round course placements, hoping that they managed to avoid the ugly outcome of being 'bumped', campus lingo for not getting placed into a requested course. The College reassures its undergraduate community that no student will be bumped from more than one requested course in first-round registration. One of the goals of the current study is to examine actual College registration data to see if this promise on the part of the administration to students is credible. Could it be that the particular distribution of course requests submitted by students in a given semester necessitates that one or more students gets multiply bumped, that is, denied of two or more of their course selections?

Those students successfully placed into all of their requested courses are gleefully done with registration, whereas those who receive a bump from one or more courses have to go through alternate registration on day five of registration week in order to complete their course schedule for the semester. The alternate registration process for students involves another day of interviewing faculty members teaching courses of interest that still have available seats after first-round registration placements were made. In alternate registration, most students are looking for only one course, but each participating student must nevertheless select and rank, in order of preference, exactly three courses on their registration form. The administration attempts to assign as many students as possible to their top ranked course in alternate registration, but the primary goal is to provide each student with a complete curricular schedule for the term. Results of alternate registration are announced by the end of the last day of registration week.

The goals in making course-student assignments in the registration process are clear—to avoid bumping, if at all possible, any student from more than one requested course in first-round registration and to finalize all students' schedules while honoring their course rankings (to the greatest extent possible) in alternate registration—but the logistical path to achieving these goals is not at all straightforward. Furthermore, the need to announce the results of course-student assignments quickly once each round of registration is complete is quite urgent. Time is of the essence during registration week.

The guiding motivation for this project was to employ mathematics (specifically, topics in analytics or operations research) to assist the College with the course-student assignment process of registration. For any given semester, given the raw data of students' requested courses and other relevant parameters (e.g., course enrollment caps), we hoped to identify an assignment of courses to students that was, in a sense, optimal. We wanted to develop techniques that would be helpful in achieving high quality results for both rounds of registration. Finally, we hoped to develop a process that could identify optimal course-student assignments in a timely manner given the great need for speed in announcing the registration results to the campus community.

Linear programming and analytics

Linear Programming is a core topic in the greater discipline of Operations Research (a.k.a. Analytics) focused on optimal decision making. Analytics have been used with tremendous success in the fields of manufacturing, marketing, government, science and professional sports, to name just a few. The methods and techniques of analytics have provided insight into making optimal decisions with regards to plane scheduling, shipping routes, work assignments, warehouse locations, etc. For the conundrum of SLC registration, linear programming can be used to find the optimal set of course-student assignments. We provide a brief discussion here of the essential components to linear programming models.

Linear programming is a toolbox of mathematical concepts and algorithms that can be used to optimize the outcome of a common type of process. Often the outcome of a process can be quantified in some manner using an **objective function**. Depending on the goal of the activity, we typically want to maximize or minimize this objective function. For example, we may wish to maximize revenue, profit, or happiness or minimize cost, processing time, or the amount of material waste created.

Whatever the nature of the objective function, we assume that its value is a direct consequence of choices we make with regard to **decision variables** that are under our control (e.g., the amount of Product A to manufacture, how much of Resource B to use in creating Product A, to what Distribution Center C to ship Product A).

Since the problems to be solved exist in the real world, only certain values can be chosen realistically for the decision variables. The restrictions stemming from reality provide the mathematical **constraints** on our choices of the decision variables. For example, limitations on the availability of Resource B places an upper limit on the number of Product A that can be produced in a given amount of time.

A model can be constructed to investigate any process that has a clear objective function, defined in terms of a set of decision variables that are subject to recognized reality constraints. The nature of the decision variables is a central aspect to the model. Some decision variables are continuous in nature and can take on any value over a certain range (e.g., the width, in yards, of a new warehouse to be constructed). Other decision variables might be integer or whole-valued (e.g., the number of people to assign to register duty in the Saturday evening shift at the local Stop 'N Shop) and others are essentially yes/no binary-valued (e.g., whether to assign student X to course Y).

Linear programming models are those for which the objective function and every constraint can be represented as linear functions of the decision variables. In solving linear programming models, we have a very powerful tool at our disposal: the simplex method; introduced originally in 1946 by George Dantzig and much improved in the subsequent decades. This algorithm brilliantly bypasses the messiness and monotony of fully enumerating and individually examining and evaluating each potential solution to the problem. In contrast, the simplex method 'visualizes' (in a space involving hundreds or thousands of dimensions) all possible solutions, bounces from one potential solution to another, always choosing a path that leads to improving values of the objective function until, in most circumstances, an optimal solution is reached. When some decision variables are integer variables or binary variables, a variation of the simplex method called "Branch-and-Bound" has been used with great success. (Hillier, 2020; Thie, 2008)

In the next pair of sections, we discuss in detail the raw data of SLC registration and modeling the registration process of making course-student assignments as a binary, linear programming problem.

First-round registration: data, model and results

We examined the raw data collected by the College for the Fall 2020 semester. All told, there were *1,268* undergraduate students who participated in first-round registration making a combined number of *3,905* requests for *499* available courses. The majority of these students (*1053*) requested exactly three courses for five credits each, the typical course load of a full-time Sarah Lawrence College student. However, some courses run for fewer credits than the standard five, and we witnessed students requesting a number of courses ranging from one to seven courses. The four students requesting seven courses were all taking part in the Art of Teaching program, and many of the other students looking for courses beyond the typical max were taking advantage of one-credit and three-credit curricular options in dance, theater and music or were participating in a first-year study (FYS) seminar requiring registration in a companion course in the studio arts.

Of the combined *3,905* requests, we found that *1,708* were for *62* overenrolled courses. These are courses where the number of student requests for the course exceeds the preset course enrollment cap. These *1,708* requests for overenrolled courses came from *1,037* different students. In comparing these *1,708* requests with the total number of available seats in overenrolled courses, we identified that at least *625* bumps were required in any assignment of courses to students for the semester. However, the optimal distribution of these *625* bumps across the *1,037* students requesting an overenrolled course remained unclear. The concern that some students might need to be multiply bumped was especially pronounced given that *553* students requested two or more overenrolled courses and a concerning *115* students requested three or more overenrolled courses.

With the goal in mind of minimizing the number of students who receive more than one bump from a course, we set out to find an optimal set of course-student assignments. In our model, the decision variables were binary, each representing whether a specific student i is placed in a specific course j (requested by the student or not). Since there were *1,268* students and *499* courses involved in first-round registration, SLC registration represents a binary linear programming model with $632,732 = 1,268 \times 499$ decision variables each denoted here by $Placement_{i,j}$. By convention we assign the binary values of $Placement_{i,j} = 1$ if course j is assigned to student i and $Placement_{i,j} = 0$ otherwise. In addition, out of convenience in establishing the registration model, we create four 'dummy' courses (courses 500 through 503) and place a student in these courses, in sequential order, if they get bumped from one or more requested courses. Accordingly, we expand the number of decision variables and define them in the same way for these dummy courses, so, for example, if student i gets bumped from exactly two courses, we have $Placement_{i,500} = 1 = Placement_{i,501}$ and $Placement_{i,502} = 0 = Placement_{i,503}$. Each assignment of a course to a student has an associated institutional 'cost' whose value $Cost_{i,j}$ depends on whether the student has requested the course and, if requested, whether the student was placed in the course or bumped. The cost table was created as follows:

Table of 'costs' based on assigning course *j* to student *i*

Assignment	**$Cost_{i,j}$**
Student *i* successfully placed in requested course *j*	0
Student *i* placed in dummy course 500 (student's first bump)	1
Student *i* placed in dummy course 501 (student's second bump)	1,000
Student *i* placed in dummy course 502 (student's third bump)	1,000,000
Student *i* placed in dummy course 503 (student's fourth bump)	1,000,000,000
Student *i* placed in (real) course *j* not requested	5,000,000,000

Our first-round registration model had the goal of is minimizing the total course assignment cost function across all placement of students into courses:

$$MINIMIZE\ Z = \sum_{i,j} Cost_{i,j} * Placement_{i,j}$$

noting that there is only institutional cost if we bump a student from a requested course (and, thereby, placing them in one of the dummy bump courses) or if we, bizarrely enough, assign a course to a student who did not request it. [Though, honestly, everybody really should take a course in Statistics whether they want to or not!]

Finally, we turn our attention to the constraints in this binary linear programming model of SLC registration. These constraints relate to the various 'rules' of SLC registration. The first set of constraints states that no course can be assigned to a number of students exceeding the course enrollment cap or the number of students requesting the course. That is, each course can be filled only to capacity or demand, whichever is smaller. The demand for the course is the number of students who have requested it, and this can be less than, equal to, or greater than the capacity (enrollment cap) for the course. Letting Capacity_j and Demand_j represent the enrollment cap and the number of students requesting course *j*, respectively, then we have the following set of constraints for each course *j*:

$$\sum_i Placement_{i,j} < Capacity_j \quad and \quad \sum_i Placement_{i,j} < Demand_j.$$

Each student must get placed into the same number of courses as they requested (remember 625 of these placements will be into dummy bump courses necessarily). If Requests_i represents the total number of courses requested in by student *i*, then the third set of constraints, in the form of equality constraints, are as follows for each student *i:*

$$\sum_j Placement_{i,j} = Requests_i.$$

The last set of constraints requires students to be placed into the dummy bumped coursed sequentially (as needed), so that for each student *i*:

$$Placement_{i,500} \geq Placement_{i,501} \geq Placement_{i,502} \geq Placement_{i,503}.$$

Applying the branch-and-bound adaptation of the simplex method to this first-round registration model, we found that an optimal assignment had a minimal value of our objective functions equal to exactly *625*. This means the algorithm found a set of course-student assignments in which 6*25* different students were bumped from a single course, and no students were multiply bumped. This is the best we could have hoped for given the raw data of students' course requests. The algorithm took *20* seconds to run and examined *2,098* potential solutions before identifying an optimal one.

We note that our set of course-student assignments is not uniquely optimal and merely one of many possible optimal solutions to the first-round registration model. Other optimal assignments exist, for example, when a bumped student is swapped for a placed student within any one of the 62 overenrolled courses (assuming the newly bumped student is not, as a result of the swap, now multiply bumped).

Without knowing students' preferences among the courses that they requested in first-round registration and without giving some students registration priority (e.g., seniors or Britney fans), we have no metric by which to measure whether one set of optimal course-student assignments is better than another.[3]

Alternate registration: data, model and results

Students who were bumped from a course in first-round registration and other students who, for some reason, did not complete their curricular schedule in that process are eligible to participate in alternate registration. For the Fall 2020 term, *591* SLC students submitted an alternate registration form. Most, but not all, of these students had been bumped from a course in first-round registration.

The vast majority (*576*) of these *591* students participated in alternate registration successfully by ranking, in order of preference, a full set of three courses as requested by the administration. However, thirteen students submitted a ranked list of only two courses, and two students requested only one course. The College warns students against submitting an incomplete form featuring fewer than three ranked courses, thereby risking invalidation of their alternate registration.

The goal in assigning courses to students in alternate registration differs significantly from that of first-round registration. For the former, we aimed to make course assignments with the goal of not bumping any student from two or more courses. In contrast, the goal of making course assignments in alternate registration is to complete every student's schedule and, to the greatest extent possible, honor student preferences as stated in their ranked course requests. So rather than assigning an institution cost in assigning a course to a student as we did in first-round registration, we now consider the benefit of making such an assignment in alternate registration. The greatest benefit comes from assigning a course to a student that listed it as the top choice. The values of the benefit, denoted $\text{Benefit}_{i,j}$ of assigning student i to course j are listed in the following table:

[3] The College in the past has given priority to some students in some situations. For example, students bumped from a particular professor's course in the past have received priority in studying with that professor in a subsequent term. Pre-health students and students in Columbia University's 3-2 combined engineering program are sometimes given priority into courses needed to progress through those programs in a timely manner. Moreover, some professors take advantage of giving some students priority in their seminar courses. The first-round and alternate registration models described in this study could be adapted readily to take account of these situations in which some students are given priority in registering for some courses.

Table of 'benefits' based on assigning course *j* to student *i*

Scenario	$Benefit_{i,j}$
Student *i* placed into top ranked course *j*	20
Student *i* placed into second ranked course *j*	10
Student *i* placed into third ranked course *j*	1
Student *i* placed into a course *j* not requested	-1,000,000

The negative benefit of *-1,000,000* indicated above for placing a student into a course not requested is meant to preclude the possibility of our algorithm making such a course-student assignment in optimizing overall assignment benefit.

For those fifteen students, mentioned above, who did not submit a complete list of three ranked courses, adjustments to the above benefit table were necessary. We decided that the top assignment benefit value of 20 would not be available for any course assignment made to these students (who might be attempting to game the system). Specifically, an assignment of a course to a student who only selected two courses would yield a benefit of 10 or 1, and an assignment of a course to a student who only selected one course would yield a benefit of 1.[4]

Though the goals of making course-student assignments have changed in moving from first-round registration to alternate registration, the decision variables have not. These are still the set of binary variables $Placement_{i,j}$ equaling 1 or 0 if student *i* is or is not placed into course *j,* respectively.

Our alternate registration model has the goal of maximizing the total course assignment benefit function:

$$MAXIMIZE\ Z = \sum_{i,j} Benefit_{i,j} * \ Placement_{i,j}$$

[4] The Draconian invalidation of the registration of any student who doesn't list a complete set of three ranked courses could be readily folded into the linear programming model of alternate registration.

The constraints in the alternate registration model are similar to those in the first-round registration model. For example, courses still cannot be filled beyond capacity or demand. However, the course capacities are not the same as those in first-round registration as some students have already been placed into these courses and thereby reducing course capacities. We denote by Capacity_j the number of remaining seats available to students in course j. Similarly, the demand for the course j, denoted by Demand_j, is the sum of the number of students placed in course j in first-round registration and the number of students requesting this course in alternate registration. Thus, we have the following pair of constraints for each course j:

$$\sum_i Placement_{i,j} < Capacity_j \ \ and \ \ \sum_i Placement_{i,j} < Demand_j.$$

In the alternate registration model each participating student must get placed into a requested course. However, there were several students who requested placement into more than one course in alternate registration. If we let Requests_i now represent the number of courses needed by student i in alternate registration, we have the following additional model constraint, as before in first-round registration, for each student *i:*

$$\sum_j Placement_{i,j} = Requests_i.$$

We note that there are no bumps in alternate registration, so no need for dummy bump courses or any bump-related constraints for our LP model of alternate registration.

In applying the branch-and-bound adaptation of the simplex algorithm to the alternate registration model, the solution found to our course-student assignment problem had a maximal associated benefit of *9,416*. In this solution, *401* (*68%*) of the *591* participating students received their top-ranked course selection, *134* (*23%*) received their second choice and only *56* (*9%*) received their third choice. This is an optimal set of course placements in terms of maximizing the total benefit points associated with course placements (as described above) across all students participating in alternate registration.

According to Sarah Lawrence College Registrar Daniel Licht, the College's manual process resulted in *370* (*63*%) of students receiving their first choice, *81* (*23*%) receiving their second choice and *81* (*14*%) receiving their third choice.

LP Model Placements	Alternate Registration Placement	College Placements[5]
401 (68%)	First-Choice Courses	370 (63%)
134 (23%)	Second-Choice Courses	136 (23%)
56 (9%)	Third-Choice Courses	81 (14%)

Once the binary linear programming model was defined, the algorithm took *13* seconds and considered *92* different potential solutions before landing on a set of optimal course-student assignments that maximized our assignment benefit function.

Analysis & Next Steps

The initial results stemming from our analytics-driven method of making course-student assignments for both first-round and alternate registration proved to be quite promising. Compared to current practices[6], the approach using analytics could relieve the College of considerable pressure in its efforts to swiftly announce results of course placements in both rounds of registration. The process is rather straightforward. First a simple spreadsheet table is created indicating which students (listed by row) requested which courses (listed by column). Once done, the binary linear programming registration model, complete with decision variables, objective function, and constraints defined herein, can be created in short order. Even a modestly fast laptop computer equipped with the right software can run the registration assignment model and find an optimal set of course-student assignments *in seconds.*

Additional value of this analytics approach stems from its tremendous flexibility. In our work on first-round registration we essentially minimized (to zero) the number of students who were multiply bumped by minimizing the total institutional cost of course-student assignments. For alternate registration, we assigned a point benefit structure to assigning courses to students depending on the student's ranking of the requested course and found a set of course assignments that maximized the benefit point total across all participating students. If the goals of either

[5] We note a four person discrepancy in the total number of students placed in these two methods

[6] We did not inquire deeply into current practices employed by the College in making course-student registration assignments, but we understand that a technology-assisted process is in use for both first-round and alternate registration in order to find an initial set of assignment which, in turn, is hand tweaked by members of the administration until an acceptable set of registration assignments is achieved.

round of registration are changed (e.g., different cost structure scheme in first-round registration or different benefit point allocations in alternate registration), the linear programming registration model can be swiftly changed in accordance and a new set of optional assignments swiftly obtained.

The analytics approach to registration also provides for simple sensitivity analysis. If there is a sudden change to the raw registration input data—Jill's list of course requests is corrected for an error found in data entry, Joan's course requests are deleted entirely due to tuition non-payment or Joseph, it is now realized, must absolutely be placed in the Organic Chemistry seminar—then once again, the model can be quickly adjusted and a revised set of optimal course-student assignments determined in a brief few moments.[7]

In the future it might also be possible for the College to create a streamlined, single-round registration process that provides optimized course-student placements. This alteration in process might require students to submit requests for more courses than they actually need and/or to rank their course requests (as is currently done in alternate registration). Another option is to allow the students themselves to define and self-distribute a fixed number of benefit points associated with being placed in any of their requested courses. We understand the College has already formed an ad-hoc committee focused on registration process reform. Analytics could play a central role in the successful pursuit of some of these promising reforms.

Acknowledgements

The authors are grateful to the donors supporting the SLC Science and Mathematics Summer Research Program for providing the funds that made this study possible. We also thank the following members of the SLC administration for providing the raw registration data from the 2020 fall term and for sharing their thoughts and understanding of current registration practices: Provost and Dean of the Faculty Kanwal Singh, Associate Dean of the College Melissa Frazier, Registrar Daniel Licht and Chief Technology Officer and Chief Information Security Officer Sean Jameson.

---

[7] The closer the established model reflects the educational values and goals of the SLC registration process, the less the entire process will necessitate manual accuracy checks and by-hand adjustments of initial assignments. A single alteration of one student's course schedule can have a sizable cascading effect, resulting in a multitude of changes affecting a large number of students and courses, before settling back (hopefully) onto an optimal solution.

References

Hillier, Frederick S. and Lieberman, Gerald J. (2020). *Introduction to Operations Research,* 11th Edition. McGraw-Hill Education Publishers.

Thie, Paul R and Keough, Gerald E. (2008). *An Introduction to Linear Programming and Game Theory*. Wiley-Interscience.