A Mathematical Model for Optimizing Matching Between Students and Advisors

Hanmi Zou (Class of 2020, Applied Mathematics, Psychology)

Chengwu Shen (Class of 2019, Applied Mathematics, Computer Science minor)

Project Advisor: Anke van Zuylen (Mathematics)

hzou@email.wm.edu, cshen04@email.wm.edu, anke@wm.edu

Abstract

We study the mathematical connection between optimal matching for college freshmen with pre-major advisors and the transportation problem model in the field of Operational Research. In particular, we use the transportation model and write a Python program based on several open source Python packages to generate an optimal matching result for freshmen and pre-major advisors with the data collected by the Office of Academic Advising in William and Mary. The results are compared with those of last year and show a significant improvement.

1 Introduction

According to a new Elon University Poll, the percentage of students reporting a very rewarding experience increased with the increase of close relationships with faculty and staff; students are also more likely to consider the college was worth it when they have strong and meaningful relationships with faculty and staff. Furthermore, according to Peter Felten, one of the authors of an article examining the impact of peer and mentor relationships on the college experience, early semesters in college are critical for developing relationships that uphold long-term learning and professional outcomes. Therefore, a like-minded pre-majors advisor, who may be the first faculty member that freshmen meet in their college experience, is even more decisive for each student.

The goal of this research project is to use the linear operational model, specifically the transportation problem algorithm, to optimize the matching between freshmen and pre-major advisors and to maximize the satisfaction degree between these two groups (students who have specific interests toward certain majors are successfully assigned to advisors with the same indicated majors). The operational research model can help the Office of Academic Advising (OAA) improve the decisionmaking process and pairing between advisees and advisors. The results of the model can provide the optimal solution and enhance both students' and advisors' educational experience by taking into the consideration both the groups' specific interests, the number of advisees each advisor desires, and the special population (transfer students, scholarship-sponsored students, undecided, red-flag students etc.).

2 Formulation of Matching Model

For the methodology, we first start by introducing a relatively simple situation. Consider 9 incoming freshmen who need to have a pre-major advisor. Each advisor will send only one slot to each student. The maximum demands of each advisor are as follows: the math professor can take 2 students; the music professor can take 3 students; and the government professor can take 4 students. First 5 students (s1, s2, s3, s4, s5), want to major in Government, s6, s7 want to major in music and finally s8, s9 want to major in math. Both groups want to match with the counterpart with the same intended major, so how can we maximize their total satisfaction degree?

We begin by defining variables: for $i=1,2,3$ and $j=1,2,3\ldots 9$, define $x_{ij} = 1$, if advisor i takes student j, otherwise $x_{ij}=0$; define parameter $p_{ij} = a$ $(a>0)$ if advisor i and student j list the same major, otherwise $p_{ij}=0$; define parameter d_i be the demand of professors (here, $d_1=2$, $d_2=3$ and $d_3=4$). We want to maximize the total satisfaction degree of both advisees and advisors, which is the product of p_{ij} and x_{ij}; but there are restrictions such as each student only being able to have one advisor, and each advisor having their upper bound. Finally, the decision variable should be nonnegative. Therefore, combining all these factors together yield the following Linear Programming formulation:

$$
\begin{aligned}
&\text{Max } z=\sum_{i=1}^{3}\sum_{j=1}^{9} p_{ij}\,x_{ij}, \text{ for all } i=1,2,3 \text{ and } j=1,2,3\ldots 9 \\
(1)\quad &\text{s.t.} \quad \sum_{i=1}^{3} x_{ij}=1, \text{ for all } j=1,2,\ldots 9 \qquad \text{(supply constraint)} \\
&\text{s.t.} \quad \sum_{j=1}^{9} x_{ij} \le d_i, \text{ for all } i=1,2,3 \qquad \text{(demand constraint)} \\
&\quad x_{ij} \ge 0 \text{ for all } i=1,2,3 \text{ and } j=1,2,3\ldots 9
\end{aligned}
$$

For this example, the optimal solution is: $z=8$, where $x_{31}=1$, $x_{32}=1$, $x_{33}=1$, $x_{34}=1$, $x_{25}=0$, $x_{26}=1$, $x_{27}=1$, $x_{18}=1$, $x_{19}=1$. Aiming to generalize the previous simple situation, in this research we propose a model based on the Transportation Problem[1] and aims at maximizing the satisfaction degree between students *j* and their assigned advisor *i*. We define the compatibility score of advisor *i* and student *j* as p_{ij}. Specifically, p_{ij} considers the number of 36 majors and 4 concerns (being a transfer, neurodiverse or first-generation student or athlete) that advisor *i* and student *j* have both stated as an interest or concern. So *c* is the total number of common majors and concerns. d_i is defined to be the maximum slots (supply) for each professor *i*; each student can only get one advisor. If professor *i* is matched with student *j*, x_{ij} = 1, otherwise x_{ij} = 0. Then we have the following optimization problem:

$$\text{Objective function: } \max \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{j=n} x_{ij} \le d_i \quad (i=1, 2, \ldots, m) \tag{2}$$

$$\text{s.t.} \quad \sum_{i=1}^{i=m} x_{ij} = 1 \quad (j=1, 2, \ldots, n)$$

$$x_{ij} = 0 \text{ or } 1, \ (i=1,2, \ldots, m; \ j=1, 2, \ldots, n)$$

3 Define Satisfaction Level & Computing P matrix

To operationalize definition of "satisfaction level" is the most essential and interesting point of this research: the idea is to first get interest scores individually for students and advisors. Students are asked to give a ranked list of between 2 and 5 majors that they are interested in. They are also asked about whether four additional concerns (being a transfer, neurodiverse, first-gen student or athlete) apply to them. For the concerns, the students response can be "Yes", "No" or "Unsure". Similarly, advisors can choose any number of majors they are interested in. For concerns, they could response "1", "2" or "3", which have corresponding meanings of "not willing to", "neutral" and "willing to" advise students with these concerns.

Therefore, we created an S matrix (student matrix) and A matrix (advisor matrix). The S matrix has a row for each student and a column for each major and concern. For student *j* and major *k*, s_{jk} is 2.5 if the student chooses major as their first choice, 2.0 for the second, 1.5 for the third, 1.0 for the fourth and 0.5 for the fifth choice. For concern k, s_{jk} = 1 if student *j* answered "Yes" or "Unsure", and 0 if the student said "No".

[1] Please refer to the appendix for the additional information about the research resource.

The A matrix similarly has a row for each advisor and a column for each major and concern. For advisor i and major k, a_{ik} = 2 if major k is advisor i's home department, and a_{ik}= 1 if advisor i listed k as another major they are interested in. For concerns, we set $a_{ik} = -100$ if advisor i responded 1 for concern k (in order to prevent matching); a_{ik} = 0.25 if response is 2, and a_{ik} = 1 if response is 3.

Finally, we multiplied the $n \times c$ matrix S (in this case, 1615 × 40) with the transpose of the $m \times c$ matrix A (in this case, 343 × 40), thus obtaining the $n \times m$ matrix P^T (in this case, 1615 × 343). For example, the value p11 between student1 and advisor1 will be calculated as below:

	1st major	2nd major	3rd major	4th major	5th major	Transfer	Neurodiverse	athlete	First-gen
Student1	Math	Biology	CS	Art		Yes	No	No	No
S matrix	2.5	2.0	1.5	1.0		1	0	0	0

A matrix	Advisor1 math	
2	Math	1st major
1	Biology	2nd major
1	Cs	3rd major
1	Art	4th major
		5th major
0.25	2	Transfer
0.25	2	Neurodiverse
1	3	athlete
0.25	2	First-gen

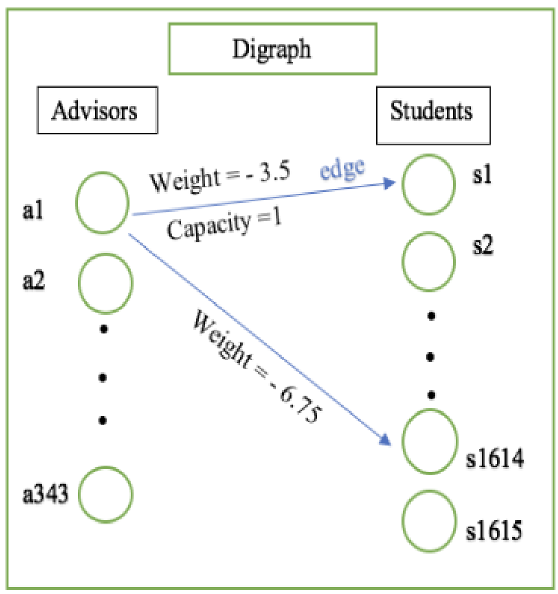
Pij = (2.5x2)+ (2x1)+(1.5x1)+(1x1)+(1x0.25)+(0x0.25)+(0x1)+(0x0.25)= 9.5

P matrix	Advisor1	Advisor2	...						Advisor 343
Student1	9.5	...	....						
...									
Student1615									

In addition, we could also calculate the two extreme cases, the worst case and best case, by using defined criteria. The lowest possible score is -400, where students and professors have no matching majors and students define himself/herself as belonging to all four concerns whereas advisors respond "1" to all concerns. The highest possible score possible is 14, where students and professors match with all five majors and students define himself/herself as belonging to all four concerns whereas advisors respond "3" to all concerns. Generally, P^T matrix is now a matrix containing compatibility scores between all professors and students. We used NetworkX package in Python to do the calculation and final matching.

4 NetworkX Matching

NetworkX is a Python package for the matching and manipulation of the structure, dynamics, and functions of complex networks. We used the *Min_Cost_Flow* class in NetworkX and Pandas package to do the final matching. We first created a digraph, i.e., directed graph, with edge costs and capacities and in which nodes have demand. In our case, advisors are supply origins: they are sending slots to students. Each student, which is a demand node, can only receive one slot, so the capacity of each edge is 1. The cost would be our compatibility scores between advisors and students. The *Min_Cost_Flow* returns a minimum cost flow satisfying all demands in digraph G. Since our objective is to maximize the positive compatibility scores, it is equivalent to minimize the negative outcome of our compatibility scores. We attach a brief example of the directed graph. The left-hand side contains advisor nodes which have 343 in total and the right-hand side has 1615 student nodes. Furthermore, the algorithm implemented by this package is the famous Simplex Algorithm, which is both efficient and accurate in practice.

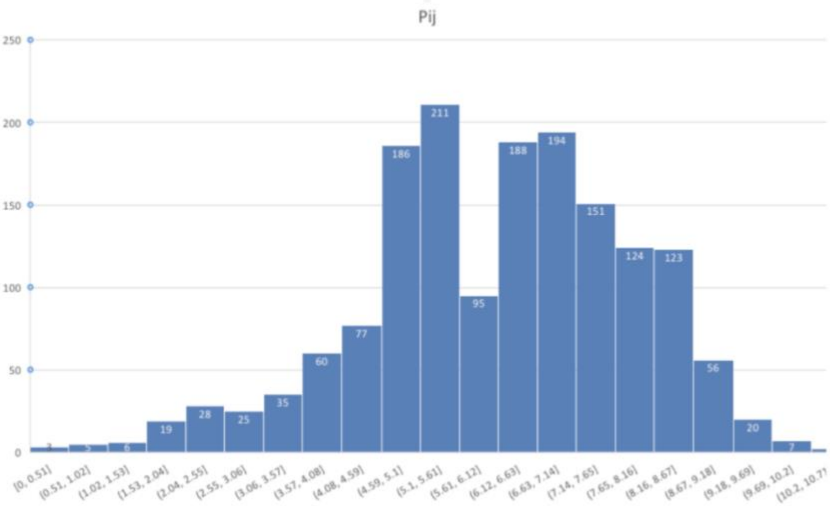


5 Effectiveness Analysis

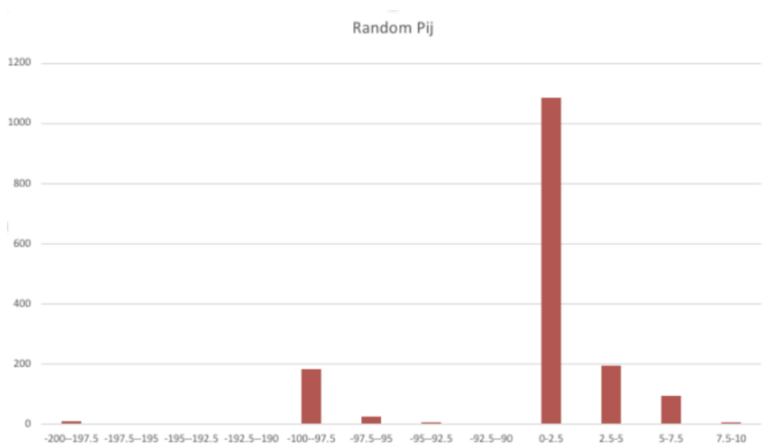
The essential goal of our research is to maximize the satisfaction degree between all students and advisors. Therefore, it is not only necessary but also important for us to analyze our matching result carefully and comprehensively. Furthermore, we compared our algorithm result this year to a random matching result in order to show our result's superiority.

*Central tendency analysis:

The range of the satisfaction degree of the final matching is from 0 (the lowest value) to 10.5 (the highest value). The average is approximately 6.166, the median is 6.4, the mode of is 5 and the sum of p_{ij} scores is 9958.5



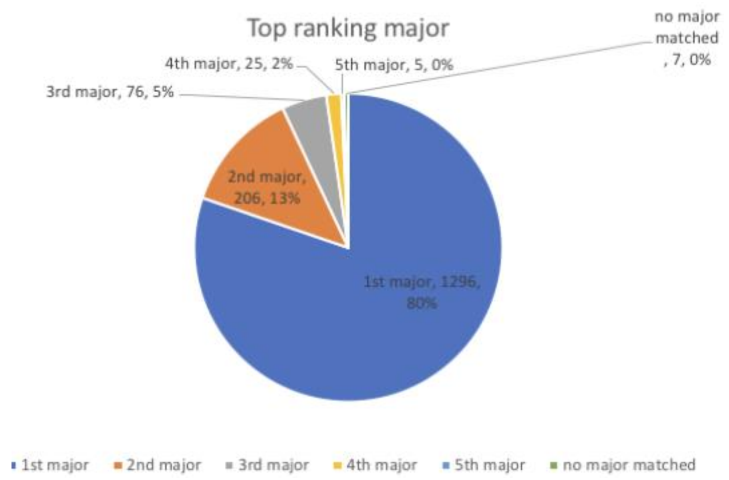
Compared with a random assignment result, the average is approximately -95.125 the median is -95.125, the mode of is 0 and the sum of p_{ij} scores is -190.25.



Therefore, in general, a significant difference is shown in the overall satisfaction degrees between these two results. We then created several pie charts to further illustrate effectiveness.

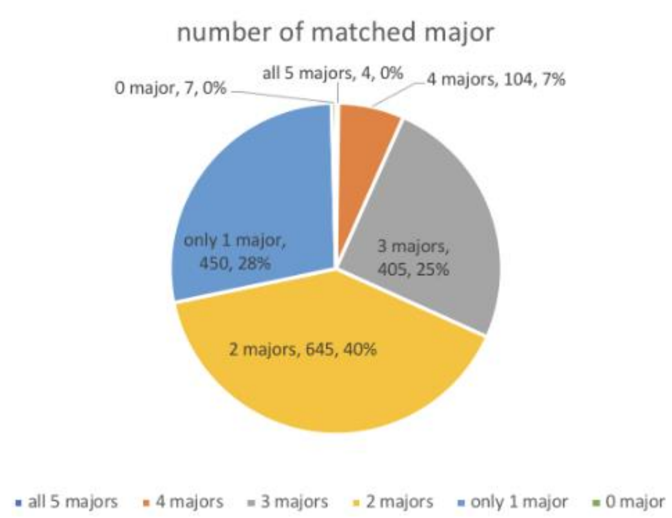
*Percentage of Top Ranking Major

A pie chart illustrating an analysis of the matchings based on the student's top ranked major is given below. It is shown that 80% of students get advisors that share their first ranked major. Only 7 students got matched to advisors that share no common majors, which was the worst case in our matching. However, we know there would be a trade-off between the satisfaction level of whole group versus individual.



*Percentage of Number of Matched Major

A pie chart analyzing the matchings based on the number of common majors of interest is given below. We usually consider one common major between advisor and student as sufficient criterion for a good matching, and our analysis shows almost 100% of students get advisors that share common majors. 72% of students got two or more majors in common with advisors.



In conclusion, our result is both valid and reliable, and it could enhance both students and advisors' academic experience greatly.

6 Related Problems/Application

Transportation Problem algorithm has a wide application in real life. This algorithm could be used not only on the matching between freshmen and advisors but also on other fields. For example, First Year Experience (FYE) at the College of William & Mary could also utilize this algorithm for matching roommates. The questionnaire they send to freshmen contains questions such as night/morning person and messy/clean person etc., so living habits are criteria for matching in this case. Furthermore, Transportation Problem algorithm could also be applied on dating website/app. Matching with a suitable girlfriend/boyfriend is just as important as matching with a pre-major advisor or roommate, and common hobbies and characteristics could be used as matching criteria. Interestingly, criteria for matching with appropriate partners could be different as some people actually prefer to match with partners who possess complementary characteristics. Moreover, transportation scheduling for railroad, flights and bus as well as route scheduling in Google map as mentioned by Will Cranford are classical examples of Transportation Problem algorithm.

7 Reflection and Conclusion

In this project, we formulated the problem of finding a match with maximum total satisfaction as a Transportation Problem, and find the optimal matching between freshman students and advisors and maximize their satisfaction degree. We worked closely with the Office of Academic Advising (OAA) to model the degree of satisfaction between a student and advisor. The result of this research leads to the improvement of the assigning task between advisees and advisors, thus empowering both student and faculty members by providing useful information to help them make better informed decisions. In the future, we would like to send a follow-up questionnaire for both advisors and students to obtain their personal opinions about the matching experience. Based on their responses, we could include more factors when considering compatibility scores. In addition, we may adjust the weight of parameters. One interesting fact we noted is that no students who answered "Yes" on concerns are matched with advisors that responded "3". A reasonable explanation is that weights of majors are higher than that of concerns. Thus, we may investigate more about this aspect and adjust weights accordingly.

Overall, this invaluable research experience prepared us for the future research process. During the process of going back and forth, this exploration deepens our understanding both of math and ourselves. These practices solidify our mathematic base and provide new perspectives on the world in which we live. On the other hand, teeming with unknown challenges, it is also a meaningful process to discover our own limits, and then, via continual study and collaborative work with the teammate, exceed them. In the end, we are very proud that our involvement actually contributed to our own community, however small.

Acknowledgment

The research is sponsored by Summer Research Scholarship from the William & Mary Charles Center. We especially thank Professor Anke van Zuylen and Dane Pascoe from OAA help us to model the satisfaction degree.

References

[1] Winston, Wayne L. *Operations research: applications and algorithms.* PWS-KENT Publ. Company, 1987.

[2] P., & L. (n.d.). The Importance of Mentors and Peers in the Undergraduate ... Retrieved from http://www.elon.edu/e/CmsFile/GetFile?FileID=1347

Appendix: Resource Description

The research project is based on the course Math323, Intro to Operational Research, and the algorithm of Transportation Problem, which was developed by Frank Lauren Hitchcock, a renowned American mathematician who is notable for vector analysis. The general formulation in *Operations Research: Applications and Algorithms*, Wayne L. Winston is:

Objective function: $min \sum_{i=1}^{i=m} \sum_{j=1}^{j=n} c_{ij} x_{ij}$

(3)

s.t. $\sum_{j=1}^{j=n} x_{ij} \leq s_i$ *(i=1, 2, . . ., m)* *(Supply constraints)*

s.t. $\sum_{i=1}^{i=m} x_{ij} \geq d_j$ *(j=1, 2, . . ., n)* *(Demand constraints)*

$x_{ij} \geq 0$, *(i=1,2, . . ., m; j=1, 2, . . ., n)*

where the s_i is the maximum units could be supplied by supply point *i*; and d_j represents that demand point *j* must receive at least d_j units of the shipped good; c_{ij} stands for a variable cost when each unit produced at supply point i and shipped to demand point j (as p_{ij}); x_{ij} stands for the number of units shipped from supply point *i* to demand point *j*. Since we assume that the cost objective function is linear, the total cost of this shipment is $c_{ij} x_{ij}$.