Sponsor Group Assignment at Pomona College as a Constraint Satisfaction Problem

Sarah Jundt and **Shannon Lubetich**
Pomona College

Abstract

Pairing college roommates is a laborious process. We confront this problem, as well as pairing hall groups known as "sponsor groups," by formulating them as Constraint Satisfaction Problems. Students are prohibited from being paired together based on preferences submitted on housing forms. Our results return successful roommate and sponsor group pairings, with sponsor groups having more intragroup similarity than intergroup similarity.

Introduction

The first year of college can be a life-changing experience, greatly affected by the people students meet, and especially the people they live with. At Pomona College, sponsor groups consisting of 10 to 20 freshmen who live in the same hall are an essential aspect of first-year culture. Roommates and sponsor groups are currently paired by Head Sponsors, who have to sift through hundreds of applications. If roommate pairs and sponsor groups could be generated by a computer, less manual work would be required and less time would be spent.

We propose a quicker and more efficient approach to this pairing and grouping process. By formulating these as Constraint Satisfaction Problems, we produce successful assignments of students in roommate pairs and sponsor groups, where students are more similar to others within their own sponsor group that others in the student population.

Related Works

SARAH TAKE IT AWAY!

Background

Sponsor groups are created by first considering all individual students and matching them with a peer, thus creating roommate pairs. Next, each pair is matched with four to nine other pairs in order to form a sponsor group. Our project focuses on mimicking this process. We determine reasonable constraints on roommates based on preferences they submitted on housing applications and use these to generate room-

mate pairs. We then use joint preferences of roommate pairs to place these students within sponsor groups.

We approach this as a Constraint Satisfaction Problem because, due to preferences, if students simply are too different to get along, we say there is a constraint between them. We generate constraints for all possible pairs of students, and then try to make successful assignments of students into roommate pairs and roommate pairs into sponsor groups that satisfy these constraints.

This assignment of students is accomplished by using backtracking search. Backtracking takes one variable at a time, assigns it to a value in its domain, and adds this to an overall assignment of all variables. Then, all constraints are checked to guarantee that this new assignment does not violate any constraints with previous assignments. If no constraints are violated and a full assignment of all variables is reached, then a complete and successful assignment is reached.

Backtracking can be improved with the addition of intelligent ordering and inference. Intelligent ordering refers to which variable will be selected from the unassigned variables to be assigned a value next, as well as the ordering of what value within its domain should be selected as the next attempt at an assignment. Inference can be used after an assignment to eliminate all possible values from other unassigned variables' domains. This helps speed up the algorithm because it minimizes the occurrence of assignments that violate constraints involving previously assigned variables.

System Description

The problem of placing students into roommate pairs and sponsor groups will be configured as two separate constraint satisfaction problems. Students will first be placed into roommate pairs and will then be placed into sponsor groups. Each of these placements will use backtracking and each will return an assignment satisfying specified constraints.

Creating Roommate Pairs

The constraints for this problem will be based on maximization of similarity between the preferences of students within a roommate pair. The constraint for roommate pairs will be that paired students do not have any preference rankings that differ by more than some constant number. The constant will

be determined by experimentation and will be the highest integer for which a successful assignment of roommate pairs can be found.

Every single individual has a list of preferences, which are the numbers 1 through 10, indicated how important a certain statement is. These statements are below:

1. Is serious about studying and will make studying a priority of our room.

2. Allows me to have visitors over as often as Id like and doesnt mind having people over in our room.

3. Will be my friend who confides in me and likes to do a lot of things together.

4. Doesnt let school take over our entire lives, and knows how to have a good time.

5. Shares responsibility for keeping our room neat.

6. Respects my need for privacy and will allow me some time to myself.

7. Respects my property and doesnt borrow my things without asking.

8. Has similar sleep habits (i.e. windows opened/closed, absolute quiet, no light, etc.)

9. There are people who share backgrounds and cultures similar to my own.

10. People are aware of, sensitive to, and willing to discuss multicultural issues.

These preferences are stored as a feature vector, along with the student's self-specified gender. So for each student we have a vector as follows.

$$[M/F, P_1, P_2, ..., P_{10}] \text{ where } 1 \le P_i \le 10$$

The first entry in the vector is important because Pomona currently only pairs same-gender roommates. That is, someone who specifies male gender will be guaranteed a roommate who also specifies male gender. Because of this distinction, we run our roommate pairing algorithm twice, once on all students with female gender and once one all students with male gender.

In formulating this as a Constraint Satisfaction Problem, our variables are ID numbers of every single student, and these variables have domains of the ID numbers of every other student, thus representing all potential roommates for a given student.

To incorporate constraints, we initially iterate through all students and compare the numerical values in their feature vectors that correspond to preferences. If any of these preferences differ by a certain constant amount, these students cannot be roommates. This is specified by removing these students from each other's domains, so that such a pairing would never occur that violates constraints. That is, given two feature vectors of students, if there exists any preference that differs by a value greater than a constant C, these students are removed from each other's domains.

Backtracking search is then applied to generate roommate pairs, with the intelligent ordering of selecting the next student to be assigned a roommate using the least-remaining values heuristic in order to reduce the running-time of backtracking. In order to select a roommate for this student, we simply randomly reorder all possible roommates in this student's domain. This allows us to return multiple different, yet still constraint satisfying, pairings over multiple runs. We then use forward-checking to remove the two roommates involved in this pairing from all other students' domains, because since they have been paired, they cannot be matched as roommates with anyone else. If this removal process leaves an unassigned student with an empty domain, this means there are no possible remaining students they can have as roommates and yet still satisfy constraints. This is a failure of the assignment, so we must backtrack and attempt to assign our next student to a different roommate that will result in pairing everyone successfully. Once a complete assignment is found of all students into roommate pairs, this assignment is returned as a success.

Creating Sponsor Groups

The constraints for this next step will be based on maximization of similarity between the preferences of roommate pairs within a sponsor group. For this problem, we combine roommate pairings into a single identifier (the lower of the two student's ID numbers) with a single associated feature vector (where the entries are the gender of the two students and the average of the remaining numerical values in their feature vectors).

Pomona College creates 30 sponsor groups for first-year students. Thus, for this Constraint Satisfaction Problem, every variable is a unique ID number that represents a roommate pair, and the domain for every variable are the numbers 0 through 29, which represent the labels of each sponsor group. When a variable is assigned a value, this is equivalent to specifying that the roommate pair associated with the given ID number is in the sponsor group with the same value as that label.

Here, we incorporate constraints by building a constraint satisfaction graph. The preferences described in the previous section do not necessarily all apply as strongly to a student's hallmates as to their roommates, so we chose to rank some as more important than others in forming a cohesive sponsor group. This importance was codified by requiring the difference between these two preferences to be within a smaller constant. So that in our constraint satisfaction graph, two roommate pairs were involved in a constraint of not being able to occupy the same sponsor group if any of their averaged preferences were different by our specified constants.

Backtracking search is then applied to generate sponsor groups, with the intelligent ordering of selecting the next roommate pair to be assigned to a sponsor group using the least-remaining values heuristic in order to reduce the running-time of backtracking. In order to select a sponsor group for this student, we reorder the domain from smallest to largest sponsor group to attempt to assign a student to a sponsor group that is more empty first before assigning to a more populated group.

There are several constraints we incorporate along the way to yield valid sponsor groups. We enforce a minimum size of 10 students (5 roommate pairs) and a maximum size

of 20 students (10 roommate pairs) for every sponsor groups, because these are the actual range of sizes of sponsor groups at Pomona College. Additionally, Pomona attempts to have a balanced number of males and females in sponsor groups, so we try to form groups with as close to a 1:1 ratio as possible. This ratio is accomplished by only assigning a roommate pair of gender M to a sponsor group if that sponsor group already does not have more pairs of M than F gender, and vice versa for assigning a roommate pair of gender F.

We also use forward-checking to determine if any roommate pair X is involved in a constraint with the newly assigned roommate pair Y by checking if there is an edge between these two pairs, X and Y, in our constraint satisfaction graph. If this is the case, then the sponsor group Y was just assigned to is removed from the domain of X.

If any of these constraints are not successfully met, this assignment is a failure, so we must backtrack and attempt to assign our next roommate pair to a different sponsor group that will satisfy all constraints. Once a complete assignment is found of all roommate pairs into sponsor groups, this assignment is returned as a success.

Results

We aim to obtain groupings of students into sponsor groups after placing individual students in roommate pairs. The success of the constraint-satisfaction problem will be evaluated using K-means clustering. The students will be clustered into 30 sponsor groups based on their preferences. Those clusterings will then be evaluated using purity, where the label associated with each student is the sponsor group number assigned to them by the constraint satisfaction results. A higher purity will indicate that our constraint satisfaction has placed students in sponsor groups with other similar students.

Roommate Pairs

Interestingly, we found that the smallest constant difference C for which students can be successfully paired into roommates differs depending on genders. For females, this occurs when $C = 4$; for males, this occurs when $C = 5$. This means that females can be subject to more strict constraints so that none of the preferences between roommates differ by more than 4 points and still find successful pairings, whereas males require that none of the preferences differ by more than 4 points.

Sponsor Groups

Bibliography

S. Gravel, V. Elser. "Divide and concur: A general approach to constraint satisfaction." Physical Review E 78, 036706 (2008). American Physical Society.

A.K. Mackworth, E.C. Freuder. "The complexity of some polynomial network consistency algorithms for constraint satisfaction problems." Artificial Intelligence, 25 (1985), pp. 65-74