Winter 12-2019

Using Goal Programming to Make Large Scale Heterogeneous Teams

Bill Young

Vic Matta

Using Goal Programming to Make Large Scale Heterogeneous Teams

Prototype Demonstration (Extended Abstract)

Bill Young
Ohio University
YoungW1@ohio.edu

Vic Matta
Ohio University
Matta@ohio.edu

Abstract

This extended abstract describes a model for using a goal programming technique to resolve the problem of creating teams from a large resource pool, with multiple competing constraints. The objective is to allocate teams based on a heterogeneous and equitable distribution of candidate's characteristics. The model uses a penalty system to optimize teams: it applies a penalty based for deviation from equitable distribution of characteristics, i.e. a soft constraint, while also adhering to hard constraints.

Keywords

Goal Programming, Heterogeneous Team Formation, Penalty Scoring

Introduction

Heterogeneous teams offer a set of balanced benefits. For instance, problem solvers and scientific thinkers of a team may need a team manager who excels at facilitation and team development deliver a product/service for a client (Fitzpatrick & Askin, 2005). We discuss an application in a large-scale academic setting in which teams needed to be carefully designed for good heterogeneity, while satisfying multiple competing priorities. Such teams are designed for balanced performance and have found to direct applications in other venues, such as sports, military, and even healthcare operations. This extended abstract describes a prototype developed to that resolves a multi-criteria equitable team formation problem that can be scaled up to 1000 candidates, based on a limitation of the software.

Project Description

The prototype was developed in an academic setting at a medium sized College of Business, in a midwestern university. Students applied to participate in a study abroad program (SAP) run by the college in one of countries. An SAP site director runs the program in each country and is managed by an SAP program director. Details provided in the student application are used for creating heterogeneous teams, e.g. gender, major, rank, GPA, as well as country preference, etc.

The information collected by the application survey was used to derive metric a called the Student Competitiveness Score (SCS). The intention of this metric was to combine factors with similar characteristics which ultimately could be used within the model to formulate assignment of students to SAP sites. The SCS was based on a student's academic grade point average (GPA), a student's SAP site preference, as well as a measure of their face-to-face interview with the director of a particular SAP site. Students were given the option to rank order the sites to which they were willing to go – measured as the number of sites they are willing to travel to. Therefore, a high score on the SCS was an indicator of a high-quality student, a highly desirable characteristic that needed to be distributed equitably across SAP sites.

If a student indicated a high preference to go to a particular site and had a high GPA and high interview score, then that particular combination would provide a higher likelihood of that particular student would be assigned to a site with a rather high student-indicated priority. However, a high SCS would not guarantee assignment to their highest indicated priority due to the interest in equitably distributing other candidate characteristics across SAP sites.

These characteristics included gender, student rank and prior business consulting course (BCC) experience. For example, it made sense to distribute the ranks of freshmen, sophomores, juniors and seniors equitably across the program sites, so no site had all freshmen, leaving all seniors to a different site; or the male-female ratio needed to be as consistent as possible across all SAP sites. The next section outlines the method followed to accomplish these objectives.

Methodology

A total of 241 students applied for the program. Of these 29 students did not meet the eligibility criteria due to reasons ranging from low GPA to prior disciplinary violations. The ideal capacity at each Study Abroad Program (SAP) site was 20. For 8 SAP sites, the target program capacity was 160, which meant that 52 students would not qualify for placement at any location based on a low competitiveness score (SCS).

The project used software called Solver ("Frontline Solutions," 2019) to accomplish the goal programming task. For the goal programming model, it was desired that each site would have roughly the same SCS score. From a metric perspective, the SCS was a simple linear combination of weights and the numeric values that were assigned with the application or interview results. A more formal definition of SCS is provided below.

$$SCS_{i,j} = \omega_{Site\ Preference} * Applicant\ Site\ Preference_{i,j} + \omega_{Interview\ Score} * Applicant\ Interview\ Score_{i,j} + \omega_{GPA} * Applicant\ GPA_{i,j}$$

where

- SCS = Student Competitiveness Score
- i = student number (ranged from 1 to 212)
- j = site number (ranged from 1 to 8)
- $\omega_{site\ preference}$ = weight placed on the GCP site preference that is indicated by the student
- $\omega_{interview\ score}$ = weight placed on the GCP interview with the GCP Director
- ω_{GPA} = weight placed on the cumulative GPA of the applicant
- *Applicant Site Preference* = the GCP site preference indicated by the student
- *Applicant Interview Score* = the quality of interview indicated by the GCP Director
- *GPA* = the cumulative GPA of the applicant

From a goal programming perspective ("Goal Programming," 2013), both “hard” and “soft” constraints were used. Hard constraints simply mean that they are constraints that cannot be violated. Soft constraints, on the other hand, are constraints that can be violated. In terms of a hard constraint, a single student would not be assigned to more than one site. However, this implied that a student may not be selected to a specific site at all. In addition, another hard constraint related to the maximum capacity at each site. Thus, the number of students assigned to a certain site had to equal the capacity of the site. However, this hard constraint could easily be modified if it was not necessary to fill every site to their maximum capacity.

Soft constraints represented certain characteristics that are considered desirable or undesirable in term of student to SAP site assignment. Goal programming was an appropriate model for an assignment problem due to all of the competing constraints at hand. For example, it was desirable to send the same number of males and females to each site. However, the application pool might not have sufficient applicants to achieve this goal in addition to the other goals related to a balanced site with respect to SCS, BCC experience, or class standing distribution. Thus, in our rendering of the solution, goal programming did not explicitly enforce that each soft constraint would achieve a certain target, but it attempted to generate a feasible solution which satisfied all of the soft constraints as closely as possible, based on a the penalty for deviation from the target for each constraint. A list of brief descriptions for each soft constraint is provided below.

- *SCS: As equal as possible across all SAP Sites, where a penalty is enforced if a particular site is lower than others*
- *Gender: As equal as possible across all SAP Sites, where a penalty is enforced if a particular site has more males than females, or more females than males*
- *BCC: As equal as possible across all SAP Sites, where a penalty is enforced when a particular site does not have more students with BCC than students without BCC*

- *Class Standing: as equal as possible across all SAP sites, where a penalty occurs if the mix of students has more lower standing students (i.e. freshman) than higher (i.e. senior).*

The SAP Solver model's objective function was a minimization of Total Weighted Percent Deviation (TWPD). This function, albeit simple, yet powerful, represents the deviation from the amount a particular solution goes above or below a certain target in the form of a percent where a penalty factor is applied for being over or under a certain target. The equation for TWPD is summarized below.

$$TWPD = \sum_{i=1}^{I} \sum_{j=1}^{J} \omega_{i,j} \frac{Under_{i,j}}{Target_{i,j}} + \omega_{i,j} \frac{Over_{i,j}}{Target_{i,j}}$$

where:

- I = The number of SAP Site locations (e.g. 8)
- J = The number of Soft Constraints (e.g. 4)
- i = A particular SAP Site (e.g. 1)
- j = A particular soft constraint (e.g. gender)
- $Target_{i,j}$ = The ideal value of a soft constraint (e.g. 12)
- $\omega\_under_{i,j}$ = The penalty weight for a solution being greater than a constraint's target value (e.g. 1)
- $\omega\_over_{i,j}$ = The penalty weight for a solution being less than a soft constraint's target value (e.g. 1)
- $Under_{i,j}$ = The amount the solution is less than the soft constraint's target value as a percentage
- $Over_{i,j}$ = The amount the solution is greater than the soft constraint's target value as a percentage
- *CSi,j* = The competitive score for an applicant at a specific SAP site (calculated in equation #1)
- *Di,j* = The binary decision for a specific applicant at a specific SAP site location

The Solver model ensured an optimal distribution of the penalty across by distributing j soft constraints equitably across i SAP site locations. The system is operational and is being used to create teams to facilitate the Study Abroad Program. A video has been created to demonstrate the full functionality that could be used at the symposium, pending acceptance. The link has been excluded in this extended abstract because it contains identifying information of the authors. At the symposium, the authors also expect to provide a graphical output of the results of the Solver system.

Outcomes and Evaluation

At this point the method is evaluated based on its ability to provide allocate students to sites in a way that is unarguably fair, i.e. at face value. For instance, SCS scores and performance of students needed to be well distributed across websites. Figures 1 & 2 show that the variation of SCS was 0.1. and that of GPA was 0.3. This was dramatically improved from the non-optimized distributions. Figure 5 also shows a distribution of communication abilities of assigned students.

Figure 1: SCS Summary of Assigned Students

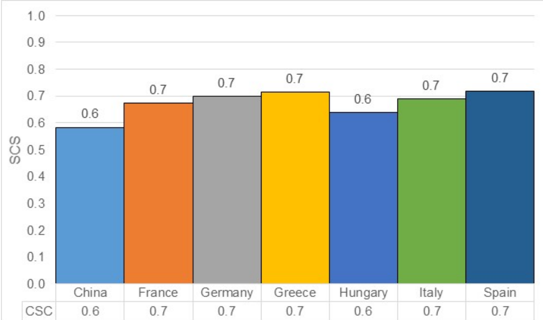
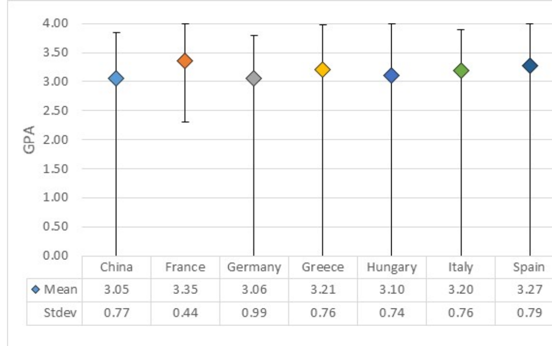


Figure 2: Performance of Assigned Students



Site preferences were far more difficult to resolve because gathered more than ten times the first preferences of the least popular site. For instance, figure 3 shows that 70% of the students selected Greece as either a first, second or a third preference. Therefore, using an approach to satisfy optimize and distribute first preferences was never an option. Instead, the objective was to satisfy the most first preferences without disrupting the optimization of SCS. Figure 4 shows the allocation of preferences, and figure 6 demonstrates that 77% of the students who had applied were able to go to their first or second site preferences.

Figure 3: Site Preference Requested

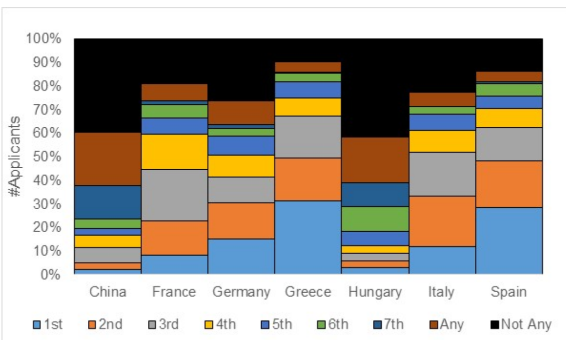


Figure 4: Site Preference Assigned

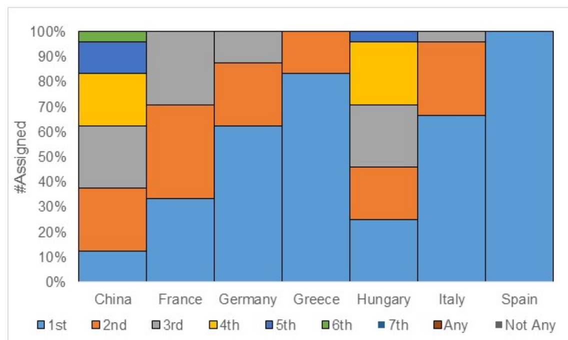


Figure 5: Interview Score of Assigned

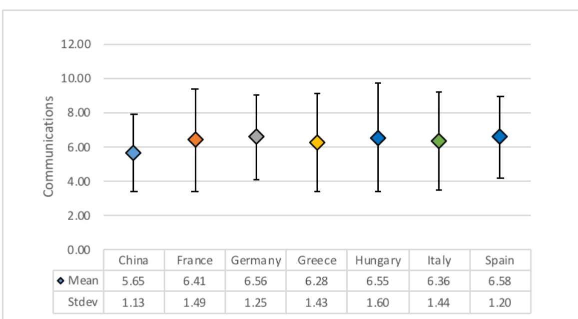
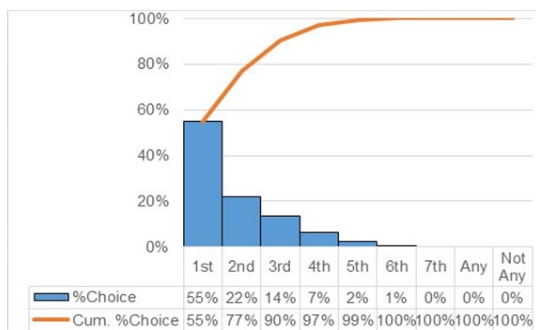


Figure 6: Preference Summary of Assigned



Next Steps

The Student Competitiveness Score (SCS) was combined to consolidate similarly desirable variables in order to reduce the degrees of freedom and therefore simply the experiment for this prototype. In the next steps, these variables would be teased out. The authors are evaluating the interest of optimizing other academic and demographic data as well.

The authors expect to compare the outputs using a mixed method approach: the authors would compare student administrative and faculty performance outcomes as well as qualitative feedback from site directors of different site.

References

Fitzpatrick, E. L., & Askin, R. G. (2005). Forming effective worker teams with multi-functional skill requirements. *Computers & Industrial Engineering, 48*(3), 593-608. doi:10.1016/j.cie.2004.12.014

Frontline Solutions. (2019). Retrieved from http://solver.com

Goal Programming. (2013). *Modelling with Xpress.* Retrieved from http://www.maths.ed.ac.uk/hall/Xpress/FICO_Docs/optimizer/HTML/section5004.html