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 the problem of creating 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, and it is often greatly affected by the people students live with. At Pomona College, sponsor groups consisting of 10 to 20 freshmen living together are an essential aspect of first-year culture. Roommates and sponsor groups are paired manually at Pomona by a small number of 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 groupings as Constraint Satisfaction Problems, we are able to produce successful assignments of students into roommate pairs sponsor groups where assigned students are more similar to others in their sponsor group than to those outside their group.

Related Works

SARAH TAKE IT AWAY!

Background

Sponsor groups are created at Pomona College by first considering individual students and matching each student with a roommate. Next, each roommate pair is matched with four to nine other pairs in order to form a 10 to 20 person sponsor group. Criteria for groupings are flexible, but sponsor groups tend to consist of people who could be classified as "similar." Before coming to Pomona, first-years fill out a housing form on which they provide qualitative information about their habits and quantitative information about the extent to which several different traits matters in their roommate or hall. Every single individual has a list of these quantitative preferences, which range from 1, indicating not important, to 10, indicating very important. 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 I'd like and doesn't 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. Doesn't 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 doesn't 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.

Statements 1 through 8 ask the student explicitly about qualities of their future roommate, whereas statements 9 and 10 refer to the area the student will live in, which includes both their roommate and the rest of the sponsor group.

The problem of assigning students to sponsor groups is easily formulated as a constraint satisfaction problem. Constraint Satisfaction Problems can be solved using the Backtracking algorithm. Backtracking assigns variables to values in a depth-first manner. Backtracking considers one variable X at a time, assigns it to a value D in its domain, and then adds the variable assignment $X = D$ to an overall assignment of all variables. Then, all constraints are checked to guarantee that this new assignment $X = D$ does not violate any constraints on the variables that have previously been assigned. If a constraint is violated, the assignment of $X = D$ is removed from the overall assignment and X is assigned to some other value D' in its domain. 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 of domains, heuristics to choose the next variable to assign, and an inference to remove inconsistent values from the domains of variables. To create even groupings of variables among domain values, domains can be ordered such that those with fewer variables assigned to them are chosen first. The Minimum Remaining Values (MRV) heuristic assigns variables with fewer possible assignment values first. This ensures that less backtracking is necessary, speeding the algorithm. Inference can be used after an assignment to eliminate all possible values from other unassigned variables' domains, working in conjunction with MRV to speed the algorithm by minimizing the occurrence of assignments that violate constraints involving previously assigned variables.

System Description

The problem of placing students into roommate pairs and sponsor groups is configured as two separate constraint satisfaction problems. Students are first placed into roommate pairs and are then placed into sponsor groups in an attempt to emulate the human process for such sponsor group pairings. Each of these placements is performed using backtracking and each returns an assignment satisfying constraints designed to help create sponsor groups with students who are similar in essential ways.

All constraints focusing on the "preferences" of students refer to the students' numeric ranking of their preferences about the traits of their roommate or surroundings as specified on their housing forms.

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.

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

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. Gender affects sponsor group assignments as well, as Pomona attempts to balance the number of males and females in each sponsor group. The remaining values in the feature vector are numeric preferences from 1 to 10 ranking the importance of statements as presented and discussed in the previous section.

Creating Roommate Pairs

The constraints for this problem are based on maximizing the similarity between the preferences of students within a roommate pair.

In formulating this as a Constraint Satisfaction Problem, our variables are unique ID numbers of every 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 Minimum Remaining Values heuristic in order to reduce the running-time of backtracking. To select a roommate assignment 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 current 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 students' ID numbers) with a single associated feature vector (where the entries are an M or F to represent the gender of the two students followed by the average of the 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 sponsor groups. 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 specified by students on their housing forms, as described in the previous section, do not necessarily apply as strongly to a student's hallmates as to their roommates, so we choose to rank some as more important than others in forming a cohesive sponsor group. This importance was codified by requiring the difference between higher importance preferences to be within a smaller constant difference. In our constraint satisfaction graph, two roommate pairs are 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 Minimum Remaining Values heuristic. 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), 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 our newly assigned roommate pair X is involved in a constraint with any other 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 X was just assigned to is removed from the domain of Y, because we have a constraint that specifies that they cannot be in the same sponsor group.

If any of these constraints are not successfully met, this assignment is a failure, so we must backtrack and attempt to assign our 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 were able to successfully produce roommate pairings and sponsor groups satisfying our specified constraints. Because we do not have data on how these students were actually paired and grouped, there is no way to test how “correct” our approach was as compared to how it is done by the Pomona College housing committee. For roommate pairs, we can examine if we found a successful matching and how strict or loose our constraints have to be in order to achieve success. For sponsor groups, our main goal was to create groupings that had preferences more similar to each other than to the overall average of all students outside of the group.

Roommate Pairs

To measure success for roommate pairs, we can examine if we found a successful matching and how strict or loose our constraints have to be in order to achieve success. 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

For sponsor groups, our main goal was to create groupings in which the roommate pairs had similar preferences to one another. We evaluate this in two ways.

- Method 1: K-means Clustering
  First, we use K-means clustering to generate 30 clusters of roommate pairs based on their average preferences. This algorithm operates by assigning a roommate pair to an existing cluster with the closest centroid to the pair’s preferences. We start with random roommate pairs as the centroids of each cluster, but then recalculate centroids by averaging the preferences of all the roommate pairs in that cluster. This process repeats until a specified number of iterations has been achieved. This approach does not weight preferences as less or more important and relevant for sponsor group pairings as our CSP algorithm does, but we wanted to see if this method resulted in similar groupings of students as our Constraint Satisfaction Problem. The similarity between outcomes is measured using purity. Here, we cluster the roommate pairs using K-means clustering, and then we label every student with the sponsor group they were assigned in the CSP. Then, each cluster is labeled with the sponsor group label of the majority of roommate pairs within the cluster. Each cluster is evaluated for the percentage of roommate pairs within it that share their sponsor group label with the majority label assigned to the cluster. These percentages are then averaged to find the purity of the clustered data.

  In generating XXX successful roommate and sponsor group assignments, we attained an average purity of XXXXXXX.

- Method 2: Average Preference Differences
  Our second method of evaluation involves calculating the difference of a roommate pair’s preferences and their sponsor group’s average preferences and comparing this to the difference of their preferences and the preferences of all the students. The idea involved here is that, to be successful, sponsor groups should contain roommate pairs who all have something in common, and are thus more similar to each other than they are to the student body population as a whole. We find the average intra-group difference and all student difference for each of the 10 ranked preferences. Then, we find the average of these differences for each set of preferences, depending on their relevance to sponsor group assignment.

  The preferences ranked as most important are 9 and 10, high importance includes 1, 3, and 4, mid importance-2 and 6, low importance-5, 7, and 8.

We successfully generated XXX roommate and sponsor group assignments, and found the average intragroup difference and intergroup differences across all these assignments, as seen in Table 1.

Preference Importance	Intragroup Difference	Intergroup Difference
Most	XXX	XXX
High	XXX	XXX
Mid	XXX	XXX
Low	XXX	XXX

Table 1: Average Preference Differences

As we can see....

Conclusion

CONCLUDE STUFF

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